

# Rapid Development of New Language Pairs at SYSTRAN

Sylvain Surcin, Elke Lange, Jean Senellart

SYSTRAN  
La Grande Arche  
1 Parvis de La Défense  
92044 Paris La Défense Cedex  
France  
{surcin,senellart}@systran.fr, elange@systransoft.com

## Abstract

One of the mottos of pure statistical MT system promoters is that it is possible to “build a new language pair” overnight, but the development of a new language pair for a proficient rule-based translator requires a great amount of effort in linguistic rules and resources description. Therefore, we are interested in rapid development techniques for rule-based systems. In this paper, we present the work that was conducted at SYSTRAN during the past year: we successfully developed 12 new language pairs in one year. This led us to shift some architecture paradigms in our translators, to expand implementations with the notion of Linguistic Families, to open our interfaces to more readable formats, and to design ways to work with fully multisource / multitarget aligned dictionaries in order to save time, in particular with the coding effort.

## Introduction

SYSTRAN’s effort over the past couple of years has focused on the development of parsers for new languages - 22 parsers are now available- and on implementing a paradigm shift in order to build more modular, streamlined translators (Attnäs et al., 2005). At the beginning of 2006, SYSTRAN’s offer included 38 commercial language pairs and 15 additional language pairs developed for specific customers. One strategic goal is to complete the matrix of available language pairs, including the new parsers and new target languages.

The ideal situation for an organization whose purpose is to develop Machine Translation systems is the ability to easily add a new language pair to its already existing offer in a very short span of time. Developing a translation engine of a new language pair for a rule-based system (such as SYSTRAN’s) requires much effort in terms of manpower and duration. This effort can of course be reduced by using a strong modular approach and powerful tools for the construction of linguistic resources.

To the contrary, proponents of statistics-based Machine Translation claim they are able to build a new engine overnight, provided a reasonable amount of training data is available (Koehn, 2005; Och & Ney, 2000), yet further studies proved this approach is not so easy to implement either (Foster et al., 2003).

During the year 2006, SYSTRAN decided to embark on an ambitious project: to develop a set of 12 new language pairs in less than a year, and to do this with a minimal team of linguists and developers. The systems were to leverage the existing code and resources by further enhancing the modularity of SYSTRAN’s code. A radically new approach of lexical resources also had to be designed in order to rapidly build and maintain 12 bilingual dictionaries under the form of a unique multisource - multi-target dictionary. And finally, we wanted these translators to interface with a statistical approach: to use automatically extracted dictionaries of multiword expressions in addition to the regular dictionaries, and be able to assign

weight to linguistic rules applied in the translator, based on statistical observation of training corpora.

This paper presents the strategies we applied during this development phase, the main problems that arose and how we solved them. Last, we draw upon a few important methodological conclusions and discuss open issues as we begin to bring these translators to market.

## Modular Development

An obvious entry point for rapid development of Machine Translation systems is to take advantage of the classical architecture of transfer-based MT systems such as SYSTRAN translators. Figure 1 recalls the schematic classical architecture with three main phases (or modules): *analysis* (parsing), *transfer* and *generation*.

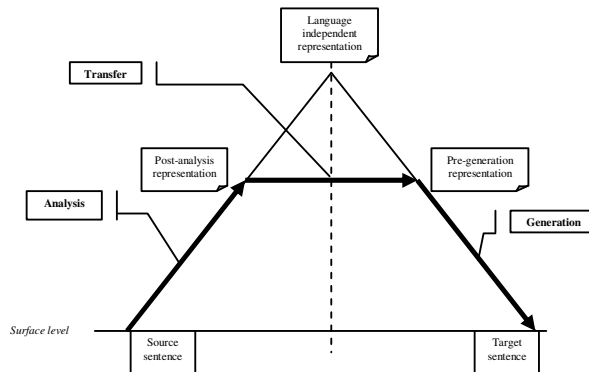


Figure 1: Classical Transfer Architecture

The ideal situation in such architecture would be to reach an abstract, language independent representation of the source sentence after analysis (Sérasset & Boitet, 2000). From this representation, the generation would build the target sentence from pure semantic and rhetorical information. But in reality, the analysis only reaches an intermediary level of abstraction, with much syntactical information about the source structure, and this has to be transferred into another intermediary structure with syn-

