SYSTRAN Translation Stylesheets: Machine Translation driven by XSLT

Pierre Senellart\textsuperscript{1,2}  Jean Senellart\textsuperscript{1}

\textit{XML Conference 2005}
November 17th, 2005
Machine Translation of Natural Language
- Machine Translation of Natural Language
- 20+ different languages
Machine Translation of Natural Language

- 20+ different languages
- 40+ language pairs
Machine Translation of Natural Language

20+ different languages

40+ language pairs

Translation service for Web portals: altavista, Google, Yahoo!...
Machine Translation of Natural Language

20+ different languages

40+ language pairs

Translation service for Web portals: altavista, Google, YAHOO!

Corporate customers: Ford, Cisco Systems, DAIMLERCHRYSLER
Machine Translation of Natural Language
20+ different languages
40+ language pairs
Translation service for Web portals: altavista, Google, Yahoo!
Corporate customers: Ford, Cisco Systems, DaimlerChrysler
Services for institutions: USA, EU
**SYSTRAN: A Leading Machine Translation Company**

- **Machine Translation** of Natural Language
- 20+ different languages
- 40+ language pairs
- Translation service for Web portals: **altavista**, **Google**, **Yahoo**
  
- Corporate customers: **Ford**, **Cisco Systems**, **DaimlerChrysler**
- Services for institutions: **USA**, **EU**
- **End-user products**
Traditional request

User or internal documentation, GUI: static

- Publishing quality
- Few tools (translation memories), human translation
Localization Industry

Traditional request

User or internal documentation, GUI: static
- Publishing quality
- Few tools (translation memories), human translation

New information media

Technical support FAQ, news, short span-life information: dynamic
- Highly technical documents
- Lower source quality (solutions by technician, not by technical writer), possible lower translation quality requirements
- Translation cost to be compared with additional support cost
- Localized content: tool for simplifying access to the information
Machine Translation

- A tool

Localization

Machine Translation and Localization
Introduction

Localization

Machine Translation and Localization

Machine Translation

- A tool
- Can be used as such for gisting

STS: Machine Translation driven by XSLT
XML Conf. 2005, 2005/11/17
Machine Translation

- A tool
- Can be used as such for gisting

Or as part of a localization process
Machine Translation

- A tool
- Can be used as such for gisting

- Or as part of a localization process
  - reviewed (support for human translators)
Machine Translation

- A tool
- Can be used as such for gisting

Or as part of a localization process
  - reviewed (support for human translators)
  - non reviewed (require customized resources and “trained” systems)
Machine Translation

- A tool
- Can be used as such for gisting

Or as part of a localization process
  - reviewed (support for human translators)
  - non reviewed (require customized resources and “trained” systems)
  - perfect translation quality can be obtained with completely controlled vocabulary and structure (e.g. Ford assembly line sheets)
Machine Translation

- A tool
- Can be used as such for *gisting*

Or as part of a *localization process*

- reviewed (support for human translators)
- non reviewed (require customized resources and “trained” systems)
- perfect translation quality *can* be obtained with completely controlled vocabulary and structure (e.g. Ford assembly line sheets)

- Main challenge: choosing the right *terminology* candidate
Machine Translation

- A tool
- Can be used as such for **gisting**

Or as part of a **localization process**

- reviewed (support for human translators)
- non reviewed (require customized resources and “trained” systems)
- perfect translation quality **can** be obtained with completely controlled vocabulary and structure (e.g. Ford assembly line sheets)

**Main challenge:** choosing the right **terminology candidate**

⇒ **customization**
How does all of this relate to XML?

Translation of an XML document

- Better machine translation quality can be obtained by using the structure of the document.
- Better yet when markup is added during authoring to help the translation process.
Translation of an XML document

- Better machine translation quality can be obtained by using the structure of the document.
- Better yet when markup is added during authoring to help the translation process.

From XML structure to natural language semantics!
How does all of this relate to XML?

Translation of an XML document

- Better machine translation quality can be obtained by using the structure of the document.
- Better yet when markup is added during authoring to help the translation process.

