Grant Agreement Number **ECP 2006 DILI 510049**

ENRICH

ENRICH Translation Stylesheet delivery report

Deliverable number	*D-6.3*
Dissemination level	*Public*
Delivery date	*15. 12. 2009*
Status	*Final*
Author(s)	*SYSTRAN S.A., France*

*e**Content**plus*

D-6.3 ENRICH Translation Stylesheet delivery report

Document Version Control

Version	Date	Change Made (and if appropriate reason for change)	Name and Institution of Commentator(s) or Author(s)
1.0	23rd November 2009	The first draft	SYSTRAN S.A., France
1.1	27th November 2009	Formal corrections	CCP

Document Review

Reviewer	Institution	Date and result of Review
Jakub Heller	Cross Czech a.s.	27. 11. 2009 Formal corrections.
Zdeněk Uhlíř	NKP	15. 12. 2009, Approved for submission

Document Signature/Approval:

Before the table of contents each document is to contain an approval signoff form.

Approved By (signature)	Date

Accepted by at European Commission (signature)	Date

Structure Summary

The Work package 6 aimed on the integration of a tailored multilingual module via user friendly sophisticated access.

Based on SYSTRAN's machine translation technology, this module provided also terminology extraction and machine translation customization tools for the construction and retrieval of personalised metadata within the aim to create new multilingual digital documents and multilingual ontologies in Czech, Polish, Spanish, Portuguese, German, Italian, English, French, Danish, Russian and Serbo-Croatian.

One of the first objectives of the workpackage was to tailor the system so as to generate & provide personalisation resources for the translation of specific fields of the manuscripts description and/or documentation. The Enrich Translation Stylesheet developed by SYSTRAN and presented in the deliverable was based on the inventory and analysis of the Enrich manuscripts structure. The XSL Transformation stylesheets are usually used to transform a document described in XML formalism into another XML formalism, to modify an XML document, or to publish content stored into an XML document to a publishing format (XSL-FO, (X)HTML…) in order to leverage easier this structure during the translation process and to keep it in the translated document, than with traditional document filters, which process the entire document linearly.

SYSTRAN Translation Stylesheets (STS) use XSLT to drive and control the machine translation of XML documents (native XML document formats or XML representations — such as XLIFF — of other kinds of document formats).

STS does not only provide a simple way to indicate which part of the document text is to be translated, but also enables the fine-tuning of translation, especially by using the structure of the document to help disambiguate natural language semantics and determine proper context. In that case, the STS would pass a title option to the translation engine. The stylesheet can activate Enrich-specific dictionaries for some parts of the document and can mark some expressions as not to be translated, in the same manner. The mechanism is implemented through XSLT extension functions. In particular, the stylesheet uses a systran:translate function to translate an XML fragment, and systran: getValue/ systran:pushValue/ systran:popValue functions for consulting and for setting linguistics options in the translation engine. Proper management of character properties is also provided so that, for instance, the translation of a phrase in bold font will appear in bold font, even if the phrase has moved within the translated sentence.
This process is highly customizable by the addition of new templates into the stylesheets.
Based on the initial corpus structure, SYSTRAN has developed an xslt formalism allowing to the system to translate specific fields of the content metadata validated by the content partners. The SYSTRAN Translation Stylesheet had been personalized during the project according to the content structure and the associated dtds utilized by the content partners, concuding to the final version utilized by the web service called by Manuscriptorium based on the TEI.

Table of contents

1   Introduction

Based on the content metadata samples provided by content partners, SYSTRAN analysed the content and developed the necessary xslt formalism allowing to translate specific content fields with respective translation parameters such as usage of customization resources, usage of specific linguistic and translation options via ENRICH Translation Stylesheet.

This document aims to present the ENRICH Translation Stylesheet developed by SYSTRAN during the project based on the implementation tests and translation evaluation results.

Following the lines of the Description of Work, ENRICH, Annex 1, the main objectives of the Multilingual and user friendly sophisticated access are described in the Work package 6.

Work package Description					

Work package number :	**6**	**Start date:**	**3**	**End date:**	**24**
Work package title:	**Multilingual and user friendly sophisticated access**				

Objectives

This work package aims to the integration of a multilingual module via a user friendly sophisticated access: multilingual search application, multilingual forums, and multilingual ontology editor.
Based on SYSTRAN's machine translation technology, this module will provide also terminology extraction and machine translation customization tools for the construction and retrieval of personalised metadata within the aim to create new multilingual digital documents and multilingual ontologies in Czech, Polish, Spanish, Portuguese, German, Italian, English, French, Danish, Hungarian, Russian and Serbo-Croatian.

Description of work

Work package leader: SYS
Task 6.1 Multilingual access development (m0-m12)
Task leader: SYS
Task participants: NKP, AIP

The project will provide two types of multilingual access:

- via the API integration in the data retrieval interface associated or independent of a multilingual search.
- a dedicated translation interface where ENRICH expert users can fine-tune dynamically the machine translation tools thanks to adapted linguistic tools for terminology extraction and translation post-editing and customization. The parameters and resources constructed will be automatically taken into account by the API in the access presented above

Task 6.2 Translation Stylesheet design and use (m3-m24)
Task leader: SYS
Task participants: NKP, AIP, CCP, NFC, NLF, IMI, ULV, SAM, CSH, DSP, NLI, BNE, BUTE, ULW
Activities:
• analysis of heterogeneity of metadata regarding machine translation.

D-6.3 ENRICH Translation Stylesheet delivery report

• implementation of STS exploiting metadata information.
• cross-language validation of STS, optimization of translation parameters.

As far as the Metadata translation module implementation is concerned SYSTRAN will provide a fully customized Translation Stylesheet.

SYSTRAN Translation Stylesheets (STS) use XSLT to drive and control the machine translation of XML documents (native XML document formats or XML representations — such as XLIFF — of other kinds of document formats). STS will provide a simple way to indicate which part of the document text is to be translated, and will enable the fine-tuning of translation, especially by using the structure of the document to help disambiguate natural language semantics and determine proper context. Thanks to STS machine translation is considered as part of the authoring and publishing process: source documents can be annotated with natural language mark-up produced by the author, a mark-up which will be processed by STS to improve the quality of translation, the gateway to the automatic publishing of a multilingual website from a monolingual (annotated) source. The mechanism is implemented through XSLT extension functions for consulting and for setting linguistics options in the translation engine. SYSTRAN will deliver this xslt file in order to fine-tune the system according to the ENRICH xml data elements.

Task 6.3 VICODI implementation (m6-m24)
Task leader: SYS
Task participants: NKP, AIP,

• definition and homogenization of initial ontology applicable for this project
• specification of user-friendly web-interface for visualization of multilingual ontology - special
interface for modification
• implementation of the web-interface

Based on previous experience in the visualization and contextualization of digital content (IST project VICODI) SYSTRAN technology has been implemented for the construction of multilingual ontologies. The Research Center for Information Technologies (FZI) constructed multilingual ontologies available under GNU Free Documentation License (FDL) thanks to the EU-funded IST project Vicodi http://www.vicodi.org/). Enrich will implement and use VICODI ontologies for the contextualization of the digital content.

(Inter-) Dependencies, milestones[1] and expected result

Based on the ENRICH Corpus Analysis (Month 6) SYSTRAN will build ENRICH Translation Stylesheet.
After a quality assessment procedure and based on the evaluation results (Month 20) SYSTRAN will proceed to the finalisation of ENRICH Translation Stylesheet (Month 24).
The WP depends mainly on the results of WP3, but is also interrelated with WP4.
The feedback is necessary from WP7.

Deliverables

D 6.1 ENRICH Corpus Analysis report (Month 6), responsible partner: SYS
D 6.2 Personalised Translation Interface delivery report (Month 12), responsible partner: SYS
D 6.3 ENRICH Translation Stylesheet delivery report (Month 24), responsible partner: SYS
D 6.4 Vicodi Ontologies implementation report (Month 24), responsible partner: SYS

2   General Presentation

The SYSTRAN Translation Stylesheet (STS) can be defined, improved and maintained by the translation user. It enables the translation of XML documents via the Extensible Stylesheet Language (XSL) by describing the method for translating a given XML input document.

Unlike the translation of ordinary documents (e.g., HTML or Word documents) in which translation options are set for the whole document, using STS it is possible to fine-tune the translation to address individual XML elements and thus obtain significantly better translations.

The Extensible Stylesheet Language (XSL) is a powerful programming language designed to manipulate and transform XML documents. In XSL, a set of transformations is expressed as a set of rules describing individual transformations, whereas a rule is composed of a pattern to be matched in the input XML document and the target structure to be generated. Originally intended for formatting XML document, XSL can be used to manipulate XML documents and implement transformations that go beyond formatting.

In particular, using the SYSTRAN Translation Stylesheet it is possible to:

- Use any translation option available for each rule;
- Distinguish between XML elements that require translation and those that do not;
- Distinguish between document style elements (e.g., titles, headings, alerts, normal text, etc.) and set translation options specific to each;
- Designate source document tags for restoration in the associated target documents;
- Generate documents that interweave source and target sentences (i.e., side-by-side
- presentation);
- Integrate document markup in either the source or target document (i.e., User Dictionary markup, Not Found Word markup);
- Leverage natural XSLT capabilities (i.e., reformatting of source and target text: stylesheet formatting, HTML conversion, text format projection, etc.)