tactical information about the target structure. Maintaining an “intermediate” level of the analysis is also strategic in terms of translation quality – based on contrastive grammar; transfer modules correctly handle similar phenomena in source and target languages and do not require “perfect” linguistic analysis of these phenomena.

In our case, we wanted to develop new language pairs consisting of all the possible combinations of source and target languages among German (De), Spanish (Es), Italian (It), and Portuguese (Pt). There are 12 combinations: DeEs, EsDe, DeIt, ItDe, DePt, PtDe, EsIt, ItEs, EsPt, PtEs, ItPt, and PtIt. Additionally, we wanted to reuse as much existing code as possible and take advantage of similarities within language families.

In fact, in most available SYSTRAN translators, the modules architecture is a little bit more complex: the paradigm in use is *analysis + transfer + synthesis + rearrangement*. The *analysis* module depends on the source language; the *synthesis* module depends on the target language only, while both *transfer* and *rearrangement* of words depend on source and target languages.

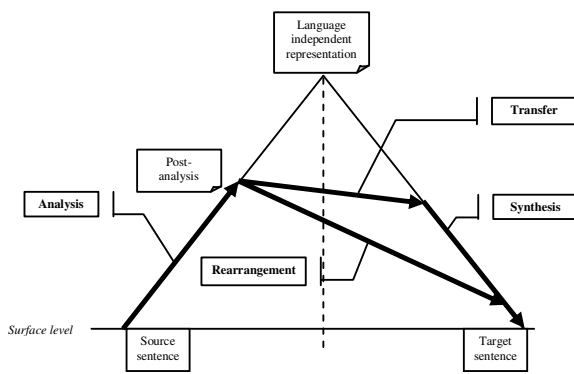


Figure 2: SYSTRAN's Two Module-Based Architectures

By shifting from the *current* paradigm to the *analysis + transfer + generation* paradigm, we aim to reduce the costs. *Transfer* is the only module that in principle is not reusable when developing a new language pair (but we will see below that it is not entirely true). *Analysis* and *generation*, which jointly represent almost 90% of the code, can be reused without modification, as shown in Figure 3, where non-reusable code is displayed in light grey.

### Reusing Existing Modules – Analysis

Except for our continuous maintenance effort, not much has to be done for these critical modules. We kept the existing analysis modules from existing language pairs since they are already stable for all the languages.

Modularity regarding language families already exists for Slavic and Romance languages.

For the Romance languages, we have a strong “Romance Language Trunk Parser”, feeding into small modules for the individual languages (Es, It, Pt). The German parser is also stable and usable for translations from German into any target language. In fact, it will be easy to create a

“Germanic Language Trunk Parser” for a number of closely related Germanic languages as soon as we develop more new language pairs with various Germanic source languages.

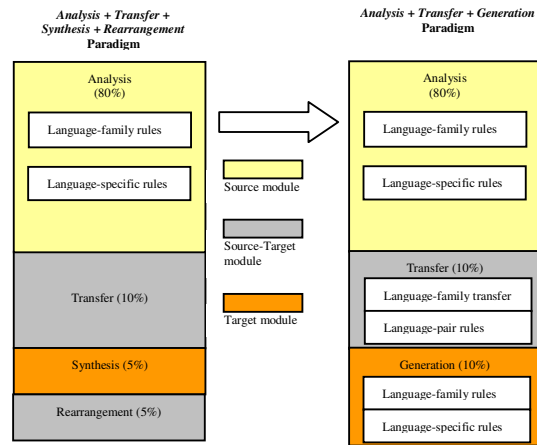


Figure 3: Paradigm Shift

No parser is ever perfect and we want to highlight that we had to make some minor changes to the system in order to make the analyses slightly more detailed (to go a little higher on the abstraction scale of Figure 1), as we worked on language pairs for which the source and the target languages pertained to more distinctive language families than before.

