



Localizzazione in Java tramite lo standard TMX

Nicola Asuni

<http://nicolaasuni.tecnick.com>



Introduzione

Durante questa presentazione verranno accennati alcuni aspetti relativi all'**internazionalizzazione** del Software, verrà introdotto lo standard **TMX** e verrà illustrato un semplice esempio di implementazione di questo standard in **Java™**.

* Java is a trademark or registered trademark of Sun Microsystems, Inc. in the U.S. and other countries.



Localizzazione

Uno degli aspetti fondamentali dell'internazionalizzazione (**i18n**) consiste nel separare dal nucleo del codice sorgente i testi, le etichette, i messaggi ed altre informazioni sensibili alla localizzazione.

Questo consente di mantenere la stessa base di codice sorgente per tutte le versioni del Software nelle diverse lingue e facilita il processo di traduzione perché tutte le **risorse sensibili al contesto locale** sono ben identificate ed isolate.



TM - Translation Memory

Le **TM** (Translation Memories), conosciute anche come **Translation Databases**, sono essenzialmente costituite da un database (archivio) dove le frasi del testo di riferimento sono associate alle corrispondenti traduzioni in una o più lingue.

L'insieme della frase di riferimento con le sue traduzioni è detto **Translation Memory Unit**.

Le **TM** introducono nella gestione delle risorse localizzabili, i concetti di **riuso, aumento della consistenza, riduzione della durata del ciclo produttivo e riduzione dei costi**.



CAT - Computer Aided Translation

Le applicazioni che gestiscono le **TM** sono dette **CAT** (Computer Aided Translation).

I **CAT**, ampiamente utilizzati dai maggiori produttori di Software, sono strumenti di ausilio alla traduzione linguistica progettati per migliorare la qualità e l'efficienza del processo di traduzione umana e non per sostituirlo.

I **CAT** attualmente certificati **TMX** sono:
TRADOS 7 (*Trados*), WorldServer (*Idiom Technologies, Inc*),
SDLX (*SDL International*), Ambassador (*GlobalSight*).