XML documents contain a mix of language elements, data elements, and document structure elements. The SYSTRAN Translation Stylesheet (STS) preserves the document structure, selectively translating elements that contain textual data while leaving data elements unchanged. Each STS is associated to a particular DTD and is used to translate all documents conforming to that DTD. Several STS can be designed for a given DTD; each designed to perform translation within different parameters and/or to reformat the result in a different manner.

Four versions of ENRICH Translation Stylesheet have been released during the project based on the different DTD utilized for the integration and evaluation tests.

STS Processing Overview

The standard use of an STS is to perform simple translations of XML documents while preserving data and structural elements in the following action sequence:

1. Validate the source document using the document DTD.
2. Traverse the XML tree, possibly updating local translation options.
3. Extract Translation Units (TUs), the selection of which is accomplished via XSLT rules. A Translation Unit (TU) can be an XML element, the value of a string attribute containing textual data, or even the content of an XML comment.
4. Pre-process the Translation Units: Convert all markup within a Translation Unit into markup that can be understood by the SYSTRAN Translation Engine.
5. Translate the pre-processed Translation Units: Send the TUs to the translation server,
complete with selected options.
6. Post-process the translated TUs: Restore typesetting and DNT markup in the translated
TUs.
7. Rebuild the target XML document.

XML Validation

The initial STS processing step is the validation of the source document structure. The document's DTD is retrieved using the "system" or the "public" reference to it, and the public reference is resolved – when possible – via a local XML catalog.

What is a Translation Unit?

A Translation Unit (TU) is the smallest logical unit that can be sent to the Translation Engine, and thus is also the smallest XML element containing language text to be translated. The content of these XML elements is sent to the translation server as a single block of text, and may contain titles, single sentences, list items or whole paragraphs of text. The translation engine segments Translation Units into "sentences". In cases where the XML structure already segments the text into sentences, these sentences can be sent directly to the translation engine, bypassing the translation engine's sentence segmentation process, provided a special translation option is used to specify that the TUs are already segmented. If the textual content of an XML document is marked using XML elements that make language-relevant distinctions, it is possible to associate different translation options to these different XML elements.

Further XML markup is possible within Translation Units, such as that employed for formatting purposes. To assure correct restoration in the translated textual unit, this supplementary markup must be converted into SYSTRAN markup. Any markup within a TU that is not converted during pre-processing will be ignored by the translation server.

Markup within a Translation Unit

Typesetting markup

Typesetting markup includes markup in document formatting – bold text, underlined text, etc. – which behaves in a particular way during the translation process (i.e., typesetting markup needs to be re-attached to translated words, even if their position has changed during translation). An example is seen in the following XML fragment, in which the **** element must move with the word **x23**.

<par>switch on the <abbrev>x23</abbrev> router</par>
<par>Allumer le routeur x23</par>

Note that in some cases typesetting blocks must be split if the translation of the block is split in the target sentence.

These elements will need to be translated to SYSTRAN typesetting elements during the Preprocessing phase.

Using XSLT, it is possible to select the text content of an XML comment – and to thus translate the comments – however, to do so an explicit rule must be invoked to select and translate the comments (by default, unselected elements of the XML document remain untranslated and are preserved as-is in the target XML document).

Elements that should not be translated

Do-Not-Translate (DNT) elements block the translation of designated textual sequences within a Translation Unit. An XML element is a DNT block that denotes a word sequence that is a genuine constituent of the sentence, and which may have a close relationship to some specific words in that sentence. These elements will need to be translated to SYSTRAN DNT elements during the preprocessing phase.

Sub-Flows

The **<comment>** element is used to describe looser textual insertions in a sentence, or subflows, which should remain inside the main sentence for translation but which are not attached to specific words within the sentence. These parenthetical segments can be translated (or mapped to DNTs) independently from the translation of the main sentence.

Functions

The SYSTRAN XSLT extension enables calls to the translation engine from within an XSL stylesheet, providing functions defined in the SYSTRAN namespace – translate, getValue, pushValue and popValue – with the following prototypes:

Functions Defined	Associated Prototype Functions	Description
translate	node-set translate(node-set, listOfOptions)	Calls the translator with the content of the XML node as specified by NodeSet and returns the translation of the content.
getValue	string getValue(string)	Queries the value of the translation engine options, be they general options (e.g., current source and target languages), global linguistic options (e.g, whether "you" in English should be translated as informal ""tu"" or polite "vous" in French), or even stylesheet-specific options set by the user.
pushValue	void pushValue(string, string)	Sends local options to the translation engine (e.g, requiring that every noun/verb ambiguity should be resolved as noun inside a particular tag).
popValue	void popValue	Mechanism that allows for a very general management of the span life of options sent to the translation engine; popValue is the counterpart of pushValue: a stack of options is maintained in the translation engine; the latest element added to the stack is removed on the call of popValue.

Input
The input to the translate function is a NodeSet with translation options specific to it.

NodeSet
A NodeSet is a set of par (for *paragraph*) trees that verifies the following DTD (attributes are not shown):

```
<!ELEMENT par (#PCDATA|mark|typo|tag)*>
<!ELEMENT mark (#PCDATA|mark|typo|tag)*>
<!ELEMENT typo (#PCDATA|mark|typo|tag)*>
<!ELEMENT tag ANY>
```

- mark tags are used to provide additional local information (in input) to translation engines, or to get additional feedbask (in output) from them.
- typo tags are used to indicate local character properties (e.g. bold font, hyperlink…) to ensure that these properties are reinserted in the translation result, at the proper location (subsequent to any necessary reordering of words).
- Everything inside a tag tag remains as-is in the translation result.
- All text content that is not inside a tag node is translated.

Translation options

All translation options are available to the translator via the network protocol. Typically, changing translation options allows for dynamic change of the *domain* of a section, the disabling of translation for uppercase words in titles, etc.

Output
The output of the Translate function is a `NodeSet`.

Writing a Translation Stylesheet using Utility Templates
Utility templates are defined in the tools.xsl and markup.xsl files for the purpose of expediting the writing of SYSTRAN Translation Stylesheets. Additionally, tools.xsl defines the templates needed in order to use the approach illustrated here for writing STS stylesheets: importing tools.xsl and markup.xsl and benefiting from the natural cascading mechanism of XSLT.

- By default, nodes are browsed in document order and textual nodes are preserved and not translated.

- Nodes whose direct content should be translated must call a `translate_par` template (or one of its variants).

- Inline elements (those which are descendants of nodes whose content will undergo translation) must define templates with the *preprocess* mode, which should produce `mark`, `typo`; tag if some processing needs to be done to recover original structures. The following elements must be included in the SYSTRAN Translation Stylesheet header:

```
<xsl:stylesheet version="1.0"
xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
<xsl:import href="tools.xsl"/>
<xsl:import href="markup.xsl"/>
```

Translating XML comments
Using XSLT, it is possible to select the text content of an XML comment – and to thus translate the comments – however, to do so an explicit rule must be invoked to select and translate the comments (by default, unselected elements of the XML document remain untranslated and are preserved as-is in the target XML document).

Preprocessing
Preprocessing is arguably the most critical step of the STS translation process. By default, the translation engine has no context regarding the XML elements used as markup within Translation Units, and as such it is necessary to convert these XML elements into a markup that is meaningful to the translation engine.
A pertinent preprocessing example is markup employed for formatting purposes. To undergo translation and eventual restoration in the target text, such markup must be mapped to SYSTRAN's internal markup.

D-6.3 ENRICH Translation Stylesheet delivery report

Postprocessing
In postprocessing, all SYSTRAN nodes are removed and the nodes are mapped back to the original format node as necessary.

STS Catalog
STSs are located in the xml directory and are referenced via the sts_catalog.xml file.

Each STS is associated to a particular DTD and is used to translate all documents conforming to that DTD. The `sts_catalog.xml` file – which is used to store this information – follows the structure defined in the STSCatalog.xml DTD located in the same directory. The three consecutive releases of ENRICH Translation Stylesheet were associated to three DTDs ceec.dtd provided by University of Köln (UZK), the dtd provided by NULI and the dtd provided Manuscriptorium (samples can be consulted in Annex A).

Options and translation profile mechanism
In the XSL file, we can pass some options to the translator:

```
Options="key1=value1,key2=value2"
<!-- Systran utility templates -->
<xsl:template name="translate_internal">
<xsl:param name="source"/>
<xsl:param name="options"/>
<xsl:param name="status"/>
<xsl:param name="internal" select="false()"/>
<xsl:if test="systran:setStatus($status)=0">
<xsl:message terminate="yes"/>
</xsl:if>
<xsl:variable name="par_source">
<par>
<xsl:if test="$internal">
<xsl:attribute name="internal">1</xsl:attribute>
</xsl:if>
<xsl:copy-of select="$source"/>
</par>
</xsl:variable>
<xsl:copy-of select="systran:translate($par_source,$options)"/>
</xsl:template>
```

Extension to a profile mechanism in the STS process
We have the possibility to pass an ENRICH-specific translation profile to the `systran:translate` method.

