

Topic Maps mit TMAPI 2.0

Tobias Redmann
7. Mai 2009 – HTW Berlin

Überblick

1. Was ist die TMAPI?
2. Arbeiten mit Topic-Maps und der TMAPI
3. Was gibt es noch?
4. Nützliches außerhalb der TMAPI
5. Werbung 😊
6. Themen für Projekt- und Abschlussarbeiten 😊

Anmerkung

ICH HASSE POWERPOINT!

Was ist die TMAPI?

- Ist ein Programmier-Interface um auf die Daten innerhalb einer Topic-Map zuzugreifen und sie zu manipulieren.
- Es werden eine Reihe von Core-Interfaces zur Verfügung gestellt, die alle Topic-Map-Engines nutzen sollten.
- Die TMAPI stellt keinerlei Funktionalität zur Verfügung (!), sondern lediglich Funktionsbeschreibungen die Engines implementieren sollten.
- Vorteile:
 - Entwickler müssen „nur noch“ den Code für ihre Engine implementieren
 - Interoperabilität von Topic-Map-basierten Anwendungen

Arbeiten mit Topic Maps und TMAPI

1. Erstellung eines Topic-Maps-System zur Verwaltung von mehreren Topic-Maps
2. Erstellen oder laden einer Topic-Map
3. Erstellen oder bearbeiten von Topics
 - Vergabe von Namen
 - Vergabe von Varianten von Namen
 - Zuweisen von Typen
 - Zuweisen von Subjekt-Identitäten
4. Anlegen oder bearbeiten von Occurrences zu Topics
 - Zuweisen von Types und Scopes
5. Erstellen oder bearbeiten von Assoziationen (Associations) um Beziehungen zwischen Topics darzustellen
 - Zuweisen von Topics
 - Zuweisen von Types und Scopes
6. Speichern der Topic-Maps

Topic-Map-System

- Verwaltet ein oder mehrere Topic-Maps
- Bietet Zugriff, erlaubt neue Topic-Maps anzulegen und vorhandene zu löschen
- Zugriff auf Topic-Maps erfolgt über Base-Locator (URL)
- Verhalten des Systems kann mit Features beeinflusst werden, z.B. Datenmodell, Merging-Verhalten etc.

Jetzt:

- Erstellen eines Topic-Map-System
- Setzen und auslesen von Features

Beispiel: Topic-Map-System erstellen

```
// neue Factory erzeugen
TopicMapSystemFactory tmSysFactory;
tmSysFactory = TopicMapSystemFactory.newInstance();

// System erzeugen
TopicMapSystem tmSys;
tmSys = tmSysFactory.newTopicMapSystem();
```

Beispiel: Setzen von Features

```
// Datenmodell TMDM setzen  
tmSysFactory.setFeature(  
    "http://tmapi.org/features/model/xtm1.1 ",  
    true  
);
```

Alle Features unter: <http://tmapi.org/features/>

Beispiel: Status eines Features ermitteln

```
boolean isFeatureSet =
    tmSysFactory.getFeature("http://tmapi.org/
    features/model/xtm1.1 ");

if (isFeatureSet) {
    System.out.println("Feature ist angestellt");
} else {
    System.out.println("Feature ist ausgestellt");
}
```

Beispiel: Feature im System vorhanden

```
boolean sysHasFeature =
    tmSysFactory.hasFeature("http://tmapi.org/
    features/model/xhtml1.1 ");

if (sysHasFeature) {
    System.out.println("Feature ist vorhanden");
} else {
    System.out.println("Feature nicht möglich");
}
```

Topic-Map

- Grundlage jeder weiteren Betrachtung (!)
- Jede Topic-Map innerhalb eines Topic-Map-Systems wird durch einen eindeutigen Locator (Base-Locator) beschrieben
- (Locator kann oft auch beliebige Zeichenkette sein)

Jetzt

- Erstellen einer Topic-Map mit Hilfe des Topic-Map-Systems
- Zugriff auf eine Topic-Map des Systems
- Löschen einer Topic-Map