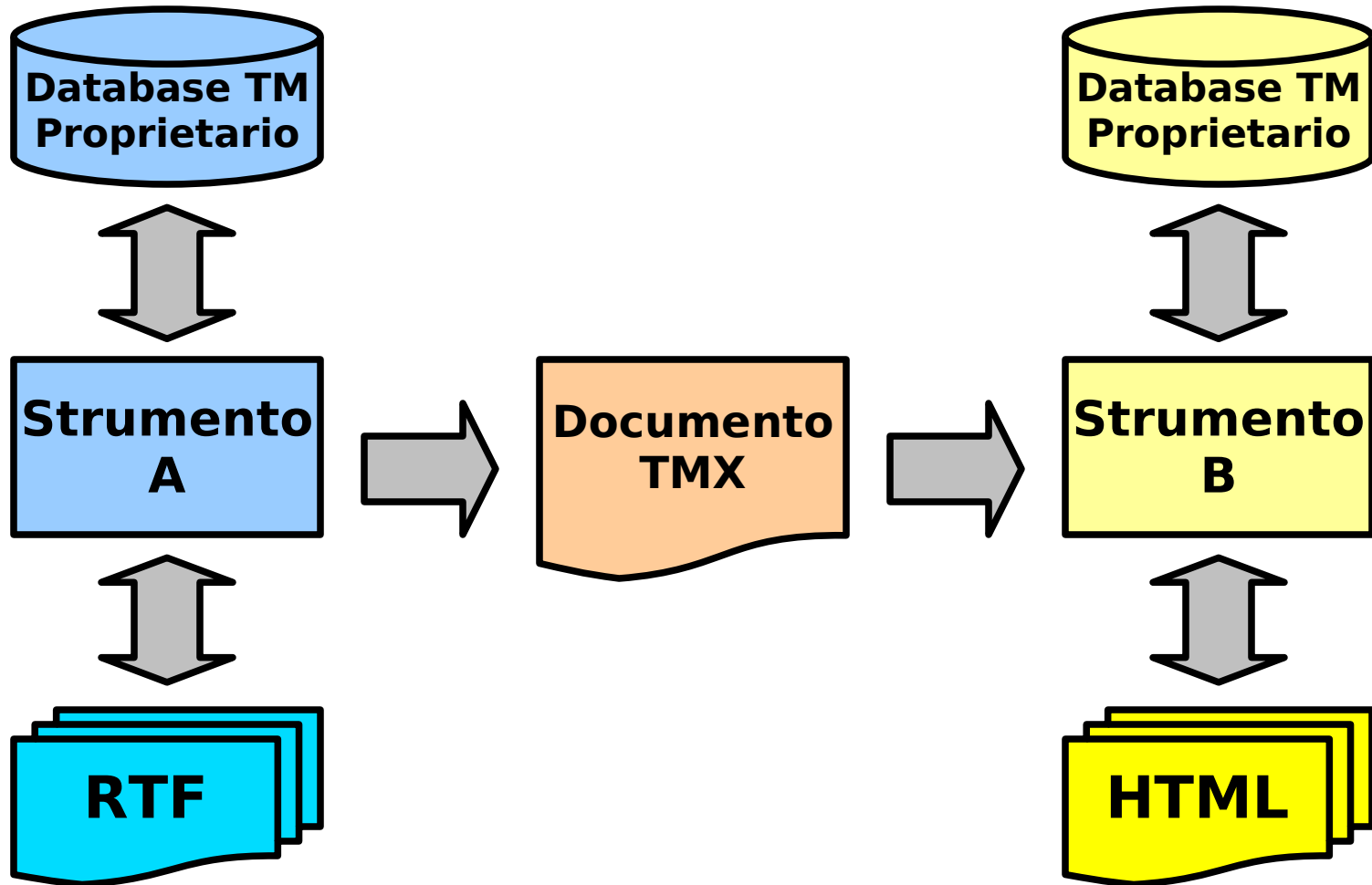


TMX - Translation Memory eXchange

TMX (Translation Memory eXchange) è un **Open Standard** che utilizza l'**XML** (eXtensible Markup Language) per l'archiviazione e l'interscambio di memorie di traduzione (**TM** - Translation Memories).

Lo standard **TMX** è un sistema neutrale per lo scambio di dati tra diversi sistemi di traduzione che minimizza o elimina la perdita di dati critici.

Esempio d'uso del TMX





LISA (<http://www.lisa.org>)

TMX è il risultato di un'iniziativa presa dallo *Special Interest Group* **OSCAR** (Open Standards for Container/Content Allowing Re-use) di **LISA** (Localization Industry Standards Association).

Fondata nel 1990, LISA è la principale organizzazione mondiale no-profit per il **GILT** (Globalizzazione, Internazionalizzazione, Localizzazione e Traduzione).

Tra le centinaia membri e co-fondatori di **LISA** spiccano le organizzazioni internazionali per la standardizzazione ed alcune tra le aziende più grandi e conosciute a livello mondiale.



Vantaggi del TMX

L'adozione dell'Open Standard **TMX** fornisce un modo neutrale per la gestione delle **TM** in maniera indipendente dagli strumenti **CAT** utilizzati, mettendoci così al riparo da futuri cambiamenti del mercato o delle tecnologie.

Secondo una ricerca condotta da **OSCAR** nel 2004 (<http://www.lisa.org/products/surveys/tm04survey.html>), lo standard **TMX** si pone a tutela degli enormi investimenti economici (milioni di dollari) fatti dai principali produttori di Software per la gestione delle proprie **TM**.



Specifiche TMX

L'attuale versione delle specifiche TMX è la **1.4b** del 26 Aprile 2005 ed è disponibile su:

<http://www.lisa.org/standards/tmx/tmx.html>

Lo standard **TMX** è ***XML-compliant*** ed utilizza vari standard **ISO** per la data/ora, per i codici delle lingue e delle nazioni.

I file TMX sono documenti **XML "well-formed"** in **Unicode** pensati per essere esportati ed importati automaticamente, e che quindi possono essere trattati senza un esplicito riferimento al **DTD** TMX (Document Type Definition).



Struttura Generale dei file TMX (1/3)

Un documento TMX è racchiuso dall'elemento radice **<tmx>** che a sua volta contiene gli elementi **<header>** e **<body>**.

L'**<header>** contiene i meta-dati del documento come attributi; può contenere informazioni a livello di documento con gli elementi **<note>** (note) e **<prop>** (property) e gli elementi **<ude>** (user-defined encoding) per elencare i caratteri definiti dall'utente.

L'elemento **<body>** contiene una collezione non ordinata di **<tu>** (translation units).



Struttura Generale dei file TMX (2/3)

Ogni elemento **<tu>** identifica attraverso l'attributo **tuid** una determinata risorsa testuale.

Ogni **<tu>** contiene almeno un elemento **<tuv>** (translation unit variant) che specifica la lingua della traduzione con l'attributo **xml:lang**.

