

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

---

## *Implementation documentation*

Milada Kovarova, Roman Repiscak, Frantisek Rudik //PosAm

Slovakia, October 21, 2010

## **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 .....	7
B.1 Short description.....	7
B.2 Public API.....	7
<i>Transcoding: getEpSOSConceptByCode()</i> .....	7
<i>Translation: getDesignationByEpSOSConcept()</i> .....	7
<i>Providing value sets: getValueSetConcepts()</i> .....	7
B.3 Setup .....	8
1.Libraries .....	8
2.Configuration.....	8
3.Database.....	8
B.4 How to use component.....	8

## **Implementation documentation**

B.5 Configuration parameters (tsam.properties) ..... 9

C.Appendix..... 11

    C.1 Complete list of TSAM Maven dependencies (testing environment) ..... 11

    C.2 Complete list of TM Maven dependencies (testing environment) ..... 11

    C.3 Spring config – TSAM - application\_context\_tsam.xml (standalone version) ..... 14

    C.4 Spring config – TM - application\_context\_tm.xml & application\_context\_tsam.xml ..... 16

    C.5 tm.properties..... 19

    C.6 tsam.properties..... 20

## 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

(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()**

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

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

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

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

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:

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

*tmConfiguration* – defines properties for document types, validation, auditing

#### Implementation documentation

*codedElementList* - defines properties for coded element list identification. Singleton which contains method for effective work with coded element list by process of transcoding/translation.  
*schematronValidator* - defines properties for schematron based validation of CDA documents. Bean si singleton which contain 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 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. (are 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 put contents of both files (*tm.properties* and *tsam.properties*) into one file. Then is important proper configuration of placeholderConfig bean, 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.

Property Name	Type	Value	Description
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 abosolute	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	Actual path to schema file.( CDA_extended.xsd)

### Implementation documentation

		or abosolute	
tm.schematron.validation.enabled	boolean	true or false	Flag indicating that schematron validation is enabled or disabled.
tm.schematron.path.xsldir	path	path relative or abosolute	Actual path to schematron XSL directory.(../resources/xsl)
tm.schematron.path.edispensation	path	path relative or abosolute	Actual path to eDispensation Schematron file. (../resources /schematron/eDispensation.sch)
tm.schematron.path.eprescription	path	path relative or abosolute	Actual path to ePrescription Schematron file. (../resources /schematron/ePrescription.sch)
tm.schematron.path.patientsummary	path	path relative or abosolute	Actual path to pasienSummary Schematron file. (../resources /schematron/PatientSummary.sch)
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.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

### Implementation documentation

## 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()*

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

#### Translation: *getDesignationByEpSOSConcept()*

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

#### Providing value sets: *getValueSetConcepts()*

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

## 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



- 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**



## 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

#### Implementation documentation

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

## Implementation documentation

Transformation Manager  
Terminology Services Access Manager



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>classpath:tsam.properties</value>
                <value>classpath: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>${tm.documenttype.edispensation}</value>
        </property>
        <property name="schemaFilePath">
            <value>${tm.schemafilepath}</value>
        </property>
        <property name="patientSummarySchematronPath">
            <value>${tm.schematron.path.patientsummary}</value>
        </property>
        <property name="ePrescriptionSchematronPath">
            <value>${tm.schematron.path.eprescription}</value>
        </property>
        <property name="eDispensationSchematronPath">
            <value>${tm.schematron.path.edispensation}</value>
        </property>
        <property name="schematronValidationEnabled">
            <value>${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>
    </bean>
</beans>
```

### Implementation documentation



```

        <property name="auditTrailFacility">
            <value>${tm.audittrail.facility}</value>
        </property>
        <property name="auditTrailSeverity">
            <value>${tm.audittrail.severity}</value>
        </property>
    </bean>

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

    <!-- ::: Service ::: -->
    <import resource="application_context_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" />
    </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}"/>
    </bean>

```

## Implementation documentation

```

        <property name="synchronizationEnabled" value="{synchronizationEnabled}"/>
    </bean>

    <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

```
# coded element processing
tm.codedelementlist.enabled=true

# actual path to coded_element_list.xml file
tm.codedelementlist.path=.\src\test\resources\coded_element_list.xml

# actual path to schema file
tm.schemafilepath=.\src\main\resources\schema\CDA_extended.xsd

# schema validation enabled
tm.schema.validation.enabled=true

# actual path to patientSummary Schematron file
tm.schematron.path.patientsummary=.\src\main\resources\schematron\PatientSummary.sch

# actual path to ePrescription Schematron file
tm.schematron.path.eprescription=.\src\main\resources\schematron\ePrescription.sch

# actual path to eDispensation Schematron file
tm.schematron.path.edispensation=.\src\main\resources\schematron\eDispensation.sch

# actual path to schematron XSL directory
tm.schematron.path.xsldir=.\src\main\resources\xsl

# schematron validation enabled
tm.schematron.validation.enabled=false

# epSos code for patient summary CDA document
tm.documenttype.patientsummary=RELMAD572

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

# epSos code for eDispensation CDA document
tm.documenttype.edispensation=RELMA9ED5

# 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
```