Beispiel: Erstellen einer Topic-Map

```
// Erstellen der Topic-Map bei tmSys  
TopicMap tm =  
tmSys.createTopicMap("http://topic-maps-buch.de");
```

Beispiel: Zugriff auf eine Topic-Map

```
// Topic-Map aus System laden
TopicMap tm =
tmSys.getTopicMap("http://topic-maps-buch.de");

if (tm == null) {
    System.out.println("TM nicht vorhanden");
}
```

Achtung:

- Gibt `null` zurück wenn die Topic-Map nicht vorhanden ist

Beispiel: Löschen einer Topic-Map

```
// Löschen der Topic-Map tm  
tm.remove();
```

Wichtige Interfaces: Construct, Scoped, Typed

- **Construct**

- `getId` – Identifier in der Topic-Map
- `getParent` – gibt das Eltern-Construct zurück
- `remove` – entfernt das Construct vom Eltern-Container
- Operationen für Item-Identifier – relevant für XTM

- **Scoped**

- `addTheme` – Topic als Scope hinzufügen
- `getScope` – Menge an Topics
- `removeTheme` – entfernt ein bestimmtes Topic vom Scope

- **Typed**

- `getType` – Topic das den Type näher bestimmt
- `setType` – Type per Topic setzen

Topic

- Zentrale Element einer Topic-Map
- es werden Aussagen über den Diskursbereich getroffen

Jetzt:

- Erzeugen eines Topics
- Setzen von Topic-Types
- Ermitteln aller Topics einer Topic-Map

Beispiel: Erzeugen eines Topics

```
// 1.) Topic ohne Subject erzeugen
Topic fruehstueck = tm.createTopic();

// 2.) Topic mit Subject-Locator
Locator broetchenSubj = tm.createLocator("http://kamps.de/1.jpg");

Topic broetchen = tm.createTopicBySubjectLocator(broetchenSubj);

// 3.) Topic mit Subject-Identifizier
Locator koernerSubj = tm.createLocator("http://bio.de");

Topic koernerBroetchen = tm.createTopicBySubjectIdentifizier(koernerSubj);
```

Beispiel: Setzen und Löschen von Topic-Types

ACHTUNG: Topic implementiert nicht Typed!

```
// 1.) koernerBroetchen zu einem broetchen machen  
koernerBroetchen.addType(broetchen);
```

```
// 2.) alle Topic-Types ermitteln  
for (Topic type : koernerBroetchen.getTypes()) {  
    System.out.println(type.getId());  
}
```

```
// 3.) Type wieder entfernen  
koernerBroetchen.removeType(broetchen);
```

Beispiel: Alle Topics einer Topic-Map ermitteln

```
// Alle Topics ermitteln
for ( Topic topic : tm.getTopics() ) {
    System.out.println(topic.getId());
}
```

Topic-Names (Names)

- (eindeutige) Bezeichnungen für Topics innerhalb einer Topic-Map
- Sollen die abzubildenden Konzepte möglichst gut beschreiben

Jetzt:

- Erstellen von Topic-Names
- Setzen und auslesen von Werten
- Alle Names eines Topics ermitteln

Beispiel: Erstellen von Topic-Names

```
// 1.) mit Wert ohne Scope
Name bN1 = broetchen.createName("Brötchen", null);

// 2.) mit Wert und Scope (berlin ist ein Topic)
Name bN2 = broetchen.createName("Schrippe", berlin);

// 3.) mit Type, Wert ohne Scope
Name bN3 =
broetchen.createName(foreign, "bun", null);

// 4.) mit Type, Wert und Scope
Name bN4 =
broetchen.createName(foreign, "croissant", french);
```

Beispiel: Setzen und auslesen von Werten

```
// Auslesen des Wertes von bN4  
System.out.println(bN4.getValue());
```

```
// Setzen eines neuen Wertes  
bN4.setValue("Croissant");
```

Beispiel: Alle Names eines Topics ermitteln

```
// Alle Names ermitteln und ausgeben
for (Name name : broetchen.getNames()) {
    System.out.println(name.getValue());
}
```

```
// Names eines Types
for (Name typedName : broetchen.getNames(foreign)) {
    System.out.println(typedName.getValue());
}
```

Variants (Variant-Names)

- Alternative Namen für bestimmte Anwendungsfälle, z.B.:
 - Lautschrift
 - Bilder
 - Morsecode
 - EAN
 - ISBN
 - ...