Ogni **<tuv>** contiene l'elemento **<seg>** (segment) che specifica le risorse testuali e può anche contenere gli elementi **<note>** and **<prop>**.



Struttura Generale dei file TMX (3/3)

Un segmento **<seg>** può contenere anche elementi di markup: gli elementi **<bpt>**, **<ept>**, **<it>** e **<ph>** permettono di incapsulare il codice nativo originale; l'elemento **<hi>** permette di aggiungere degli elementi di markup non previsti; l'elemento **<sub>** delimita le risorse testuali incapsulate all'interno di regioni di markup.



Esempio di file TMX (1/2)

Di seguito un semplice file TMX che utilizzeremo nei nostri esempi.

```
<?xml version="1.0" ?>
<tmx version="1.4">
  <header
    creationtool="XYZTool"
    creationtoolversion="1.01-023"
    datatype="PlainText"
    segtype="sentence"
    adminlang="en-us"
    srclang="EN"
    o-tmf="ABCTransMem">
  </header>
```



Esempio di file TMX (2/2)

```
<body>
  <tu tuid="hello">
    <tuv xml:lang="en">
      <seg>hello</seg>
    </tuv>
    <tuv xml:lang="it">
      <seg>ciao</seg>
    </tuv>
  </tu>
  <tu tuid="world">
    <tuv xml:lang="en">
      <seg>world</seg>
    </tuv>
    <tuv xml:lang="it">
      <seg>mondo</seg>
    </tuv>
  </tu>
</body>
</tmx>
```



Bridge Software per TMX

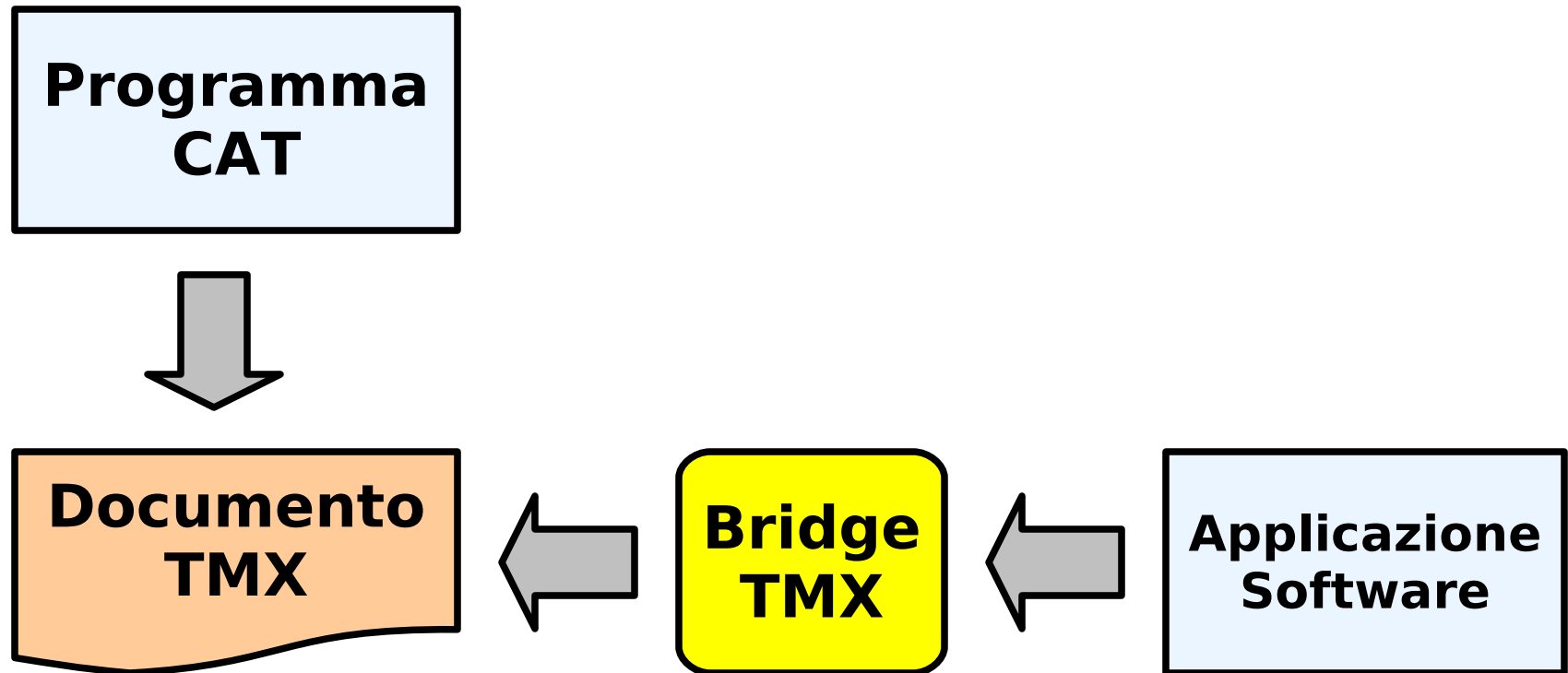
Oltre che come file di interscambio tra **TM**, il **TMX** può anche essere utilizzato direttamente per archiviare le risorse testuali dei nostri progetti software, separando così la base di codice sorgente dalle informazioni sensibili alla localizzazione (testi, etichette, messaggi, ...).