From XML structure to natural language semantics!

How?

Use XSLT to drive the translation process.
Outline

1. Introduction

2. The SYSTRAN Translation Stylesheet
   - Traditional document filtering
   - XSLT
   - Technical description
   - Examples
   - STS Catalog

3. Multilingual publishing workflows

4. Applying STS to non XML content

5. Conclusion
The SYSTRAN Translation Stylesheet

Traditional document filtering workflow

- Affordably priced translation software helps you break the language barrier fast.
- Transform your business into a multilingual workplace.
- Automatic translation solutions that help corporations think globally and communicate locally.
- Individual Users & Home Office
- Small & Medium Businesses
- Large Businesses
- Text: Enter up to 150 words for translation
- Web page: Enter the Web address of the page you wish to translate

Out of Context Sentence List

Translation

With global parameters

Translated List of Sentences

Senellart, Senellart (SYSTRAN & INRIA)
Selection of user-defined dictionaries
Translation customization in traditional workflow

- Selection of user-defined dictionaries
- Modification of linguistic parameters (imperative translation, pronoun genders...)

Do Not Translate expressions

Rule parameterization

Traditional workflow: translation customization is global

No way to differentiate sentences once they are extracted.

Need to be able to use the structure of a document to:
- Modify translation options in a local way (different translations for the same source depending on the context)
- Define DNT expressions in a dynamic way
- Introduce new types of parameterization
Translation customization in traditional workflow

- Selection of user-defined dictionaries
- Modification of linguistic parameters (imperative translation, pronoun genders...)
- Do Not Translate expressions
Translation customization in traditional workflow

- Selection of user-defined dictionaries
- Modification of linguistic parameters (imperative translation, pronoun genders...)
- Do Not Translate expressions
- Rule parameterization
Translation customization in traditional workflow

- Selection of user-defined dictionaries
- Modification of linguistic parameters (imperative translation, pronoun genders...)
- Do Not Translate expressions
- Rule parameterization

Traditional workflow: translation customization is **global**
Translation customization in traditional workflow

- Selection of user-defined dictionaries
- Modification of linguistic parameters (imperative translation, pronoun genders...)
- Do Not Translate expressions
- Rule parameterization

Traditional workflow: translation customization is **global**

No way to differentiate sentences once they are extracted.
Translation customization in traditional workflow

- Selection of user-defined dictionaries
- Modification of linguistic parameters (imperative translation, pronoun genders . . .)
- Do Not Translate expressions
- Rule parameterization

Traditional workflow: translation customization is **global**

- No way to differentiate sentences once they are extracted.
- Need to be able to use the **structure** of a document to:
  - Modify translation options in a local way (different translations for the same source depending on the context)
Translation customization in traditional workflow

- Selection of user-defined dictionaries
- Modification of linguistic parameters (imperative translation, pronoun genders...)
- Do Not Translate expressions
- Rule parameterization

Traditional workflow: translation customization is global

No way to differentiate sentences once they are extracted.

Need to be able to use the structure of a document to:
- Modify translation options in a local way (different translations for the same source depending on the context)
- Define DNT expressions in a dynamic way
Translation customization in traditional workflow

- Selection of user-defined dictionaries
- Modification of linguistic parameters (imperative translation, pronoun genders...)
- Do Not Translate expressions
- Rule parameterization

Traditional workflow: translation customization is **global**

- No way to differentiate sentences once they are extracted.
- Need to be able to use the **structure** of a document to:
  - Modify translation options in a local way (different translations for the same source depending on the context)
  - Define DNT expressions in a dynamic way
  - Introduce new types of parameterization
eXtensible Stylesheet Language: Transformation
eXtensible Stylesheet Language: Transformation