We have 4 different levels of user profiles (from the lower to the higher priority):

- On the translator side, with the default user profile located in the DICT directory, for instance *dict/profile_enrich_ENFR.xml*. This user profile is read when the translator is started.
- On the translator side, with the option *--set user_profile=path* passed to the SystranTranslationEngine.
- On the application side (SYSTRAN Enterprise Server 7 light client (http://enrich.systran.fr/), a default user profile is passed to the translator in each SOAP queries.
- On the application side (Manuscriptorium), the end user can change the default profile passed to the translator in each SOAP queries.

In the STS file, we will be able to pass an user profile to the translator as an argument of the `Systran:translate` method for each paragraph. This user profile will have the highest priority.

3   Implementations achieved

University of Köln (UZK) contributed to the first implementation tests of Enrich Translation Stylesheet in their workflow with two data-sets for harvesting:

_ Codices Electronici Ecclesiae Coloniensis (CEEC)
http://www.ceec.uni-koeln.de/

and

_ Verteilte Digitale Inkunabelbibliothek (vdIb)
http://www.ub.uni-koeln.de/bibliothek/projekte/vdib/index_ger.html

For development and testing purposes Systran Enterprise Server 6 has been set-up at enrich.hki.uni-koeln.de. The local installation is living at enrich:/usr/local/systran/. The Systran server provides a public user interface at http://enrich.hki.uni-koeln.de:8080/ and a SOAP Web Service endpoint (calling the API) at http://enrich.hki.uni-koeln.de:8080/ws.

Two Translation Stylesheets (STS) have been developed for CEEC and vdIb. For testing purposes they have been registered with the local Systran server installation. The main working directory for all described actions is `enrich:/usr/local/systran/jboss-4.0.5/server/default/deploy/systran.war/engines/dict/xml/`.

_ Each XML document to be translated needed a DTD declaration:

```
<!DOCTYPE CEC SYSTEM "[ceec|vdIb].dtd">
```

_ Both the stylesheet (`[ceec|vdIb].xsl` provided by Systran) and the DTD
(`ceec| vdIb].dtd`) were copied to the working directory.

_ The DTD needs to be registered in catalog.xml:

```
<system systemId="[ceec|vdIb].dtd" uri="[ceec|vdIb].dtd"/>
```

_ The STS was registered in sts_catalog.xml:

```
<application stylesheet="[enrich|vdIb].xsl" version="1.0"
mimetype="application/x-[enrich|vdIb]+xml">
<root>CEC</root>
</application>
```

Additionally, a Java program has been developed to enable bulk processing of CEEC and vdIb XML files. This program sequentially reads all files within a certain directory and uses the SOAP endpoint for translation.
It can be downloaded at `svn://enrich.hki.uni-koeln.de/fileTranslator`:

```
svn --username enrich
    --password enrich
svn://enrich.hki.uni-koeln.de/trunk/fileTranslator/
```

Example of translation configuration applied for CEEC & vdIb needs

All authorities except of `Anderson' will be translated from German to English:

```
//*[@authority!='Anderson']
```

Some XML tags have been excluded from translation based on the corpus analysis (Deliverable 6.1) and the content partners feedback:

```
//bibl
//explicit
//implicit
//locus
//msContents
//msIdentifier
//ref
//title
```

Since some authorities mix European and Latin text, Systran server has been instructed to recognize foreign language sentences based on the Not Found Words (NFW) count by setting an option within the stylesheet (an option needs to be transmitted with each translation request). Example for a translation request from German into English

```
…
Request request = new Request();
option = new Option[3];
option[0] = new Option("lp", "de_en");
option[1] = new Option("input_file_data", inputFileData);
option[2] = new Option("translate_source", "file_data");
request.setOption(option);
Response res;
// call the web service
try {
res = tws.translate(request);
} catch (systran.soapapi.service.Error e) {
System.out.println("Error (service call): " + e.toString());
return null;
} catch (java.rmi.RemoteException e) {
System.out.println("Error (remote): " + e.toString());
return null;
}
…
```

D-6.3 ENRICH Translation Stylesheet delivery report

The stylesheet that has been used for translation based on the above requirements is formalized as follows

```xml
<?xml version="1.0" encoding="UTF-8"?>
<xsl:stylesheet version="1.0"
xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
xmlns:date="http://exslt.org/dates-and-times"
xmlns:systran="http://www.systransoft.com/xslfunctions"
xmlns:sysel="http://www.systransoft.com/xslelements"
xmlns:exsl="http://exslt.org/common"
extension-element-prefixes="date systran"
exclude-result-prefixes="date systran sysel">
<xsl:import href="tools.xsl"/>
<xsl:import href="markup.xsl"/>
<xsl:output encoding="UTF-8" indent="yes"/>
<xsl:template match="*" mode="detect-type">element</xsl:template>
<xsl:template match="text()" mode="detect-type">text</xsl:template>
<xsl:template
match="bibl|explicit|implicit|locus|msContents|msIdentifier|ref"
mode="dnt">1</xsl:template>
<xsl:template name="enrichTranslate">
<xsl:param name="source" />
<xsl:param name="options" />
<xsl:for-each select="$source">
<xsl:variable name="type">
<xsl:apply-templates mode="detect-type" select="." />
</xsl:variable>
<xsl:choose>
<xsl:when test="$type='text'">
<xsl:call-template name="translate_text">
<xsl:with-param name="source" select="."/>
<xsl:with-param name="options" select="$options" />
</xsl:call-template>
</xsl:when>
<xsl:when test="$type='element'">
<xsl:variable name="dnt">
<xsl:apply-templates mode="dnt" select="." />
</xsl:variable>
<xsl:choose>
<xsl:when test="$dnt=1">
<xsl:copy-of select="."/>
</xsl:when>
<xsl:otherwise>
<xsl:variable name="translation">
<xsl:element name="{name()}">
<xsl:apply-templates select="@*" />
<xsl:for-each select="node()">
<xsl:call-template name="enrichTranslate">
<xsl:with-param name="source" select="." />
<xsl:with-param name="options" select="$options" />
</xsl:call-template>
</xsl:for-each>
</xsl:element>
</xsl:variable>
<xsl:copy-of select="$translation"/>
</xsl:otherwise>
</xsl:choose>
</xsl:when>
<xsl:otherwise>
<xsl:copy-of select="."/>
</xsl:otherwise>
</xsl:choose>
</xsl:for-each>
</xsl:template>
<xsl:template match="//*[@authority='Anderson']">
<xsl:copy>
<xsl:apply-templates select="@*" />
<xsl:call-template name="enrichTranslate">
<xsl:with-param name="source" select="node()" />
<xsl:with-param name="options" select="'max_nfw_percentage=1'" />
</xsl:call-template>
</xsl:copy>
</xsl:template>
</xsl:stylesheet>
```

D-6.3 ENRICH Translation Stylesheet delivery report

For vdIb translation needs the stylesheet has been slightly modified:

```
<?xml version="1.0" encoding="UTF-8"?>
<xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
xmlns:date="http://exslt.org/dates-and-times"
xmlns:systran="http://www.systransoft.com/xslfunctions"
xmlns:sysel="http://www.systransoft.com/xslelements" xmlns:exsl="http://exslt.org/common"
extension-element-prefixes="date systran" exclude-result-prefixes="date systran sysel">
...
<xsl:template match="//text']">
<xsl:copy>
<xsl:apply-templates select="@*" />
<xsl:call-template name="enrichTranslate">
<xsl:with-param name="source" select="node()" />
<xsl:with-param name="options" select="'max_nfw_percentage=1'" />
</xsl:call-template>
</xsl:copy>
</xsl:template>
...
</xsl:stylesheet>
```

Translation example for CEEC

English text:
```
…
<history authority="AndersonTest">
<origin>
<p>"Codex sancti petri scriptus sub pio patre hildebaldo archiepiscopo" (<locus>f. 1r</locus>
-
<person>Hildebald Abp. of K&ouml;ln 786-819</person>).</p>
<p>annales Colonienses in Paschal table <locus>f. 76v</locus>
</p>
<p>list of names, mentioning also Hildebald and Charlemagne, <locus>f. 219v</locus>; see also
Calendarium additions <locus>ff. 72v-76r</locus>.</p>
<p>This manuscript is a collection of texts from Irish and Anglo-Saxon schools, according to
<bibl>W. Stevens, "Sidereal Time...", loc. cit.</bibl> This is one of the four manuscripts
which Bischoff named as being among those which without any doubt originated at K&ouml;ln
(<bibl>"Panorama...", loc. cit.</bibl> ); this ms. is probably also listed in the <ref
type="altMs">K&ouml;ln catalogue of 833</ref>. Evidence for origin at K&ouml;ln
includes owner marks <locus>ff. 1r and 219v</locus> (mention of Hildebald and Charlemagne),
Calendarium <locus>ff. 72v-76r</locus>, and notes added in paschal tables <locus>f.
76v</locus>. Uniform page layout, similar rubrication, and quire signatures written by text
hands indicate that the codex was intended as a unified book, although built up from booklets
of various texts, apparently begun in <date>798</date> and completed in <date>805</date>;
<locus>ff. 2-14</locus> (quires i-ii) may be the earliest part of the codex, where text on
<locus>14v</locus> calculates the current date as <date>A.D. 798</date>, which is also 1ˢᵗ year
of paschal table <locus>f. 76v</locus>; or this ms. may be copied from an exemplar dating from
<date>798</date>; date <date>805</date> given as current year <locus>ff. 55r and 76r</locus>.
Quires xxi-xxiv form a booklet with distinctively more elaborate decoration; elsewhere there
are blank pages or half-pages at ends of some quires; changes of hand usually occur between
quires. As noted by <bibl>C.W. Jones, <title>Bedae Pseudepigrapha</title> , p. 115</bibl> ,
this ms. is "one of the most important of the early computistical manuscripts in every
respect, but the texts are carelessly copied." The many errors remain, for the most part,
uncorrected. Not identifiable with <bibl>Decker, nr. 36</bibl> ("Item Bedae de temporibus),
which could be <ref type="altMs">Hs. 103</ref>, unless <ref type="altMs">Hs. 103</ref> is
<bibl>Decker's nr. 92</bibl> ("Compotum Bedae vol. I")(see <bibl>Frenken, loc. cit.</bibl> );
Hs. 83-II should be identified with <bibl>Decker's nr. 91</bibl> .</p>
</origin>
<provenance>Darmstadt library stamp, <locus>f. 1v</locus>
</provenance>
</history>
…
```