Anziché inserire le stringhe direttamente nel codice sorgente, ne richiameremo il loro valore specificando l'indice di una collezione di risorse testuali.

Es: `System.out.println("ciao");` diventa
`System.out.println(tmx.getString("ciao_id"));`



Schema del Bridge TMX





Bridge Java per TMX

Attraverso la classe **java.util.ResourceBundle**, Java dispone già di una soluzione pratica per la localizzazione. È infatti possibile estrarre gli elementi testuali dal codice sorgente originale ed isolarli in un componente **ResourceBundle** quale ad esempio la classe **ListResourceBundle** o un file di proprietà.

La soluzione ottimale per l'implementazione del bridge consiste quindi nell'estendere la classe **ResourceBundle** in modo che possa leggere direttamente i dati da file **TMX**.



TMXResourceBundle in Java

La classe *Open Source (LGPL)* **TMXResourceBundle**, disponibile su <http://tmxjavabridge.sourceforge.net>, implementa un bridge minimale in Java per TMX.

L'utilizzo di questa classe è piuttosto semplice:

```
// includo la classe TMXResourceBundle
import com.tecnick.tmxjavabridge.TMXResourceBundle;

// istanzio un oggetto della classe
final static TMXResourceBundle tmxres =
    new TMXResourceBundle("file_tmx.xml", "it");

// stampo il valore di una risorsa testuale
System.out.println(tmxres.getString("hello", ""));
```



Funzionamento del bridge Java

Il metodo **parseXmlFile** di questa classe fa uso del parser XML di Sun (**javax.xml.parsers.DocumentBuilder.parse**) che provvede a caricare la struttura del file XML in un oggetto di tipo **org.w3c.dom.Document** (**DOM** - Document Object Model).

I nodi del documento vengono analizzati in modo iterativo e le risorse testuali vengono aggiunte ad **hashcontents** (un oggetto di tipo **java.util.Hashtable**) usando come chiavi i valori degli attributi **tuid** dei nodi **tu**.

Il metodo **getString(String key)** ereditato da **ResourceBundle** permette di ricavare la risorsa associata ad una determinata chiave.



Riferimenti

- Asuni N, “Localizzazione in PHP tramite lo standard TMX” [online] 2004-10-21, http://www.tecnick.com/public/code/cp_dpage.php?aiocp_dp=article_tmphp.
- Asuni N, “Localizzazione in Java tramite lo standard TMX” [online] 2004-10-14, http://www.tecnick.com/public/code/cp_dpage.php?aiocp_dp=article_tmx.
- Asuni N, “TMXResourceBundle - TMX PHP Bridge” [online] 2005-01-08, <http://tmxphpbridge.sourceforge.net>.
- Asuni N, “TMXResourceBundle - TMX Java Bridge” [online] 2005-01-08, <http://tmxjavabridge.sourceforge.net>.
- Itagaki M, “Use XML as a Java Localization Solution” [online] 2000-11-10, <http://www.ftponline.com/javapro/archives/mi0011/default.asp>.
- O'Conner J, “Java Internationalization: Localization with ResourceBundles” [online] 1998-10-01, <http://java.sun.com/developer/technicalArticles/Intl/ResourceBundles/>.
- OSCAR - LISA, “TMX - Translation Memory eXchange” [online] 2004-10-01, <http://www.lisa.org/standards/tmx>.
- OSCAR - LISA, “TMX 1.4b Specification” [online] 2005-03-26, <http://www.lisa.org/standards/tmx/tmx.html>.
- W3C, “Extensible Markup Language (XML)” [online] 2005-08-02, <http://www.w3.org/XML/>.



Grazie per l'attenzione.