A simple example concerns disambiguation of verb forms. When translating from German to English, it did not matter whether the ambiguity infinitive/finite verb was correctly analyzed when the word appeared all by itself (“laufen” – “run”). But with Romance languages we needed to refine our disambiguation rule to get the infinitive form (“laufen” – “correr” rather than “corren”)

Analysis changes thus amount to refinement and corrections of issues that were overlooked because they were not as visible in the previous language pair’s configuration.

### Adapting Existing Modules – Transfer

With 12 new language-pair combinations, we might have looked at 12 new transfer modules. However, there would have been much repetition and overlap. We therefore decided to use a model based on our already existing “Slavic to English Transfer” module, which serves for the transfer of six Slavic source languages to the English target.

In order to factorize as much code as possible at the transfer level for our 12 new language pairs, three special sub-modules were developed: a *Romance-to-Romance* transfer, a *German-to-Romance* transfer and a *Romance-to-German* transfer.

This division is understandable with a bit of linguistic intuition: the pairs Spanish/Italian/Portuguese to/from Spanish/Italian/Portuguese do not require much work in transfer as the source and target linguistic structures are very similar (they are closely related to Romance languages). In contrast, the Spanish-to-English pair requires

some more effort on transfer, but not that much because modern Spanish shares most of its verbal tenses system with English (without considering the subjunctive). But Spanish/Italian/Portuguese and German are radically different, and more detailed transfer rules in each direction are needed if we want to handle word reordering correctly.

The transfer modules are written in traditional SYSTRAN rule format. Their output is an abstract representation that can be used directly by other traditional *synthesis + rearrangement* modules or via a new XML interface by the new *generation* modules described below.

One part of transfer code involves “lexical routines”. These are specialized rules triggered by some source language words and/or syntactic/semantic features. They are set up in such a way that one set of programs handles the analysis of the source language phenomena and outputs an abstract “meaning/structure identifier”, which can be used by target language programs. The con-

cept of a language family is important here as well. We apply the same division for the target lexical routines as for the rest of transfers: Romance-to-Romance, German-to-Romance, and Romance-to-German language families as needed.

### New XML-Based Interface

During this effort, we seized the opportunity of *creating a new generation module* to introduce a new format for passing information from the *transfer* to the *generation*.

After streamlining our architecture (Attnäs et al., 2005) we wanted to introduce a new open interface based on XML, according to the well-known framework of feature-based representations. To do this we introduced a mapping module from SYSTRAN’s proprietary, closed format into explicit, open features. This module is also responsible for synthesizing syntagms and clauses from SYSTRAN’s internal flat representation of word relationships (see Figure 4).