Transformation Language for XML documents
- eXtensible Stylesheet Language: Transformation
- Transformation Language for XML documents
- Templates which match nodes in the document
- eXtensible Stylesheet Language: Transformation Language for XML documents
- Templates which match nodes in the document
- Cascading system (template or stylesheet level)
eXtensible Stylesheet Language: Transformation Language for XML documents
- Templates which match nodes in the document
- Cascading system (template or stylesheet level)
Used for:
- eXtensible Stylesheet Language: Transformation Language for XML documents
- Templates which match nodes in the document
- Cascading system (template or stylesheet level)
- Used for:
  - Translating from one XML application to another
XSLT

- eXtensible Stylesheet Language: Transformation Language for XML documents
- Templates which match nodes in the document
- Cascading system (template or stylesheet level)
- Used for:
  - Translating from one XML application to another
  - Publishing XML documents to a publishing format (XSL-FO, (X)HTML...)

Senellart, Senellart (SYSTRAN & INRIA)
eXtensible Stylesheet Language: Transformation Language for XML documents
Templates which match nodes in the document
Cascading system (template or stylesheet level)

Used for:
- Translating from one XML application to another
- Publishing XML documents to a publishing format (XSL-FO, (X)HTML...)
- Modifying XML documents
**STS**: SYSTRAN Translation Stylesheet
STS: SYSTRAN Translation Stylesheet

Standard XSLT 1.0 stylesheet
STS: SYSTRAN Translation Stylesheet
Standard XSLT 1.0 stylesheet
With:
STS: SYSTRAN Translation Stylesheet

- Standard XSLT 1.0 stylesheet
- With:
  - extension functions in a SYSTRAN namespace
- **STS**: SYSTRAN Translation Stylesheet
- Standard XSLT 1.0 stylesheet
- With:
  - extension functions in a SYSTRAN namespace
  - pre-defined templates (cascade)
**STS**: SYSTRAN Translation Stylesheet

Standard XSLT 1.0 stylesheet

With:

- extension functions in a SYSTRAN namespace
- pre-defined templates (cascade)

**Basic idea**

Machine Translation: just an XSLT extension function
Extension functions — translation

=node-set systran:translate(node-set)=

Returns the translation of the argument node-set, which must validate against:

<!ELEMENT par (#PCDATA|mark|typo|tag)*>  
<!ELEMENT mark (#PCDATA|mark|typo|tag)*>  
<!ELEMENT typo (#PCDATA|mark|typo|tag)*>  
<!ELEMENT tag ANY>
node-set systran:translate(node-set)

Returns the translation of the argument node-set, which must validate against:

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

- **mark**: provides additional information to or get additional feedback from translation engines
- **typo**: local character properties (e.g. bold font, hyperlinks)
- **tag**: must remain as is in translation output

**#PCDATA**: text to be translated
Extension functions — translation: example

Example (Source)

```xml
<par>
  <typo type="bold">Quick</typo>
  <mark action="set" type="domain"
       value="information_technology">boot</mark>!

  <tag>This <foo>will not be</foo> translated.</tag>
</par>
```

Example (Target)

```xml
<par>
  Démarrage <typo type="bold">rapide</typo>!
  <tag>This <foo>will not be</foo> translated.</tag>
</par>
```
Extension functions — options

```cpp
string systran:getValue(string)
void systran:pushValue(string,string)
void systran:popValue()
```

Manage a stack of translation engine options:

- General options (e.g. current source language)
- Feedback from translator (e.g. metrics)
- Global linguistic options (e.g. should “you” be translated as “tu” or “vous”?)
- Stylesheet-specific options
Utility templates

- Utility tools.xsl stylesheet, imported from all STS
Utility templates

- Utility tools.xsl stylesheet, imported from all STS
- Number of utility templates
Utility templates

- Utility tools.xsl stylesheet, imported from all STS
- Number of utility templates
- Wrapper template translate_par for systran:translate
Utility templates

- **Utility tools.xsl stylesheet**, imported from all STS
- Number of utility templates
- Wrapper template `translate_par` for `systran:translate`
- Default preprocess/postprocess behaviors
Utility templates

- Utility tools.xsl stylesheet, imported from all STS
- Number of utility templates
- Wrapper template translate_par for systran:translate
- Default preprocess/postprocess behaviors
- Coding style for STS
The SYSTRAN Translation Stylesheet