Jetzt:

- Erstellen von Variants
- Alle Variants eines Names auslesen und Unterscheidung verschiedener Datentypen

Beispiel: Erstellen von Variants

```
// 1.) Mit Wert und Scope
Variant bV1 = bN4.createVariant("crs", french);

// 2.) Mit Ressource und Scope
Locator crsMorse = tm.createLocator("morse.mp3");

Variant bV2 = bN4.createVariant(crsMorse, french);

// 3.) Mit Wert, Datatype und Scope
Locator strLoc =
tm.createLocator("http://www.w3.org/2001/XMLSchema#string");

Variant bV3 =
bN4.createVariant("French Croissant", strLoc, french);
```

Beispiel: Unterscheiden der Variants

```
// alle Variants von bN4 ermitteln
for (Variant variant : bN4.getVariants()) {

    // Variant mit Wert oder Ressource?
    if (variant.getDataType().getReference() ==
        "http://www.w3.org/2001/XMLSchema#string") {

        // Wert
        System.out.println(variant.getValue());
    } else {

        // es wird die Referenz ausgegeben
        System.out.println(variant.getValue());
    }
}
```

Occurrences

- es können Informationsressourcen oder konkrete Werte mit Topics verknüpft werden
- Beispiele:
 - Website
 - Bilder
 - Texte
 - ...

Jetzt:

- Erstellen von Occurrences
- Alle Occurrences eines Topics ermitteln und Unterscheidung verschiedener Datentypen am Beispiel des Zählens von Occurrences mit String-Werten

Beispiel: Erstellen von Occurrences

```
// Occurrences haben immer einen Type
// 1.) Mit Wert, ohne Scope
Occurrence occAge =
tobias.createOccurrence(alter, "29", null);

// 2.) Mit Wert und Scope
Occurrence occWeight =
tobias.createOccurrence(weight, "75", today);

// 3.) Mit Ressource, ohne Scope
Locator tobiImg = tm.createLocator("http://flickr.com/1");

Occurrence occPotrait =
tobias.createOccurrence(image, tobiImage);

// 4.) Mit Ressource und Scope
Locator tobiOldImg = tm.createLocator("http://flickr.com/2");

Occurrence occOldPortrait =
tobias.createOccurrence(image, tobiOldImg, twentyYearsAgo);
```

Beispiel: Zählen der Occurrences mit Werten

```
int occ = 0;

for (Occurrence occurrence : tobias.getOccurrences()) {

    // Hat Occurrence String-Wert?
    if( occurrence.getDataType().getReference() == "http://
        www.w3.org/2001/XMLSchema#string") {

        occ++;

    }

}

System.out.println(" Anzahl: " + occ);
```

Associations / Assoziationen

- Associations stellen „irgendeine“ Verbindung zwischen Topics dar

Kurz und gut:

```
// Associations haben immer einen Type
// 1.) Association ohne Scope
Association ass1 =
tm.createAssociation(default, null);
```

```
// 2.) Association mit Scope
Association ass2 =
tm.createAssociation(default, today);
```

Association-Roles

- Legen die „Teilnehmer“ einer Association fest
- Teilnehmer sind Topics
- Typen sind Topics

Jetzt:

- Roles einer Association erstellen
- Roles einer Association ermitteln

Beispiel: Roles einer Association erstellen

```
// Rolle 1 erstellen  
Role role1 = ass1.createRole(sohn, tobias);  
  
// Rolle 2 erstellen  
Role role2 = ass1.createRole(vater, peter);
```

