

The Compendium Translator System

Juan A. Alonso – Compendium España S.L.

Gregor Thurmair – Compendium Deutschland GmbH

1 Historical Background

The architecture of MT systems must be adapted to the user roles to be supported.

1.1 MT for Professional Translators

The first generation of MT systems was designed to give support to **professional translators** and documentation experts (cf. Proceedings of MT-Summit 1986). This had consequences for the system design, namely:

- Support of text handling formats, mainly of publishing systems. **Layout conversion** was one of the prominent system features
- The possibility of having a very fine-grained **human intervention** with the MT system on each processing level
- Elaborate lexicon maintenance and **coding** tools.

It turned out that MT systems were not overwhelmingly adopted by translation and documentation experts, although some installations showed significant productivity gains.

1.2 The PC-Age: Simplifying MT

The boost of the PC / Windows industry led to a switch in the MT paradigm. The MT systems competed to be available on single user PCs, to support personal translation of **end users**.

As a consequence, many features which allowed to tune the systems were switched off, as they were too complicated to use for non-trained users, and simplifications of the interfaces and the possibilities of influencing the translation process were implemented.

The difficulty lies in the fact that simplifying the handling of the system requires significantly more intelligence on the system side, to cope with missing information by users. In cases where such additional intelligence was not provided, the translation accuracy and quality decreased.

However, there was a certain success of the MT boxes, as systems like Systran, Globalink's Power Translator, Linguatrec's Personal Translator (IBM-derivate) (Lehmann, 1995), Langenscheidts T1 (METAL-derivate) (Schwall, 1997), ProMT and others show.

1.3 The Ruling of Internet

With the upcoming internet technology and the idea of language portals the architecture of the MT systems changed again. It is not just a requirement to support HTML as text format, but the complete internal structure of the systems had to be adapted.

One of the requirements was to support **multi-vendor platforms**, i.e. portals where different MT providers' systems ran simultaneously. Examples were Alis or L&H's iTranslator, where four different MT systems had to be integrated. The consequence of this was that

- MT engines would be encapsulated to run as a server component, and provide an API to support translation requests
- A special component needed to be produced to do the scheduling and task management, like DTS (McLaughlin, 2001) or LTC (Barrett, 2001).

Another consequence was a clear model of **user types**:

- **End users**, e.g. members of a company who want to translate something, need an easy-to-use interface by which they can upload a document, set the target language, and trigger the translation.
- Users who are responsible for the **linguistic administration** of the server, and need a special administration client to edit system resources, maintain the dictionaries etc., to provide the best possible translation results.

This distinction is the basis for all types of language portals, be it in the internet (as ASP providers) or in the intranet.

For the MT systems involved, this means that they must fit such constellations:

- Provide end-user (web-based) clients by which translation jobs can be launched
- Provide administration capabilities (admin clients) to maintain the system.
- Provide tuning and adaptation options.

One key element of success is that the MT system must be able to be tuned to a specific application environment. Such tuning implies:

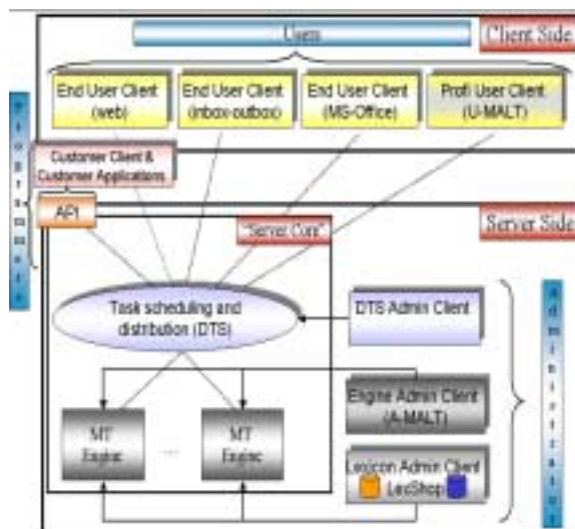
- Tools for **text pre- and postprocessing**, depending on the users' main text formats.
- Tools to **import and control user specific terminology**.
- Tools to **improve the overall performance** of the system, by setting specific translation parameters, global control of missing vocabulary, and options for logging and reporting.