Translation to German:
```
..
<history authority="AndersonTest">
<origin>
<p>"Codex sancti petri scriptus sub pio patre hildebaldo archiepiscopo" ( <locus>f.
1r</locus>-
<person>Hildebald Abp. von K&#195;&#182;ln 786-819</person>).</p>
<p>annales Colonienses in Paschal table <locus>f. 76v</locus></p>
<p>list of names, mentioning also Hildebald and Charlemagne, <locus>f. 219v</locus>; see also
Calendarium additions <locus>ff. 72v-76r</locus>.</p>
<p>Dieses Manuskript ist eine Ansammlung Texte von den irischen und
angels&#195;&#164;chsischen Schulen, entsprechend <bibl>W. Stevens, "Sidereal
Time...", loc. cit.</bibl>This is one of the four manuscripts which Bischoff named as being
among those which without any doubt originated at K&#195;&#182;ln (
<bibl>"Panorama...", loc. cit.</bibl>); diese Frau wird vermutlich auch in verzeichnet <ref
type="altMs">K&#195;&#182;ln catalogue of 833</ref>. Evidence for origin at
K&#195;&#182;ln includes owner marks <locus>ff. 1r and 219v</locus>(mention of
Hildebald and Charlemagne), Calendarium <locus>ff. 72v-76r</locus>und Anmerkungen in den
paschal Tabellen hinzugef&#195;&#188;gt <locus>f. 76v</locus>. Uniform page layout,
similar rubrication, and quire signatures written by text hands indicate that the codex was
intended as a unified book, although built up from booklets of various texts, apparently begun
in <date>798</date>und innen abgeschlossen <date>805</date>; <locus>ff. 2-14</locus>(Quires
III) sein kann das fr&#195;&#188;heste Teil des Kodexes, in dem Text an
<locus>14v</locus>berechnet das Tagesdatum wie <date>A.D. 798</date>, das auch 1. Jahr der
paschal Tabelle <locus>f. 76v</locus>; oder diese Frau kann von einem
herr&#195;&#188;hrend vonen Exemplar kopiert werden <date>798</date>; Datum
<date>805</date>gegeben als laufendes Jahr <locus>ff. 55r and 76r</locus>. Quires xxi-xxiv
form a booklet with distinctively more elaborate decoration; anderwohin gibt es Leerseiten
oder Halbseiten an den Enden einiger Quires; &#195;&#8222;nderungen der Hand treten
normalerweise zwischen Quires ein. Wie vorbei gemerkt <bibl>C.W. Jones, <title>Bedae
Pseudepigrapha</title>, p. 115</bibl>, this ms. is "one of the most important of the early
computistical manuscripts in every respect, but the texts are carelessly copied." Die vielen
St&#195;&#182;rungen bleiben in den meisten F&#195;&#164;llen unkorrigiert.
Nicht identifizierbar mit <bibl>Decker, nr. 36</bibl>("Item Bedae de temporibus), which could
be <ref type="altMs">Hs. 103</ref>, es sei denn <ref type="altMs">Hs. 103</ref>ist
<bibl>Decker's nr. 92</bibl>("Compotum Bedae vol.. I") (sehen <bibl>Frenken, loc.
cit.</bibl>); HS. 83-II should be identified with <bibl>Decker's nr. 91</bibl>.</p>
</origin>
<provenance>Darmstadt-Bibliotheksstempel, <locus>f. 1v</locus></provenance>
```

```
</history>
```

All translation examples can be found as a bundle together with the Java code mentioned above.
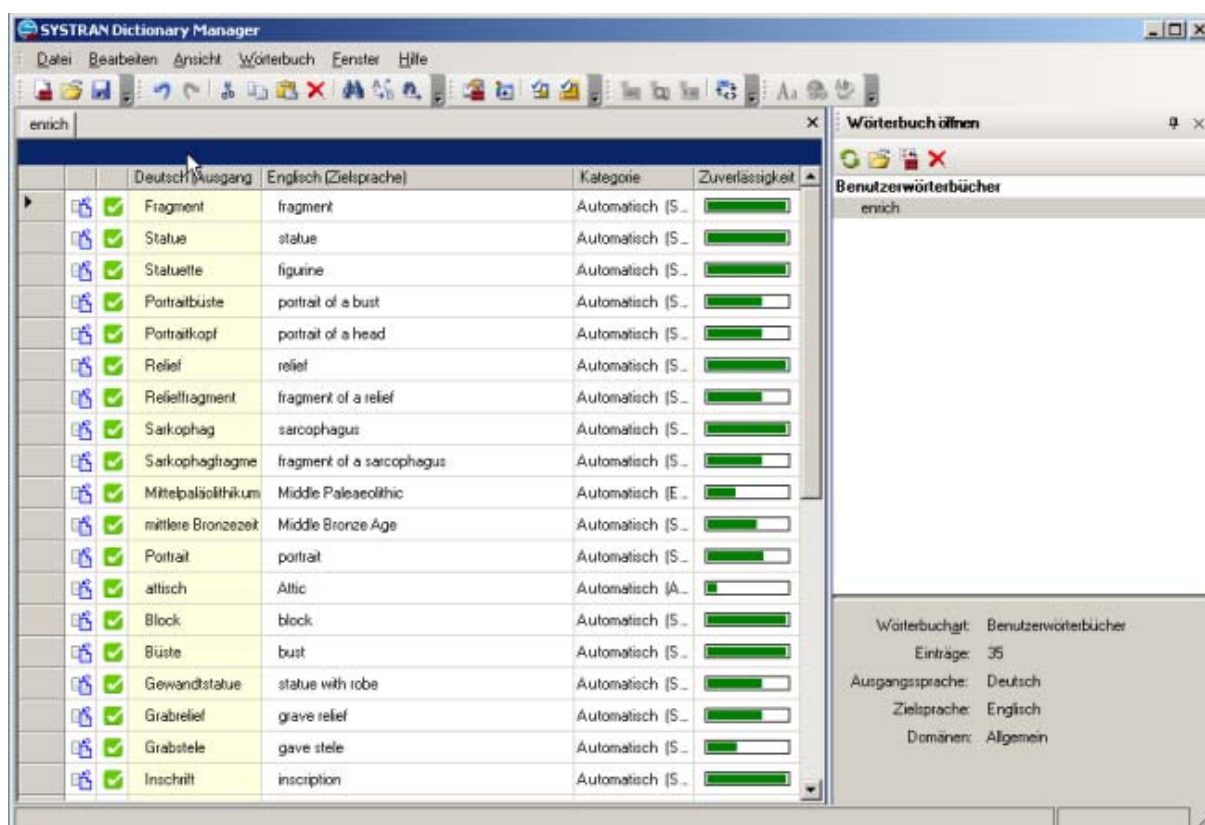
After crafting the stylesheets and tweaking the Systran translation options, the translation quality cannot be further enhanced. A banner has been integrated in order to notify the users that it is a machine translation result

```
<xsl:template match="//*[@authority!='Anderson']">
   <xsl:copy-of select="." />
   <xsl:copy>
     <xsl:apply-templates select="@*" />
     <xsl:attribute name="note">
        <xsl:text>Translated to </xsl:text>
        <xsl:value-of select="$tgtlng"/>
        <xsl:text> by SYSTRAN</xsl:text>
     </xsl:attribute>
     <xsl:call-template name="enrichTranslate">
       <xsl:with-param name="source" select="node()" />
       <xsl:with-param name="options" select="'max_nfw_percentage=20'" />
     </xsl:call-template>
   </xsl:copy>
 </xsl:template>
```

generating

```
<generalDescription authority="Glaube und Wissen" auth-range="196" note="Translated to en by
SYSTRAN">
```

In addition to this a domain dictionary for paleography had been started. A first version of paleography dictionary was thus built using SYSTRAN Dictionary Manager, the heavy client application that allows to the users to build customization dictionaries associated to the translation profiles.