Examples

Generic XML translation

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

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

<xsl:template match="text()">
    <xsl:call-template name="translate_par">
        <xsl:with-param name="source" select="."/>
    </xsl:call-template>
</xsl:template>

</xsl:stylesheet>
```
<xsl:template
  match="h:p|h:li|h:h1|h:h2|h:h3|h:h4|h:h5|h:h6">
  <xsl:copy>
    <xsl:apply-templates select="@*"/>
    <xsl:call-template name="translate_par">
      <xsl:with-param name="source" select="node()"/>
    </xsl:call-template>
  </xsl:copy>
</xsl:template>
XHTML — character properties

```xml
<xsl:template
    match="h:b|h:strong"
    mode="preprocess">
    <typo bold="1">
        <xsl:apply-templates mode="preprocess"/>
    </typo>
</xsl:template>

<xsl:template match="typo[@bold='1']" 
    mode="postprocess">
    <xsl:element
        name="strong"
        namespace="http://www.w3.org/1999/xhtml">
        <xsl:apply-templates mode="postprocess"/>
    </xsl:element>
</xsl:template>
```
XHTML — passing options

```xml
<xsl:template match="h:title">
  <xsl:value-of select="systran:pushValue(’TITLE’,’1’)" />
  <xsl:copy>
    <xsl:apply-templates select="@*"/>
    <xsl:call-template name="translate_par">
      <xsl:with-param name="source" select="text()"/>
    </xsl:call-template>
  </xsl:copy>
  <xsl:value-of select="systran:popValue()" />
</xsl:template>
```
XHTML — fine selection of what to translate

```xml
<xsl:template match="@title|@alt">
  <xsl:call-template name="translate_par">
    <xsl:with-param name="source" select="."/>
  </xsl:call-template>
</xsl:template>

<xsl:template match="@title[local-name(../)=\'abbr\'
    or local-name(../)=\'acronym\']">
  <xsl:copy/>
</xsl:template>
```
<xsl:template match="@title[local-name(..)='img']">
  <xsl:value-of select="systran:pushValue('DOMAIN','INFORMATION_TECHNOLOGY')" />
  <xsl:call-template name="translate_par">
    <xsl:with-param name="source" select="."/>
  </xsl:call-template>
  <xsl:value-of select="systran:popValue()" />
</xsl:template>
XHTML — keep both source and target

```xml
<xsl:template match="h:blockquote">
  <xsl:copy-of select="." />
  <xsl:copy>
    <xsl:apply-templates select="@*"/>
    <xsl:call-template name="translate_par">
      <xsl:with-param name="source" select="node()"/>
    </xsl:call-template>
  </xsl:copy>
</xsl:template>
```
How to select the STS used to translate a document?

- Manual selection
How to select the STS used to translate a document?

- **Manual** selection
- **Automatic** selection, from information in a catalog:
How to select the STS used to translate a document?

- **Manual** selection
- **Automatic** selection, from information in a catalog:
  - by PUBLIC DTD identifier
How to select the STS used to translate a document?

- **Manual** selection
- **Automatic** selection, from information in a catalog:
  - by PUBLIC DTD identifier
  - by root namespace
How to select the STS used to translate a document?

- **Manual** selection
- **Automatic** selection, from information in a catalog:
  - by PUBLIC DTD identifier
  - by root namespace
  - by root node name
How to select the STS used to translate a document?

- **Manual** selection
- **Automatic** selection, from information in a catalog:
  - by PUBLIC DTD identifier
  - by root namespace
  - by root node name

**Example (XHTML)**