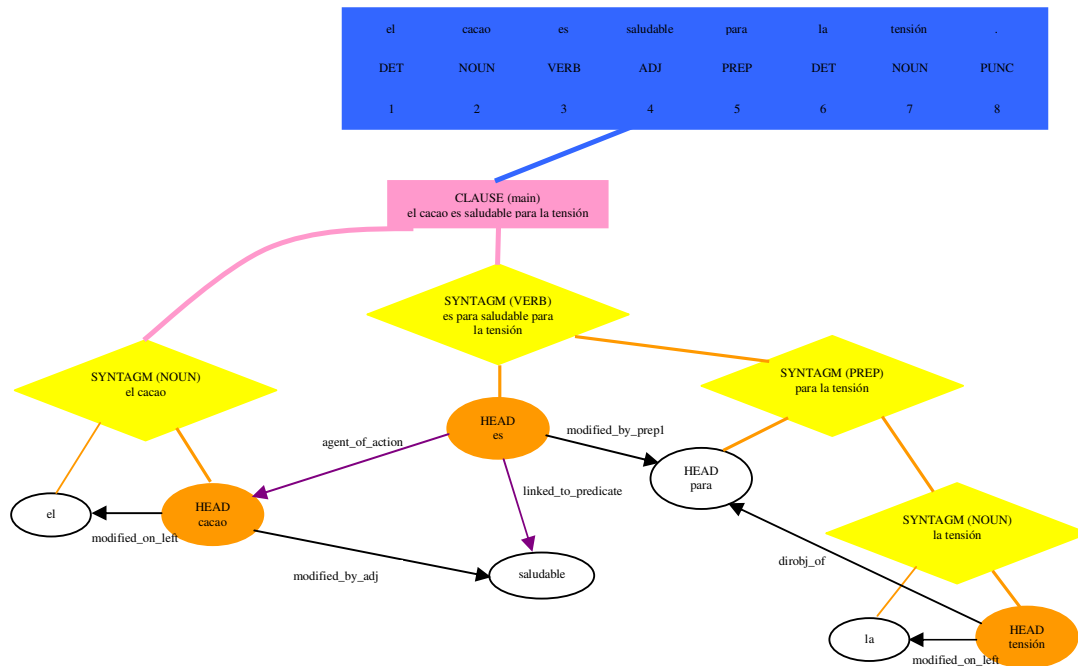


Figure 4: Syntactic Structure Passed to the Generation

### Creating New Modules – Generation

The four *generation* modules were developed from scratch. We decided to design a new framework that would enable the highest degree of code factorization to facilitate future development of new language generation modules. This framework is also based on the notion of linguistic families. We implemented abstract C++ classes representing the various syntactic constituents (word, syntagm, clause, and sentence). We equipped them with methods representing *language independent operations* of generation, such as: updating the structures (for instance, to ensure inflection agreement between a noun and its adjectives, or a verb and its auxiliaries), and the synthesis

of missing elements (for example when coming from a pro-drop language such as Spanish, where subject personal pronouns are not expressed, the German generation must synthesize them). We also have operations that reorder the various sentence’s constituents, and operations to perform a last refinement of the target tree (regarding punctuation and typography). We derived from these base classes a new set of classes for Western languages constituents. These classes are more “specialized” (as they contain much more linguistic knowledge). They override some methods or add new methods for generation operations common to all Western languages (such as management of verb complexes, negation, person pronouns...). From there, two new sets of classes were derived: one for Romance languages and one for Germanic languages. For

each of them we overrode existing methods to make them even more specialized, or we added new methods covering phenomena proper to the language family. Finally, we created the last sets of classes: Spanish, Italian and Portuguese classes inherit from the Romance set, while German inherits from the Germanic set. We further specialized the methods and enlarged the coverage by adding new methods.

We tried to implement as much code as possible at each level because increased specificity results in reduced code requirements (for instance, the Spanish, Italian and Portuguese classes each contain less than 500 lines of code focused mainly on verbal auxiliaries and management of enclitic pronouns). We also heavily relied on C++ native inheritance mechanisms to minimize the development effort.

### Interfacing with Statistics

Recently, much hope has been placed in hybrid approaches to increase the quality of Machine Translation systems (Koehn, 2006; Dugast et al., 2007). In order to test such hypotheses, we designed our new *generation* module in such a way that it can use *probability* or *confidence* information about what it is executing. This information can be adapted by linguists but the primary expectation is that it will be provided by corpus analysis tools and will automatically bind with corpus-based decision algorithms.

Although we have not yet used this feature the mechanism is available and ready for us to conduct first experiments in the forthcoming months.