D-6.3 ENRICH Translation Stylesheet delivery report

UZK contributed also by adjusting the Enrich-specific Translation Stylesheet for metadata that follows a modified ebind-Format. Since there was no DTD for that, oXygen was used for a generation:

```xml
<?xml version="1.0" encoding="UTF-8"?>
<xsl:stylesheet version="1.0"
  xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
  xmlns:date="http://exslt.org/dates-and-times"
  xmlns:systran="http://www.systransoft.com/xslfunctions"
  xmlns:sysel="http://www.systransoft.com/xslelements"
  xmlns:exsl="http://exslt.org/common"
  extension-element-prefixes="date systran"
  exclude-result-prefixes="date systran sysel">

  <xsl:import href="tools.xsl"/>
  <xsl:import href="markup.xsl"/>

  <xsl:output encoding="UTF-8" indent="yes"/>

  <xsl:template match="*" mode="detect-type">element</xsl:template>
  <xsl:template match="text()" mode="detect-type">text</xsl:template>

  <xsl:template              match="bibl|explicit|implicit|locus|msContents|msIdentifier|ref"
mode="dnt">1</xsl:template>

  <xsl:template name="enrichTranslate">
    <xsl:param name="source" />
    <xsl:param name="options" />

    <xsl:for-each select="$source">

      <xsl:variable name="type">
        <xsl:apply-templates mode="detect-type" select="." />
      </xsl:variable>

      <xsl:choose>
        <xsl:when test="$type='text'">
          <xsl:call-template name="translate_text">
            <xsl:with-param name="source" select="."/>
            <xsl:with-param name="options" select="$options" />
          </xsl:call-template>
        </xsl:when>

        <xsl:when test="$type='element'">
          <xsl:variable name="dnt">
            <xsl:apply-templates mode="dnt" select="." />
          </xsl:variable>

          <xsl:choose>
            <xsl:when test="$dnt=1">
              <xsl:copy-of select="."/>
            </xsl:when>

            <xsl:otherwise>
              <xsl:variable name="translation">
                <xsl:element name="{name()}">
                  <xsl:apply-templates select="@*" />
                  <xsl:for-each select="node()">
                    <xsl:call-template name="enrichTranslate">
                      <xsl:with-param name="source" select="." />
                      <xsl:with-param name="options" select="$options" />
                    </xsl:call-template>
                  </xsl:for-each>
                </xsl:element>
              </xsl:variable>
              <xsl:copy-of select="$translation"/>
            </xsl:otherwise>
          </xsl:choose>
        </xsl:when>
        <xsl:otherwise>
          <xsl:copy-of select="."/>
        </xsl:otherwise>
      </xsl:choose>
    </xsl:for-each>
  </xsl:template>

  <xsl:template match="//*[@authority='Anderson']">
    <xsl:copy>
```

```
      <xsl:apply-templates select="@*" />
      <xsl:call-template name="enrichTranslate">
        <xsl:with-param name="source" select="node()" />
        <xsl:with-param name="options" select="'max_nfw_percentage=1'" />
      </xsl:call-template>
    </xsl:copy>
  </xsl:template>

</xsl:stylesheet>
```

Example of translation configuration applied for NULI needs

AiPBeroun modified the ENRICH Translation Stylesheet for NULI translation needs following the same methodology. The modifications made consisted in the change of the Do Not Translate (DNT) set of information. The DNT elements left after all elements explicitly required for translation by the partners and the restriction required explicitly by OUCS. The `msContents` was set back into translation and named explicitly the elements from `msContents` which obviously should not be translated (colophon for instance).

```
<?xml version="1.0" encoding="UTF-8" ?>
    <xsl:stylesheet        version="1.0"        xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
  xmlns:date="http://exslt.org/dates-and-times"
  xmlns:systran="http://www.systransoft.com/xslfunctions"
  xmlns:sysel="http://www.systransoft.com/xslelements"        xmlns:exsl="http://exslt.org/common"
  extension-element-prefixes="date systran" exclude-result-prefixes="date systran sysel">
<xsl:import href="tools.xsl" />
<xsl:import href="markup.xsl" />
<xsl:output encoding="UTF-8" indent="yes" />
<xsl:template match="*" mode="detect-type">element</xsl:template>
<xsl:template match="text()" mode="detect-type">text</xsl:template>
    <!--
     replaced by TP/AIP
     <xsl:template
     match="bibl|explicit|implicit|locus|msContents|msIdentifier|ref"
     mode="dnt">1</xsl:template>

     dnt added according to D6.1:
     facsimile
     term[ancestor::desc]

     name[@type='person']
     persName
     author
     docAuthor
     docImprint
     docTitle
     colophon

     dnt according to D6.1 not added yet:
     gloss[@xml:lang]
     desc[@xml:lang]

     dnt removed (may be discussed of course!):
     msContents

  -->
                                                                <xsl:template
  match="bibl|explicit|implicit|locus|msIdentifier|ref|facsimile|name[@type='person']|persName|a
  uthor|docAuthor|docImprint|docTitle|colophon|term[ancestor::desc]" mode="dnt">1</xsl:template>
<xsl:template name="enrichTranslate">
<xsl:param name="source" />
<xsl:param name="options" />
<xsl:for-each select="$source">
<xsl:variable name="type">
<xsl:apply-templates mode="detect-type" select="." />
   </xsl:variable>
<xsl:choose>
<xsl:when test="$type='text'">
<xsl:call-template name="translate_text">
<xsl:with-param name="source" select="." />
<xsl:with-param name="options" select="$options" />
   </xsl:call-template>
   </xsl:when>
<xsl:when test="$type='element'">
```

```xsl
<xsl:variable name="dnt">
<xsl:apply-templates mode="dnt" select="." />
   </xsl:variable>
<xsl:choose>
<xsl:when test="$dnt=1">
<xsl:copy-of select="." />
   </xsl:when>
<xsl:otherwise>
<xsl:variable name="translation">
<xsl:element name="{name()}">
<xsl:apply-templates select="@*" />
<xsl:for-each select="node()">
<xsl:call-template name="enrichTranslate">
<xsl:with-param name="source" select="." />
<xsl:with-param name="options" select="$options" />
   </xsl:call-template>
   </xsl:for-each>
   </xsl:element>
   </xsl:variable>
<xsl:copy-of select="$translation" />
   </xsl:otherwise>
   </xsl:choose>
   </xsl:when>
<xsl:otherwise>
<xsl:copy-of select="." />
   </xsl:otherwise>
   </xsl:choose>
   </xsl:for-each>
   </xsl:template>
<xsl:template match="//*[@authority!='Anderson']">
<xsl:copy-of select="." />
<xsl:copy>
<xsl:apply-templates select="@*" />
<xsl:attribute name="note">
<xsl:text>Translated by SYSTRAN</xsl:text>
   </xsl:attribute>
<xsl:call-template name="enrichTranslate">
<xsl:with-param name="source" select="node()" />
<xsl:with-param name="options" select="'max_nfw_percentage=20'" />
   </xsl:call-template>
   </xsl:copy>
   </xsl:template>
   </xsl:stylesheet>
```

Following the implementation into Manuscriptorium the TEI STS has been added in the sts_catalog.xml

```
sts_catalog
  application
    dtd_public_id -//W3C//DTD XHTML 1.0 Strict//EN dtd_public_id
    dtd_public_id -//W3C//DTD XHTML 1.0 Transitional//EN dtd_public_id
    dtd_public_id -//W3C//DTD XHTML 1.0 Frameset//EN dtd_public_id
    dtd_public_id -//W3C//DTD XHTML 1.1//EN dtd_public_id
    namespace http://www.w3.org/1999/xhtml namespace
  application
  application
    namespace urn:oasis:names:tc:xliff:document:1.1 namespace
    root xliff root
  application
  application
    dtd_public_id -//LISA OSCAR:1998//DTD for Translation Memory eXchange. dtd_public_id
    root tmx root
    options
      option_group
        description Formatting and filters description
        option_group
          description Translation Memory eXchange STS description
          description STS Translation Memory eXchange. description
          option
            name TMX_PHASE name
            description TMX processing phase description
            description Phase de traitement du TMX description
          option
        option_group
      option_group
    options
  application
  application
    dtd_public_id -//SYSTRAN//DTD XML Internal Format 1.0//EN dtd_public_id
  application
  application
    root CEC root
    root TEI root
  application
sts_catalog
```

4 Conclusions

Based on the feedback from the content partners SYSTRAN constructed the first version of the SYSTRAN Translation Stylesheet according to the content partners' requirements. That feedback allowed SYSTRAN to analyze the ENRICH content and tailor the translation module according to the current requirements. Based on content partners' feedback about the elements to be translated and evaluation tests, SYSTRAN Translation Stylesheet has been developed preventing side-effects related to the translation of homograph Latin words.

The first version of the SYSTRAN Translation Stylesheet has been modified and maintained by UZK, NULI & AiPBeroun at different steps of content processing according to the metadata description formalism utilized.

Enrich Translation Stylesheet developed by SYSTRAN is responding to a sophisticated tailoring of the translation options according to the metadata xml structure and can be used for different purposed as for instance the TEI documentation translation using another release of Enrich Translation Stylesheet modified by OUCS. It is easy to maintain and make modifications via the respective configuration files which are taken into account automatically by the application.

Enrich Translation Stylesheet can be also further enriched by domain-specific dictionaries in order to reach higher translation quality.