```xml
<application stylesheet="xhtml.xsl" version="1.0"
mimetype="application/xhtml+xml">
    <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>
```
Outline

1. Introduction

2. The SYSTRAN Translation Stylesheet

3. Multilingual publishing workflows
   - Traditional localization workflow
   - Machine translation as part of the publishing process
   - Authoring with machine translation in mind

4. Applying STS to non XML content

5. Conclusion
Localization performed on **publishing format** (Word, HTML with presentation markup, PDF...).
Localization performed on **publishing format** (Word, HTML with presentation markup, PDF...).

Translation engines must deal with all kind of presentation/formatting/non-structural aspects.
Localization performed on **publishing format** (Word, HTML with presentation markup, PDF...).

Translation engines must deal with all kind of presentation/formatting/non-structural aspects.

Translation sometimes left to the user, who does not know how to customize it best.
• Localization performed directly on **authoring format** (typically, an XML format).
Localization performed directly on **authoring format** (typically, an XML format).

Possible to exploit the **structure** of the content with STS.
Localization performed directly on authoring format (typically, an XML format).

Possible to exploit the structure of the content with STS.

Possible for the author, who knows the content, to customize the translation process.
Multilingual publishing workflows

Authoring with machine translation in mind

Original source document

- Multilingual Authoring
- Publishing
- Translating

STS: Machine Translation driven by XSLT
XML Conf. 2005, 2005/11/17
Authoring with machine translation in mind

- Original source document
- Translated, and annotated by translation engines with:

Multilingual Authoring -> Publishing

Multilingual Authoring

Translating
Authoring with machine translation in mind

- Original source document
- Translated, and **annotated** by translation engines with:
  - ambiguity markup

Multilingual publishing workflows
Authoring with machine translation in mind

- Original source document
- Translated, and annotated by translation engines with:
  - ambiguity markup
  - not found words markup
Authoring with machine translation in mind

- Original source document
- Translated, and *annotated* by translation engines with:
  - ambiguity markup
  - not found words markup
  - linguistic complexity metrics
Authoring with machine translation in mind

- Original source document
- Translated, and annotated by translation engines with:
  - ambiguity markup
  - not found words markup
  - linguistic complexity metrics
  - dictionary coverage metrics
Authoring with machine translation in mind

- Original source document
- Translated, and annotated by translation engines with:
  - ambiguity markup
  - not found words markup
  - linguistic complexity metrics
  - dictionary coverage metrics
- Authoring uses this feedback to adapt content or add markup to help the translation process
With more than 1.5 million installed, the Cisco 2500 series is one of the most popular solutions for a wide range of cost-effective configurations, including dual LAN, integrated router/hub, and integrated access server models.

Avec plus de 1.5 million étant installé, les 2500 séries de Cisco sont l’une des solutions les plus populaires pour un éventail de configurations, y compris le LAN duel, de routeur/hub intégrés, et de modèles intégrés de serveur d’accès.

For example, integrated call switching and call handling features enable small or branch offices to use their Cisco access solution for call handling and remote access instead of having to invest in a PBX system.

Par exemple, la commutation d’appel et les dispositifs intégrés de manipulation d’appel permettent succursales de petites ou d’employer leur solution d’accès de Cisco pour la manipulation d’appel et l’accès à distance au lieu de devoir investir dans un système de PBX.
## Authoring tools

The director comments on the making of the film.

Le directeur présente ses observations sur la fabrication du film.

<table>
<thead>
<tr>
<th>Expression</th>
<th>Choice</th>
</tr>
</thead>
<tbody>
<tr>
<td>comments</td>
<td>noun</td>
</tr>
<tr>
<td>film</td>
<td>noun</td>
</tr>
<tr>
<td>comments</td>
<td>noun</td>
</tr>
<tr>
<td>film</td>
<td>noun</td>
</tr>
</tbody>
</table>

Senellart, Senellart (SYSTRAN & INRIA)

STS: Machine Translation driven by XSLT

XML Conf. 2005, 2005/11/17
The director comments on the making of the film.

The director comments on the making of the film.

The director <amb type="noun/verb" choice="verb" confidence="75">
  comments
</amb>
on the making of the film.