### Lexicographic Issues

An even bigger effort has been devoted to lexicographic resources. They have been and remain the core of SYSTRAN's expertise for over 30 years and constitute huge amounts of data. The size of SYSTRAN's dictionaries ranges from 100,000 to 800,000 entries for each available language pair. These dictionaries are based on monosource / multitarget structures, i.e. one entry in the source language (e.g. English) is associated to all potential translations into all possible target languages. As a result, we have to manage large amounts of redundancies. Redundancies may happen in various situations: when multiple translations for the same concept in different dictionaries are available among dictionaries (e.g. the translations for "car" in English and "voiture" in French are the same into Spanish, Italian, Portuguese, German, etc.). We also have redundancies when symmetric dictionaries (e.g. Spanish-to-X and X-to-Spanish) contain some of the same entries but with different (monolingual) coding. Another case of redundancy occurs when there are compound entries for which the meaning is translated independently (in a consistent or inconsistent manner) from each simple compound word entry. Problems in maintaining large multilingual terminology is discussed in (Senellart et al., 2001).

In parallel, we heavily relied on SYSTRAN's proprietary IntuitiveCoding® technology (Senellart et al., 2003), in order to greatly reduce costs of dictionary coding and maintenance. IntuitiveCoding allows a user (in our case, the lexicographers) to enter only the very minimum set of

linguistic information to code bilingual (or multilingual) transfer dictionaries. Most of the information is "guessed" by the dictionary compiler from monolingual information (e.g. gender, number for nouns, or preposition government for verbs... but also more generally: inflection paradigms or homographs). Only the information that cannot be guessed, usually discriminating non-predictable idiomatic meanings, has to be entered by the user.

This forced us to shift from a dictionary model with 80% manually coded entries to a model with more than 95% of automatically guessed entries.

### Rapid Dictionary Development

All SYSTRAN language pairs use sets of dictionaries that consist of a main, extremely detailed dictionary with hundreds of thousands of bilingual entries and much smaller "update dictionaries" based on the IntuitiveCoding technology.

For each language pair, the main dictionary is derived from the monosource / multitarget dictionaries: e.g. the bilingual main dictionary for French-to-English is obtained from the multitarget French source dictionary. All entries are carefully coded by lexicographers in a proprietary and complex format. They contain all necessary information about the source words, the target words, and the bilingual link between source and target.

In contrast, the update dictionaries use IntuitiveCoding and are purely bilingual. The source and target monolingual information is obtained from the existing main dictionary (the list of all known words). When a new bilingual entry contains at least one unknown word its monolingual information is "guessed" from lexicographic rules and heuristics about morpho-syntax (Senellart et al., 2003). Very little information remains to be coded by the lexicographers in the bilingual dictionary and thus, many translations can be added in a short time, and the entries are much easier to maintain.

For the project, we decided to restrict the size of the "main" dictionary of all language pairs to the minimum, and entered as many entries as possible in what we called the "transfer" dictionaries using IntuitiveCoding technology.

In each language pair's main dictionary we kept only the "grammatical words": prepositions, pronouns, particles, conjunctions and very fundamental verbs: copulas, auxiliaries, linking verbs and verbs involved in very common periphrastic idioms (such as "ir" in Spanish). We also kept words with very heavy homographic phenomena that require complex manual coding. In total, these dictionaries contain around 1,500 entries for each of the four source languages. We handle them in the traditional monosource / multitarget format and derive bilingual dictionaries from them for each language pair.

These dictionaries were initially obtained by applying cross-dictionary transitivity in order to obtain a first set of bilingual entry candidates, making bold assumptions on the use of a pivot language, inspired by (Mangeot & Kuroda, 2003). For instance, the entries for Spanish-to-German were obtained by selecting the English translations from the existing Spanish source multitarget dictionary, and then looking for their translations into German in

the English source multitarget dictionary. Human editing was of course needed.