5 ANNEXES

- CEEC.dtd (provided by UZK)

```
<?xml version="1.0" encoding="UTF-8"?>
<!ENTITY % a.global "id ID #IMPLIED
               n CDATA #IMPLIED
               authority CDATA #IMPLIED
               auth-range CDATA #IMPLIED
               lang (la | en | de | fr | it) #IMPLIED
               rend CDATA #IMPLIED">
<!ENTITY % e.global "bibl | date | note | ref | term | title">
<!ENTITY % e.layout "lb | pb | sub | sup | img | displayScript">
<!ENTITY % e.measures "height | width | depth | format">
<!ENTITY % e.textIdentifier "incipit | explicit | colophon | initium | rubric">
<!ELEMENT CEEC (CEC+)>
<!ELEMENT CEC (TEIHeader, body?, addenda?, cover?)>
<!-- das Wurzelelement der xml-Dateien ist CEC; Veraenderungen an den Daten werden der
Einheitlichkeit
       willen allerdings an einer "Gesamt"-Datei vorgenommen, fuer die das Wurzelelment CEEC
       vorgesehen ist -->
<!ELEMENT TEIHeader (fileDesc, profileDesc?, revisionDesc?)>
<!ATTLIST TEIHeader
       CodexId CDATA #REQUIRED
       lang (de | la | en | fr) #REQUIRED
>
<!ELEMENT abbr (#PCDATA)>
<!ATTLIST abbr
       type (withLine) #REQUIRED
       expan CDATA #IMPLIED
>
<!-- im Head eines CEEC-XML-Files wird nur 'abbr' erlaubt mit dem Attribut 'expan', nicht aber
       umgekehrt auch 'expan' mit dem Attribut 'abbr' -->
<!ELEMENT accMat (#PCDATA | locus | ref)*>
<!ELEMENT acquisition (#PCDATA | person)*>
<!ATTLIST acquisition
       notBefore (1540 | 1861 | 1872) #REQUIRED
       notAfter (1540 | 1872) #IMPLIED
>
<!-- dies sind die in unseren Daten bisher tatsaechlich vorkommenden
       Werte. Sie stehen in der DTD, weil sie uns zur ueberpruefung der
       Konsistenz unserer Daten gedient haben. -->
<!ELEMENT addenda (text)>
<!ELEMENT additional (#PCDATA | accMat | date | person | bibl)*>
<!ATTLIST additional
       %a.global;
>
<!ELEMENT additions (#PCDATA | date | locus | title | term | orig | person | bibl | ref)*>
<!ELEMENT altName (#PCDATA | locus | sup | date | sub)*>
<!ATTLIST altName
       authority CDATA #REQUIRED
       auth-range CDATA #IMPLIED
       type (shelfmark | formerShelfmark | catNo | microfilmNo | nickname) #REQUIRED
>
<!ELEMENT author (#PCDATA | date | note)*>
<!ATTLIST author
       lang (de | en | it | la) #REQUIRED
       type (attributed | commentator | interpreter | translator) #IMPLIED
       reg CDATA #IMPLIED
>
<!-- die Werte fuer das Attribut 'type' sind diejenigen Funktionen eines Autors, die uns in
den
       Katalogen begegnet sind -->
<!-- reg ist fuer normalisierte (regularized) Form des Autornamens -->
<!ELEMENT availability (#PCDATA)>
<!ELEMENT bibl (#PCDATA | biblScope | author | title | locus | note | orig | sup | hi |
bibl)*>
<!ATTLIST bibl
       authority CDATA #IMPLIED
       auth-range CDATA #IMPLIED
       type (catalogue | edition | faksimile) #IMPLIED
       usage CDATA #IMPLIED
       biblTarget CDATA #IMPLIED
>
<!-- usage="short" zeigt an, daß die Aufloesung des (Kurz-)Titels anhand der Literatur-
```

```
        konkordanz notwendig ist -->


<!-- biblTarget wird verwendet, wenn in bibl keine Informationen zur Verfuegung stehen,
        die kleio (unsere Datenbank) zu einer Literaturverweis umbauen koennte.
        Der Wert von biblTarget sollte einer Kurz- oder Langangabe entsprechen,
        wie sie in der Literaturkonkordanz verzeichnet sind -->
<!ELEMENT biblScope (#PCDATA)>
<!ATTLIST biblScope
        value CDATA #IMPLIED
>
<!-- sollte der Inhalt von biblScope nicht formell aufzuloesen sein, kann hier eine Angabe
gemacht werden,
        die vom Rechner zur Bestimmung des bibliographischen Ziels verwendet wird -->
<!ELEMENT binding (#PCDATA | locus | dimensions | date | bibl | ref | displayScript | orig |
person | p)*>
<!ELEMENT bindingDesc (binding+)>
<!ELEMENT body (text)>
<!-- der 'body' wird im Rahmen des Projektes weiter definiert werden -->
<!ELEMENT catchwords (#PCDATA)>
<!ELEMENT cell (#PCDATA | %e.layout; | p | locus | abbr)*>
<!ATTLIST cell
        valign CDATA #IMPLIED
        colspan (2 | 6) #IMPLIED
>
<!ELEMENT change EMPTY>
<!ATTLIST change
        date CDATA #REQUIRED
        resp CDATA #REQUIRED
>
<!ELEMENT collation (#PCDATA | date | formula | catchwords | locus | sup | bibl | ref | person
| note)*>
<!ELEMENT collection (#PCDATA)>
<!ELEMENT colophon (#PCDATA | locus | span | date)*>
<!ELEMENT columns (#PCDATA | dimensions | locus)*>
<!ELEMENT condition (#PCDATA | locus | material | date | abbr | ref | person)*>
<!ELEMENT corr (#PCDATA)>
<!ATTLIST corr
        sic CDATA #REQUIRED
>
<!ELEMENT country (#PCDATA)>
<!ELEMENT cover (text)>
<!ELEMENT date (#PCDATA | sup)*>
<!-- sup fuer lateinische Jahreszahlen (Millesimo = M<sup>o</sup> -->
<!ELEMENT decoNote (#PCDATA | %e.layout; | %e.textIdentifier; | decoNote | locus | dimensions
| date | orig | term | bibl | author | title | person | place | hi | ref | list | p)*>
<!ATTLIST decoNote
        type (colophon | diagram | figure | gloss | initial | miniature | ornament | overview
| rubric | title) #IMPLIED
>
<!-- oft findet sich bei MASTER die Aussage, daß eine Liste der vorkommenden Typen
ausgearbeitet
        werden sollte; hier fangen wir damit an. Die Liste kann und soll natuerlich ausgedehnt
        werden -->
<!ELEMENT decoration (decoNote*)>
<!ELEMENT depth (#PCDATA)>
<!ATTLIST depth
        units CDATA #REQUIRED
>
<!ELEMENT dimensions (#PCDATA | %e.measures; | locus)*>
<!ATTLIST dimensions
        type (columns | leaves | written | objects | miniature) #IMPLIED
>
<!ELEMENT displayScript (#PCDATA)>
<!-- ein neues Element fuer die Kennzeichnung der Verwendung spezieller Schrifttypen in
Katalogen -->
<!ELEMENT div (pobject+)>
<!-- die Scans werden als 'pobject' eingebunden. -->
<!ELEMENT edition EMPTY>
<!ELEMENT editionStmt (edition)>
<!ELEMENT explicit (#PCDATA | %e.layout; | orig | locus | span | date | corr | title | person
| bibl | author | ref | place | note)*>
<!ELEMENT extent (#PCDATA | dimensions | material | locus | sup | date | ref | note)*>
<!ELEMENT fileDesc (titleStmt, editionStmt?, publicationStmt, seriesStmt, sourceDesc)>
<!ELEMENT foliation (#PCDATA | date | orig | locus | sup)*>
<!ATTLIST foliation
        type CDATA #IMPLIED
>
<!ELEMENT foliationDesc (foliation+)>
<!-- parallel zu 'bindingDesc' mit mehreren Unterelementen 'binding' -->
<!ELEMENT format (#PCDATA)>
<!ELEMENT formula (#PCDATA | sup | locus | note | date | bibl)*>
<!-- bibl fuer z.B. "siehe dazu..." -->
```

```
<!ELEMENT generalDescription (#PCDATA | p | bibl | date | person | orig | place | title)*>


<!ATTLIST generalDescription
        authority CDATA #REQUIRED
        auth-range CDATA #IMPLIED
>
<!-- generalDescription enthaelt die ausfuehrlichen allgemeinen Beschreibungen, wie sie z.B.
der Ausstellungskatalog
        GlaubeUndWissen    liefert.    Laengere    Beschreibungen    und    Wuerdigungen    der
Gesamthandschrift      sollten auch formal
        zu erkennen sein, da wir denken, daß die Texte inhaltlich in kein anderes
Element "hineinpassen" -->
<!ELEMENT generalNotes (#PCDATA | ref | p | locus | date | person | bibl | orig | title |
place | note)*>
<!ATTLIST generalNotes
        authority CDATA #IMPLIED
        auth-range CDATA #IMPLIED
>
<!-- allgemeine Bemerkungen werden mit generalNotes und aehnlichem ausgezeichnet;
     note enthaelt fussnotenartige Informationen -->
<!ELEMENT gloss (#PCDATA)>
<!ELEMENT handDesc (#PCDATA | locus | person | term | bibl | orig | ref | date)*>
<!ATTLIST handDesc
        scribe CDATA #IMPLIED
        scope (major | minor) #IMPLIED
>
<!ELEMENT handShift EMPTY>
<!ELEMENT head (#PCDATA)>
<!ELEMENT height (#PCDATA)>
<!ATTLIST height
        units CDATA #REQUIRED
>
<!ELEMENT hi (#PCDATA)>
<!ATTLIST hi
        rend (bold | italic) #IMPLIED
>
<!-- in CEEC soll nur 'hi' verwendet werden, nicht aber 'emph', weil 'emph' eine inhaltliche
        Deutung nahelegt, in den Katalogen ist allerdings nur eine Beschreibung vorgesehen -->
<!ELEMENT history (acquisition | origin | provenance | users)*>
<!ATTLIST history
        authority CDATA #REQUIRED
        auth-range CDATA #IMPLIED
>
<!ELEMENT idno (#PCDATA | locus)*>
<!ATTLIST idno
        authority CDATA #IMPLIED
        auth-range CDATA #IMPLIED
        type (webpage) #IMPLIED
>
<!ELEMENT img EMPTY>
<!ATTLIST img
        src (3punkte.jpg | chirho.jpg | kreuz.jpg | sonder.jpg) #REQUIRED
        alt CDATA #REQUIRED
        style CDATA #IMPLIED
>
<!-- in manchen Katalogen, z.B. Jaffé-Wattenbach, kommen Zeichen vor, die nicht direkt
        abgebildet werden koennen, weshalb wir sie als Bilder repraesentieren. -->
<!ELEMENT incipit (#PCDATA | orig | locus | span | corr | date | title | bibl | hi | author |
person | abbr | ref | place)*>
<!-- index auch fuer Items, wenn z.B. Kapitelueberschriften vorausgestellt sind -->
<!ELEMENT initium (#PCDATA | locus | orig)*>
<!ELEMENT institution (#PCDATA)>
<!ELEMENT item (#PCDATA | locus)*>
<!-- change fuer revisionDesc -->
<!ELEMENT keywords (list)>
<!ATTLIST keywords
        scheme CDATA #REQUIRED
>
<!ELEMENT layout (#PCDATA | dimensions | columns | writtenLines | writtenSpace | ruling |
ruledLines | preparationOfThePage | locus | date)*>
<!-- 'columns', 'ruledLines', 'writtenLines' sind bei MASTER Attribute, bei uns aber Elemente,
da wir auf
        diese Weise mehr Spielraum zur genauen Auszeichnung der Texte gewinnen -->
<!ELEMENT lb EMPTY>
<!ELEMENT list (item+ | change)>
<!ATTLIST list
        type (ordered | unordered) #REQUIRED
>
<!ELEMENT listBibl (head?, bibl+)>
<!-- innerhalb der Listen gibt es manchmal Spezifizierungen, um welche Art von Angaben