The availability of such options makes systems again attractive for professional documentation and translation, a user group called "professional end users". Selection of subject areas, memory modules, preprocessing options etc. can again be supported.

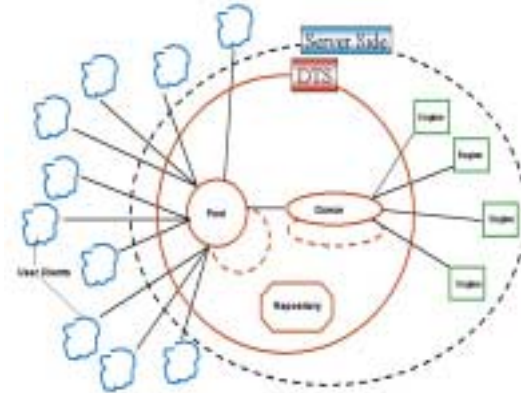
The result of these developments is the basis of the design of the Compendium Translator, which continues the METAL, T1, and iTranslator technology in this new setting.

2 A Flexible System Architecture

The Compendium Translator incorporates a flexible, distributed client-server architecture, as can be seen in the following picture:



All the components in the system can be placed in one or more computers interconnected via Internet, Intranet or LAN:



Within a typical workflow, users issue translation requests from the different user clients; these translation requests are then handled by the DTS module and sent to the available translation engines. Once the request has been translated, the results are sent back to the user.

2.1 The Server Side

On the server side, the following modules can be identified:

- One or more translation engines, which are the components that take care of the pure translation tasks (including not only MT, but also translation memories) and process the translation requests coming from the clients. The engines contain all the linguistic knowledge of the system.
- The DTS module, in charge of scheduling and distributing incoming translation requests from the user clients among the available translation engines. This component consists of one or more pools, where translation requests are stored, one or more queues, where translation requests are processed and filtered and one repository, where all the entities known to the system (pools, queues, engines, etc.) are registered.
- The administration clients, used by administrators in order to configure, monitor and maintain the system (see below).
- The Client Application Programming Interface (Client API) allows customer applications to access the Compendium

Translator system. The Client API offers three types of interfaces: Java, CORBA and SOAP.

2.2 The User Side

On the user side there are two types of entities interacting with the system server side:

- The User Clients, which can be of different types: Web clients, Inbox/Outbox clients, MS-Office Clients, Professional Clients, etc. (see below)
- Customer applications, interacting with the server part through the Client API.

3 The Translation Engines and the Linguistic Components

The translation engines contain all the machinery directly related to the machine translation process:

- All the kernel software
- Converters and Text Handling modules
- Pre-processing and post-processing modules, with which the user can define string-level operations to operate on the translation input and on the translation output.
- Translation Memory modules
- The linguistic components used to perform the actual machine translation process

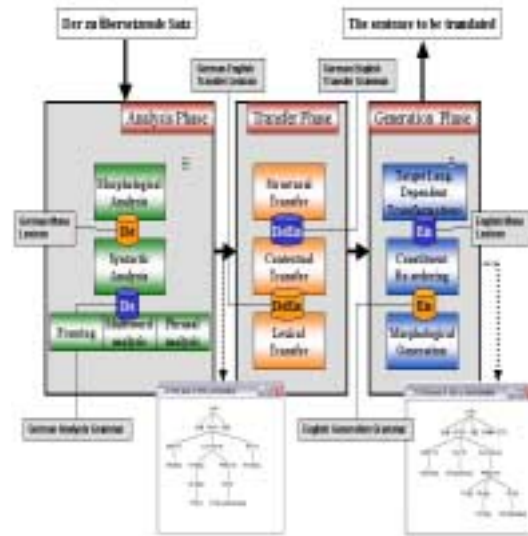


The linguistic components consist of

- Lexicons (mono- and bilingual)
- Grammars for analysis, transfer and generation