The transfer dictionaries based on IntuitiveCoding technology were built incrementally, and contain “full” words: nouns, verbs, adjectives, and adverbs, as well as idiomatic sequences. We ran the analysis on corpora from a wide selection of electronic newspapers (for each source language, this corpus ranged from 150 Mb to 250 Mb) in order to capture all Not Found Words (NFWs), i.e. all the words whose monolingual information is known by IntuitiveCoding but for which we have no translated equivalent (no mapping from source to target).

We first selected the 5,000 most frequent entries (by lemma frequency) for Spanish source. We applied the same kind of dictionary transitivity with English as a pivot in order to suggest translation candidates into German (Spanish-to-English + English-to-German), into Italian (Spanish-to-English + English-to-Italian) and into Portuguese (Spanish-to-English + English-to-Portuguese). These entries were then sent to a translation agency for review and post-editing. We asked the agency to provide each entry’s Part-of-Speech and *domain* (general, science, business, legal...), the most general meaning(s) indexed by its lemma, and to make sure that the translation meanings from Spanish into the three other languages were aligned.

We used the agency’s feedback to produce initial “transfer” IntuitiveCoding dictionaries. We added them to the “main” dictionary of each language pair and ran a corpus analysis with the goal of obtaining the 20,000 most frequent NFWs.

### Fully Multilingual Dictionaries Management

At this step, we started facing specific issues concerning the multilinguality of our transfer dictionaries. By asking the translation agency to provide only the “most general” meanings for all Spanish source entries (or at most the two most general meanings), in addition to keeping all translations aligned, it was possible to *reverse* the dictionary. This meant that we could shift the column order of the list. In other words, Italian was to become the source column and all the other columns became target languages.

The first problem with this assumption was that we obtained valid translations from Italian into German, Spanish and Portuguese, but not necessarily for the *most frequent* entries in Italian. We could not assume that the distribution of lemmas in Spanish and in Italian were the same, even for (hypothetical) perfect translations. As a result we started *reversing* our transfer dictionaries and running the translators on the Italian, Portuguese, and German corpora, searching for the most frequent NFWs in each language. More precisely, we first searched for Italian NFWs, sent them to the translation agency and used their feedback to search for Portuguese NFWs, and then did the same for German NFWs.

Searching for German NFWs was a bit more complex because of the typical German phenomenon of compounding words, e.g. “ministro de la guerra” (Spanish) → “Kriegsminister” (German, only one word). In parallel we ran a NFW search with our new translators, and also ran

the existing German-to-English translator tagging all the compound words it knew on the same corpus. We then removed all the entries already present in our German source “transfer” dictionary *reversed* from Spanish, Italian and Portuguese sources.

By the end of this iterative process, we reached a total of more than 40,000 aligned entries in the four language pairs.

The next important issue raised was *conflicting synonyms*. For instance, different source words in Spanish may have the same “general” translation into German (see the noun “Falte” in Table 1):

De	Es	It	Pt
Falte	arruga	ruga	ruga
Falte	doblez	piega	dobrez
Falte	plegamiento	sdoppamiento	dobramento
Falte	repliegue	ripiego	recolha

Table 1: Synonyms for “Falte” (German source)

This can happen for more than one reversed source at the same time. For example, see the adjective “alt” (German) and “vecchio” (Italian) in Table 2:

De	It	Es	Pt
alt	vecchio	añejo	envelhecido
alt	vecchio	viejo	velho

Table 2: Synonyms for “alt” / “vecchio”

We extracted all of these *synonym clusters* from the dictionary and asked the translation agency to select a unique “main” meaning in each cluster, accounting for source languages (only German in Table 1; German and Italian at the same time in Table 2), the Part-of-Speech, and the domain of the entry.

By adding a special comment alongside each entry, we were able to track the “main” meaning for each source language, instead of looking up “alternative meanings”. We maintained the multisource / multitarget dictionary and comments as a whole piece which we refer to as the “multilingual matrix”, and extracted monosource / multitarget IntuitiveCoding dictionaries for each possible source language, using only the main meaning of each source word.