Senellart, Senellart (SYSTRAN & INRIA)
Outline

1 Introduction

2 The SYSTRAN Translation Stylesheet

3 Multilingual publishing workflows

4 Applying STS to non XML content
   - Actual documents are not always XML...
   - XLIFF
   - Using XLIFF and STS together

5 Conclusion
Sometimes, no clean, structured, XML content to translate
Sometimes, no clean, structured, XML content to translate

But even Word documents, RTF, HTML, PDF, LaTeX... expose some kind of internal structure that could help the translation process
Applying STS to non XML content

Actual documents are not always XML...

- Sometimes, no clean, structured, XML content to translate
- But even Word documents, RTF, HTML, PDF, LaTeX... expose some kind of internal structure that could help the translation process
- Plus, there might be some application-dependent authoring styles, too (every phrase in fixed width font is code, and should not be translated)
Sometimes, no clean, structured, XML content to translate

But even Word documents, RTF, HTML, PDF, LaTeX... expose some kind of internal structure that could help the translation process

Plus, there might be some application-dependent authoring styles, too (every phrase in fixed width font is code, and should not be translated)

HTML: use HTML DOM as if it were an XML DOM, and apply STS
Sometimes, no clean, structured, XML content to translate

But even Word documents, RTF, HTML, PDF, LaTeX... expose some kind of internal structure that could help the translation process

Plus, there might be some application-dependent authoring styles, too (every phrase in fixed width font is code, and should not be translated)

HTML: use HTML DOM as if it were an XML DOM, and apply STS

Other formats: need for something different
OASIS standard
OASIS standard

Stores in an XML document:
OASIS standard

Stores in an XML document:

- **Skeleton** of a non-XML document
OASIS standard

Stores in an XML document:
- **Skeleton** of a non-XML document
- Extracted **text content**
OASIS standard

Stores in an XML document:

- **Skeleton** of a non-XML document
- Extracted *text content*
- Normalized information about the **document structure** (lists, tables...)

Senellart, Senellart (SYSTRAN & INRIA)
STS: Machine Translation driven by XSLT
XML Conf. 2005, 2005/11/17
OASIS standard

Stores in an XML document:

- Skeleton of a non-XML document
- Extracted text content
- Normalized information about the document structure (lists, tables...)
- Normalized character properties (bold font, italic...)
Applying STS to non XML content

XLIFF: XML Localization Interchange File Format

- OASIS standard
- Stores in an XML document:
  - **Skeleton** of a non-XML document
  - Extracted **text content**
  - Normalized information about the **document structure** (lists, tables...)
  - Normalized **character properties** (bold font, italic...)
- Allows for the reconstruction (**merge**) of the original document, or of another document with localized text
Using XLIFF and STS together
Outline

1. Introduction
2. The SYSTRAN Translation Stylesheet
3. Multilingual publishing workflows
4. Applying STS to non XML content
5. Conclusion
   - Summary
   - Perspectives
Localization of new media requires the use of Machine Translation
Localization of new media requires the use of Machine Translation.

Machine Translation benefits dramatically from XML structure.
Localization of new media requires the use of Machine Translation
Machine Translation benefits dramatically from XML structure
STS: XSLT drives the translation process
Localization of new media requires the use of Machine Translation.

Machine Translation benefits dramatically from XML structure.

STS: XSLT drives the translation process.

Integrating Machine Translation inside the publishing process.
Localization of new media requires the use of Machine Translation
Machine Translation benefits dramatically from XML structure
STS: XSLT drives the translation process
Integrating Machine Translation inside the publishing process
Better: toward real multilingual authoring
SYSTRAN translation stylesheets able to handle multiple language pairs in a single pass
SYSTRAN translation stylesheets able to handle multiple language pairs in a single pass

Facilitate “Authoring with MT in mind”: 
SYSTRAN translation stylesheets able to handle multiple language pairs in a single pass

*Facilitate “Authoring with MT in mind”:*
  - Authoring tools oriented towards multilingual publishing
Perspectives

- SYSTRAN translation stylesheets able to handle **multiple language pairs** in a single pass
- Facilitate “Authoring with MT in mind”:
  - **Authoring tools** oriented towards multilingual publishing
  - Machine Translation supported **CMS**