Beispiel: Rollen auslesen

```
// 1.) alle Rollen auslesen
for (Role role : ass1.getRoles()) {

    // Teilnehmer ermitteln
    Topic rolePlayer = role.getPlayer();

    // Ausgabe der ID
    System.out.println(rolePlayer.getId());
}

// 2.) alle Väter auslesen
for (Role vaterRolle : ass1.getRoles(vater)) {

    // Vater-Teilnehmer ermitteln
    Topic vaterRollenSpieler = vaterRolle.getPlayer();
}
```

Es gibt noch mehr in TMAPI

- Indexe für den schnellen Zugriff auf Topic-Map-Constructs
 - LiteralIndex: Namen, Occurrences und Variants per Locator oder Wert ermitteln
 - ScopedIndex: Associations, Names, Occurrences und Variants anhand eines Scopes ermitteln
 - TypeInstanceIndex: Associations, Names, Occurrences, Roles und Topics anhand des Types ermitteln
- Reification: Aussagen über Topic-Map-Constructs
- Vordefinierte Klassen die durch PSI genutzt werden können, z.B. Sub-Super-Class-Relation

Nützliches (nicht in TMAPI)

- Lesen von Topic-Maps
- Speichern von Topic-Maps

Beides ist kein Bestandteil von TMAPI, sondern für tinyTiM verfügbar

Packages: org.tinytim.mio

URL: <http://tinytim.sourceforge.net/docs/2.0/mio/index.html>

Beispiel: Lesen von XTM-Dateien

```
// Welche Datei soll gelesen werden
File source = new File("tm.xtm");

// Reader erstellen
XTMTopicMapReader reader = new XTMTopicMapReader(tm,
    source);

// lesen
reader.read();
```

Beispiel: Speichern von Topic-Maps

```
// Welche Datei soll geschrieben werden
OutputStream out = new FileOutputStream(new
    File("out.cxtm"));

// Writer erstellen
TopicMapWriter writer = new CXTMTopicMapWriter(out,
    "http://tinytim.sourceforge.net/examplemap.ctm");

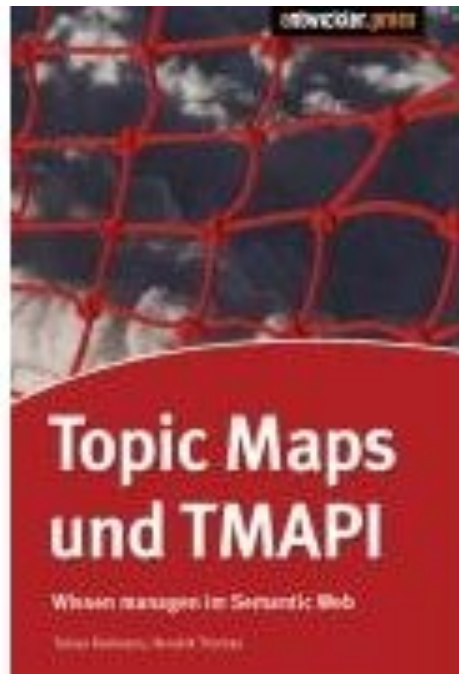
// Schreiben
writer.write(tm);
```

Ressourcen

- TMAPI 2.0 Homepage:
<http://tmapi.org/2.0/>
- tinyTiM Homepage:
<http://tinytim.sourceforge.net/>
- ISO-Topic-Maps:
<http://isotopicmaps.org/>
- Topic Maps mit TMAPI: Website zum Buch mit TMAPI 1.0 Beispielen:
<http://www.topic-maps-buch.de>

Werbung!

Kaufen Sie mein Buch, wenn es endlich fertig ist!



Topic Maps und TMAPI

Wissen managen im Semantic Web

von Tobias Redmann und Hendrik Thomas

ISBN: 103868020101

Abschlussarbeit?

- **Persistenzschicht basierend auf SimpleDB für tinyTim 2.0**
- **TMAPI-kompatible Topic-Map-Engine für J2ME (mit Persistenzschicht) / nur Core ohne Index**
- **Grafischer Topic-Map-Editor basierend auf GTM alpha**

... und was Sie wollen?!