        es sich handelt. Beispiel: "Zu den Prologen:<bibl> ...". Diese werden im 'head'
        aufgenommen -->
```

```
<!ATTLIST listBibl


        authority CDATA #REQUIRED
        auth-range CDATA #IMPLIED
>
<!ELEMENT locus (#PCDATA)>
<!ATTLIST locus
        type CDATA #IMPLIED
        n CDATA #IMPLIED
>
<!-- der type veranlasst kleio eine andere Interpretationslogik zu verwenden -->
<!-- n liefert eine verwendbare Referenz, wenn sie sich nicht aus dem Inhalt von <locus>
        ergibt -->
<!ELEMENT material (#PCDATA | locus)*>
<!ELEMENT msContents (overview | %e.textIdentifier; | msItem | generalNotes | listBibl)*>
<!ATTLIST msContents
        authority CDATA #REQUIRED
        auth-range CDATA #IMPLIED
>
<!-- listBibl ist hier eingefuehrt, weil es manchmal thematisch zusammengehoerige
        bibliographische Angaben gibt, die dann z.B. durch eine Spezifizierung
        ("Die Glossen abgedruckt bei ...") eingeleitet werden -->
<!ELEMENT  msDescription  (generalDescription*,  msIdentifier,  msHeading*,  physDesc*,
msContents*, history*, msPart*, additional?)>
<!-- eine DTD bildet zu gewissem Grad eine Struktur in der Abfolge der Elemente ab. Hier wird
versucht, abzubilden, daß
        eine Handschrift aus Teilen bestehen kann und die Beschreibung in diesem Fall auf
hoher Ebene zunaechst das Element
        'msPart' enthalten muß. Die Elemente 'msHeading', 'physDesc', 'msContents' etc.
sind in der Hierarchie entweder 'msPart'
        oder direkt 'msDescription' untergeordnet. -->
<!ATTLIST msDescription
        status (compo | frag | uni) #REQUIRED
        type (carta | codex) #REQUIRED
>
<!ELEMENT msHeading (author | title | origPlace | origDate | textLang | generalNotes)+>
<!ATTLIST msHeading
        lang (de | en | la) #REQUIRED
        authority CDATA #REQUIRED
        auth-range CDATA #IMPLIED
>
<!ELEMENT  msIdentifier  (country,  settlement,  institution,  repository,  collection?,  idno,
altName*, secFol*)>
<!ELEMENT  msItem  (#PCDATA | %e.layout; | %e.textIdentifier; | msItem | author | title |
subtitle | locus | date | listBibl | bibl | person | place | orig | decoNote | note | ref |
term | material | handDesc | handShift | condition | table | cell | abbr | hi | p)*>
<!ATTLIST msItem
        authority CDATA #IMPLIED
        auth-range CDATA #IMPLIED
        id CDATA #IMPLIED
        type (TableOfContent | TironianNotes | condition | dryPointWriting | emptyPage | gloss
| marginalia | ownershipInscriptions | p_ContemporaryCorrections | p_Corrections |
p_CorrectionsAdditions | penTrials | textGloss) #IMPLIED
        textLang CDATA #IMPLIED
>
<!-- auch fuer 'msItem' haben wir eine Typenliste angefangen. Da in XML aber ein Attribut
nicht mehr als eine Auspraegung
        haben kann, muessen alle vorkommenden Kombinationen in die Liste eingetragen werden. -
->
<!ELEMENT msPart (idno, msHeading*, physDesc+, msContents*, history*, additional?)>
<!ELEMENT msWriting (#PCDATA | %e.layout; | term | locus | date | handDesc | decoNote | orig |
ref | title | bibl | p | person | overview)*>
<!ATTLIST msWriting
        hands (1 | 2 | 3 | 4 | 5 | many) #IMPLIED
>
<!ELEMENT musicNotation (#PCDATA | locus | date)*>
<!ELEMENT note (#PCDATA | %e.textIdentifier; | date | bibl | locus | title | person | ref |
orig | sup | span | hi | author)*>
<!-- note enthaelt fussnotenartige Informationen; allgemeine Bemerkungen werden
   mit generalNotes und aehnlichem ausgezeichnet -->
<!ATTLIST note
        authority CDATA #IMPLIED
        auth-range CDATA #IMPLIED
        id CDATA #IMPLIED
>
<!ELEMENT orig (#PCDATA | %e.layout; | %e.textIdentifier; | person | title | table | hi |
author | abbr | bibl | locus | date | gloss | p)*>
<!ATTLIST orig
        type CDATA #IMPLIED
>
<!ELEMENT origDate (#PCDATA)>
<!ATTLIST origDate
```

```
        certainty CDATA #IMPLIED
        evidence CDATA #IMPLIED


        exact CDATA #IMPLIED
        n (1 | 2) #IMPLIED
>
<!ELEMENT origPlace (#PCDATA)>
<!ATTLIST origPlace
        certainty CDATA #IMPLIED
        evidence CDATA #IMPLIED
        n (1 | 2) #IMPLIED
>
<!ELEMENT origin (#PCDATA | locus | place | date | person | bibl | ref | orig | decoNote |
dimensions | title | note | pb | p)*>
<!ATTLIST origin
        certainty CDATA #IMPLIED
        evidence CDATA #IMPLIED
>
<!ELEMENT overview (#PCDATA | person | ref | author | bibl | title | incipit | orig | locus |
date | place)*>
<!ELEMENT p (#PCDATA | %e.layout; | %e.textIdentifier; | %e.global; | locus | person | place |
table | hi | author | orig)*>
<!ATTLIST p
        type CDATA #IMPLIED
        style CDATA #IMPLIED
>
<!ELEMENT pb EMPTY>
<!ATTLIST pb
        n CDATA #REQUIRED
>
<!ELEMENT person (#PCDATA | date | abbr)*>
<!ATTLIST person
        role (donor | owner | scribe) #IMPLIED
        type (author | myth) #IMPLIED
>
<!ELEMENT physDesc (support | extent | collation | layout | msWriting | musicNotation |
decoration | bindingDesc | foliationDesc | additions | condition | generalNotes | note)*>
<!ATTLIST physDesc
        authority CDATA #REQUIRED
        auth-range CDATA #IMPLIED
>
<!ELEMENT place (#PCDATA)>
<!ATTLIST place
        type CDATA #IMPLIED
>
<!ELEMENT pobject (zobject?)>
<!ATTLIST pobject
        digref CDATA #REQUIRED
        extref CDATA #REQUIRED
        Zensus CDATA #IMPLIED
        roemisch CDATA #IMPLIED
        extref1 CDATA #IMPLIED
        extref2 CDATA #IMPLIED
        extref3 CDATA #IMPLIED
>
<!ELEMENT preparationOfThePage (#PCDATA | locus | dimensions)*>
<!-- 'preparationOfThePage ist ein neu eingefuehrtes Konzept: Inhalt ist die Art, wie
        eine Seite fue die Beschriftung vorbereitet worden ist (z.B. die Zirkeleinstiche).
        In MASTER gibt es dafue noch kein Konzept. 'ruling' beschreibt dagegen die Liniierung
-->
<!ELEMENT profileDesc (textClass)>
<!ELEMENT provenance (#PCDATA | person | date | locus | orig | bibl | ref | term | p | place |
hi)*>
<!ATTLIST provenance
        type (donor | patron) #IMPLIED
        notAfter CDATA #IMPLIED
>
<!ELEMENT pubPlace (#PCDATA)>
<!ELEMENT publicationStmt (publisher, pubPlace, idno, availability, date)>
<!ELEMENT publisher (#PCDATA)>
<!ELEMENT ref (#PCDATA | locus | date | sup)*>
<!ATTLIST ref
        type (altMs) #IMPLIED
        usage CDATA #IMPLIED
        target CDATA #IMPLIED
        subtype (shelfmark) #IMPLIED
>
<!ELEMENT repository (#PCDATA)>
<!ELEMENT respStmt (#PCDATA | person | sponsor)*>
<!ELEMENT revisionDesc (list)>
<!ELEMENT row (cell+)>
<!ELEMENT rubric (#PCDATA)>
```

```
<!ELEMENT ruledLines (#PCDATA)>
<!ELEMENT ruling (#PCDATA | locus | dimensions)*>


<!-- ruledLines gibt die ZAHL der Linien an und ruling die ART der Linienziehung -->
<!ELEMENT secFol (#PCDATA | %e.textIdentifier; | locus | date | orig | term | person | abbr |
note | ref | bibl)*>
<!ATTLIST secFol
        authority CDATA #REQUIRED
        auth-range CDATA #IMPLIED
>
<!-- 'secFol' ist eines der Elemente, die auf oberster Ebene integriert werden, weshalb das
uebergeordnete
        Element nicht die authority verraet; also muß das Element selber diese
Information tragen, deshalb
        ist die 'authority' REQUIRED -->
<!ELEMENT seriesStmt (title)>
<!ELEMENT settlement (#PCDATA)>
<!ELEMENT sourceDesc (listBibl*, msDescription)>
<!ELEMENT span (#PCDATA)>
<!ATTLIST span
        class CDATA #REQUIRED
>
<!-- bisher fuer die Darstellung griechischer Zeichen und die automatische Uebersetzung nach
HTML gebraucht -->
<!ELEMENT sponsor (#PCDATA)>
<!ELEMENT sub (#PCDATA)>
<!ELEMENT subtitle (#PCDATA)>
<!ELEMENT sup (#PCDATA)>
<!ELEMENT support (#PCDATA | material)*>
<!ELEMENT table (msItem | row)*>
<!ATTLIST table
        align CDATA #IMPLIED
>
<!ELEMENT term (#PCDATA)>
<!ATTLIST term
        type CDATA #REQUIRED
>
<!ELEMENT text (div)>
<!ELEMENT textClass (keywords)>
<!ELEMENT textLang (#PCDATA)>
<!ELEMENT title (#PCDATA | person | orig | locus | corr | date | abbr | span | hi | incipit |
note)*>
<!ATTLIST title
        usage CDATA #IMPLIED
        type (subtitle | descriptive | genus | ms | nickname | p_Werktitel) #IMPLIED
        lang (de | it) #IMPLIED
        level CDATA #IMPLIED
>
<!ELEMENT titleStmt (title, respStmt)>
<!ELEMENT users (#PCDATA | date | locus)*>
<!ELEMENT width (#PCDATA)>
<!ATTLIST width
        units CDATA #REQUIRED
>
<!ELEMENT writtenLines (#PCDATA | locus)*>
<!ELEMENT writtenSpace (#PCDATA | locus | height | width)*>
<!-- Stimmt nicht mit dem Konzept <dimensions type=written> ueberein,
    dieses enthaelt eine Beschreibung des Schriftspiegels, die komplexer als eine
    blosse Maßangabe sein kann -->
<!ELEMENT zobject EMPTY>
<!ATTLIST zobject
        digref CDATA #REQUIRED
        extref CDATA #REQUIRED
>
<!ENTITY auml "Ã¤">
<!ENTITY Auml "Ã„">
<!ENTITY aacute "Ã¡">
<!ENTITY agrave "Ã ">
<!ENTITY acirc "Ã¢">
<!ENTITY aring "Ã¥">
<!ENTITY aelig "Ã¦">
<!ENTITY abrevis "???">
<!ENTITY euml "Ã«">
<!ENTITY eacute "Ã©">
<!ENTITY Eacute "Ã‰">
<!ENTITY egrave "Ã¨">
<!ENTITY iacute "Ã­">
<!ENTITY iuml "Ã¯">
<!ENTITY icirc "Ã®">
<!ENTITY ouml "Ã¶">
<!ENTITY Ouml "Ã–">
<!ENTITY oacute "Ã³">
```

```
<!ENTITY ocirc "Ã´">
<!ENTITY uuml "Ã¼">


<!ENTITY Uuml "Ãœ">
<!ENTITY szlig "ÃŸ">
<!ENTITY ccedil "Ã§">
<!ENTITY ccaron "#010C">
<!ENTITY tcedil "???">
<!ENTITY mdash "-">
<!ENTITY ndash "â€"">
<!ENTITY para "Â¶">
<!ENTITY sect "Â§">
<!ENTITY ETH "Ã ">
<!ENTITY Alpha "Î'">
<!ENTITY Gamma "Î"">
<!ENTITY Eta "Î—">
<!ENTITY Iota "Î™">
<!ENTITY Omicron "ÎŸ">
<!ENTITY Rho "Î¡">
<!ENTITY Sigma "Î£">
<!ENTITY Theta "Î˜">
<!ENTITY Lambda "Î›">
<!ENTITY Pi "Î ">
<!ENTITY Upsilon "Î¥">
<!ENTITY deg "Â°">
```