Further research on this issue will be conducted at a later stage. The initial goal was to reach a dictionary size comparable to that of the existing language pairs (i.e. at least 100,000 entries), and refined coding based on the surrounding context (mainly using verb framing).

### Corpus-Based Expressions Extraction

In addition to the *main dictionary* and to the *transfer dictionary* just discussed, we built an *expression dictionary* for each language pair. These dictionaries come from the statistical extraction of phrase pairs in aligned corpora such as Europarl (Koehn, 2005). The system considers these entries as alternative meanings with low priority to prevent it from discarding translation choices already made in the *main* and *transfer* dictionaries.

A communication regarding the impact of such dictionaries on the performance of the translators will soon be published

Compared to “phrase-tables” used in phrase-based statistical decoders, the dictionaries used by SYSTRAN’s rule-based translators have to be linguistically coded. In other words, each entry must be linguistically relevant, and the Part-of-Speech and inflection features must be identified.

To perform this we used different levels of linguistic knowledge to code relevant entries, and excluded those deemed irrelevant. The underlying principle of these successive filters allows for the use of more computationally intensive filtering processes as the volume of candidate entries is reduced. Based on this concept, we successively applied massive preprocessing (based on forbidden boundary words), monolingual coding and bilingual coding. At each step, an entry could have been discarded.

The linguistic coding was performed via the SYSTRAN Coding Engine using the IntuitiveCoding technology (Senellart et al., 2003), with more restrictive coding patterns. These patterns were selected by the linguistic team, using pattern reviews with example entries. We minimized the required effort by:

- Revising the entries by high frequency and fine-tuning the corresponding patterns;
- Generalizing the resources among a language family

The impact on translation quality was evaluated for each of the new language pairs. The BLEU score improved around 1.5 points on the WMT2007 Europarl test corpus. A complementary human evaluation proved that the quality gain is promising considering the relatively small size of these dictionaries.

## Outcome

The effort invested in this project involved two linguistic developers (responsible for syntax aspects and the design and implementation of the linguistic families framework), and four lexicographers over a 12 month period. All participants worked slightly over 50% of their time on this project.

The goal of this project was to simultaneously create a “high” number of new language pairs with a rule-based approach by reusing existing code and resources with guaranteed incremental quality level: our first objective was to deliver the 12 new fully functional translators covering all the fundamental morphologic and syntactic aspects of the four involved languages by late Spring 2007. The second objective was to increment the translation quality by end of 2007 to a level comparable with our most recently developed language pairs developed in a classical way.

Each translator comes with dictionaries (*main* and *transfer*) composed of approximately 100,000 entries (lemmas), which is comparable to our most recently developed language pairs. In addition, we provide a statistical dictionary of expressions (ranging from 10,000 to 60,000 expressions).

We shifted from the *analysis + transfer + synthesis + rearrangement* paradigm to the *analysis + transfer + generation* paradigm for these new language pairs. This allows us to reuse much more code and to reduce the size of a specific language pair’s dependent code to the mini-

um. The concept of linguistic families has expanded from *analysis* to *transfer* and most importantly to *generation*, where high-level Object Oriented Programming mechanisms are in responsible for applying linguistic rules and factorizing the code to the maximum.

We introduced open interfaces in our MT systems that use of statistical information and represent the results of *analysis* and *transfer* in a friendlier exchangeable format for NLP use.

For the first time we started working with fully multi-source / multitarget dictionaries to rapidly increase the dictionary size, and to ensure that all dictionaries are aligned and easily maintained.

During the project we observed the fast evolution of quality (through automatic and regular human evaluation) for each of these language pairs focused on the coverage of generation grammar and on the lexical coverage.

## Open Issues

During this project we faced some important design and technical problems. We solved them as quickly as possible as our goal was to obtain fully functional translators, but we plan to work on some issues before starting a new project to develop 18 new translators during the next year using the same techniques..

The two main open issues follow.