General Linguistic Information

As already mentioned, these linguistic components are used during the machine translation process:



4 User Clients: Different Ways to Access the System

Users can send translation requests to the system from a number of different User Clients:

- **Web Client:** this is the default User Client offered with the system. Users interact with the system through a series of JSP Web pages from their local browser. Typically, the layout, the display language(s) and the functionality of these pages are customized according to the particular needs of the customer. Nevertheless, the basic functionality of these pages includes:
 - Translation of short texts
 - Translation of documents (RTF, HTML, TXT, and optionally, MS-Word DOC, PDF, etc.)
 - Translation of URLs
 - Setting of translation parameters
 - Price Quoting before translation
 - Possibility of getting back the translation results via e-mail
- **Inbox/Outbox Client:** This client cyclically scans for documents under a user-specified input directory (inbox). It takes the documents from there, translates them and puts the results onto a user-

specific output directory (outbox). The I/O Client has a User Interface to control the Client operation and through which the translation parameters can be set.

- **MS-Office Client:** The user can send documents to translation from within the Microsoft Office applications (Word®, PowerPoint®, Excel®, etc.).
- **Professional Translation Client:** This is a client that offers a complete translation environment for professional translators with the full tuning possibilities.
- **Customer's own Applications and Clients:** As mentioned above, through the use of the system Client API, it is possible to send translation requests either from a customer's existing application, or from a new User Client designed at the customer's site, using e.g. SOAP.

5. Admin Clients: How to Administrate and Tailor the System

5.1 Lexicon Administration

There is a specialised system component which enables users to administer lexicons on a professional level. All lexicon features and values are available for editing and interactive coding, including complex contextual transfers. Powerful import / export facilities enable lexicon administrators to import quickly large amounts of customer terminology (several ten thousands of terms); monolingual and bilingual term extraction tools, and defaulting components support quick identification of the relevant material.

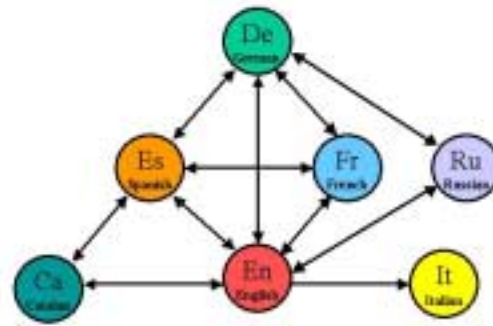
5.2 System Administration

A special system administration client is responsible for defining system parameters, translation settings, pre- and post-editing options, administration of translation memories, editing of system resources, and the like.

6. Summary of Main System Features

- Use of state-of-the-art technologies (Java, CORBA, SOAP, JSP, etc.)
- Use of a transfer-based machine translation approach.

- Up to 21 translation directions available:



- Possibility to create, administrate and use translation memory modules.
- High translation performance (around 1,200 pages/hour per engine).
- Full scalability by using more engines, pools, queues etc.
- Client-Server architecture with automatic load distribution among the engines.
- User authentication
- Transaction logging and reporting
- Possibility of system tuning by the customer (translation memories, pre- and post-processing string-level operations, lexicon coding and maintenance)
- Possibility of integration into customer systems through the Client API.
- Easy expandability to other services apart from machine translation.

Citations

- Juan Alonso et al., 2001: “*Collapsing morphological information in lexical databases for NLP applications*”. Proc. MT-Summit-VIII, Santiago
- Alan Barrett et al., 2001: “*Lotus Translation Components*” TQPro Report
- Ulrike Bernardi et al., 2001: “*A touch of MALT*”. Proc. MT Summit-VIII, Santiago
- Hubert Lehmann, 1995: “*Machine Translation for Home and Business Users*”. Proc. MT-Summit-V Luxemburg.
- Steve McLaughlin et al., 2001: “*DTS, a delivery system for translations and translation-related services*”. Proc. ASLIB 2001
- Ulrike Schwall et al., 1997: “*From Metal to TI*”. Proc. MT-Summit-VI, San Diego