- SYSTRAN Translation Stylesheet

```xml
<?xml version="1.0" encoding="UTF-8" ?>
<xsl:stylesheet     version="1.0"    xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
xmlns:date="http://exslt.org/dates-and-times"
xmlns:systran="http://www.systransoft.com/xslfunctions"
xmlns:sysel="http://www.systransoft.com/xslelements"
xmlns:exsl="http://exslt.org/common"    extension-element-prefixes="date    systran"
exclude-result-prefixes="date systran sysel">
<xsl:import href="tools.xsl" />
<xsl:import href="markup.xsl" />
<xsl:output encoding="UTF-8" indent="yes" />
<xsl:template match="*" mode="detect-type">element</xsl:template>
<xsl:template match="text()" mode="detect-type">text</xsl:template>
    <xsl:template     match="bibl|explicit|implicit|locus|msContents|msIdentifier|ref"
mode="dnt">1</xsl:template>
<xsl:template name="enrichTranslate">
<xsl:param name="source" />
<xsl:param name="options" />
<xsl:for-each select="$source">
<xsl:variable name="type">
<xsl:apply-templates mode="detect-type" select="." />
  </xsl:variable>
<xsl:choose>
<xsl:when test="$type='text'">
<xsl:call-template name="translate_text">
<xsl:with-param name="source" select="." />
<xsl:with-param name="options" select="$options" />
  </xsl:call-template>
  </xsl:when>
<xsl:when test="$type='element'">
<xsl:variable name="dnt">
<xsl:apply-templates mode="dnt" select="." />
  </xsl:variable>
<xsl:choose>
<xsl:when test="$dnt=1">
<xsl:copy-of select="." />
  </xsl:when>
<xsl:otherwise>
<xsl:variable name="translation">
<xsl:element name="{name()}">
<xsl:apply-templates select="@*" />
<xsl:for-each select="node()">
<xsl:call-template name="enrichTranslate">
<xsl:with-param name="source" select="." />
```

```xslt
  <xsl:with-param name="options" select="$options" />
    </xsl:call-template>
    </xsl:for-each>
    </xsl:element>
    </xsl:variable>
  <xsl:copy-of select="$translation" />
    </xsl:otherwise>
    </xsl:choose>
    </xsl:when>
- <xsl:otherwise>
  <xsl:copy-of select="." />
    </xsl:otherwise>
    </xsl:choose>
    </xsl:for-each>
    </xsl:template>
- <xsl:template match="//*[@authority!='Anderson']">
  <xsl:copy-of select="." />
- <xsl:copy>
  <xsl:apply-templates select="@*" />
- <xsl:attribute name="note">
  <xsl:text>Translated by SYSTRAN</xsl:text>
    </xsl:attribute>
- <xsl:call-template name="enrichTranslate">
  <xsl:with-param name="source" select="node()" />
  <xsl:with-param name="options" select="'max_nfw_percentage=20'" />
    </xsl:call-template>
    </xsl:copy>
    </xsl:template>
    </xsl:stylesheet>
```