First, we want to reach a higher abstraction level of analysis in order to further reduce the importance of *transfer*. We also need to more clearly define the boundary between *transfer* and *generation*, so that the *transfer* only contains what is not be handled by the *generation*.

The second, issue relates to the way we cope with synonym clusters as described above. Today only a “main” meaning in each cluster is selected and we do not use the others. This is neither an ideal situation nor the best solution. As a first step, we would assign “priorities” to the different meanings in a cluster: preferred meaning, second preferred meaning, and so on. A better approach would be to associate some kind of unique identifier to each meaning, such as a Wordnet ID, so that there is only one ID on each line of the multilingual matrix.

This implies that we should be able to leverage contextual information (syntactic, semantic and domain context) in order to better discriminate between the different meanings. Today many of the meanings in our clusters are not true synonyms. More contextual information would eliminate these artificial synonyms.

## Acknowledgements

Our best thanks go to the translators of the translation agency Consonant S.L.

We also want to thank the members of the statistics team at SYSTRAN: Loïc Dugast and Tania Samoussina.

## Bibliographical References

Attnäs M., Senellart P., Senellart J. (2005). Integration of SYSTRAN MT systems in an open workflow. In Proceedings of the Tenth Machine Translation Summit (pp. 211-218). Phuket, Thailand.

- Dugast L., Senellart J. & Koehn P. (2007). Introducing Corpus-Based Technology into SYSTRAN Rule-Based Translators. System description for ACL2007 Shared Task on Statistical Machine Translation (WMT07). Submission to ACL 2007, Prague, Czech Republic.
- Foster G., Gandrabur S., Langlais P., Plamondon P., Russel G. & Simard M. (2003). Statistical Machine Translation: Rapid Development with Limited Resources. In Proceedings of MT Summit 2003 (pp. 110-117). New Orleans, Louisiana.
- Koehn P. (2006). Statistical machine translation and hybrid machine translation (contribution to panel on "Hybrid machine translation": <http://www.mt-archive.info/AMTA-2006-Koehn.pdf>). AMTA 2006: Seventh Conference of the Association for Machine Translation in the Americas, "Visions for the Future of Machine Translation", August 8-12, 2006, Cambridge, Massachusetts, USA.
- Koehn P. (2005). Europarl: A Parallel Corpus for Statistical Machine Translation. In Proceedings of the Tenth Machine Translation Summit (pp. 79-86). Phuket, Thailand.
- Mangeot M. & Kuroda K. (2003). Divergences interlinguistiques dans le dictionnaire multilingue Papillon. In Proceedings of the First International Conference on Meaning-Text Theory (pp. 33-42). Paris, France.
- Och F.J. & Ney H. (2001). What Can Machine Translation Learn from Speech Recognition? In Proceedings of the Eighth Machine Translation Summit Workshop (pp. 26-31). Santiago de Compostela, Spain.
- Papinen K., Roukos S., Ward T., Zhu W.-J. (2002). BLEU: a method for automatic evaluation of machine translation. In Proceedings of the 40<sup>th</sup> Annual Meeting of the Association for the Computational Linguistics (ACL) (pp. 311-318), Philadelphia, USA.
- Senellart J., Yang J. & Rebollo A. (2003). SYSTRAN IntuitiveCoding® Technology. In Proceedings of the Ninth Machine Translation Summit (pp. 346-353). New Orleans, USA.
- Senellart J., Plitt M., Bailly C. & Cardoso F. (2001). Resource alignment and implicit transfer. In Proceedings of the Eighth Machine Translation Summit Workshop (pp. 26-31). Santiago de Compostela, Spain.
- Sérasset G. & Boitet C. (2000). On UNL as the future "html of the linguistic content" & the reuse of existing NLP components in UNL-related applications with the example of a UNL-French deconverter. In Proceedings of the 18th conference on Computational linguistics (pp. 768-774). Saarbrücken, Germany.