WEBTRAN: A Controlled Language Machine Translation System for Building Multilingual Services on Internet

Aarno Lehtola
Jarno Tenni
Catherine Bounsaythip
Kristiina Jaaranen
VTT Information Technology
Finland

Abstract

This paper presents how to embed a fully automatic controlled language (CL) translation into online services on internet. CLs are natural sublanguages with specific domains, limited vocabulary, restricted syntax, and minimised ambiguities. The CLs enable practical and cost-effective language processing solutions, e.g., for machine translation and text mining. We have designed a generic CL translation software, Webtran, which is intended for building multilingual services on Internet. We describe the Augmented Lexical Entries formalism provided for a controlled language definition. These entries can be simple, single or multilingual, word or idiom descriptions. They may define dependency relations, control semantic admissibility, specify language repair, and even depict high-level document syntax. Webtran includes a language-modelling tool which supports the ALE-formalism and implements human-assisted machine learning to intensify the modelling. We present our experiences of adapting Webtran in translating product articles in an online catalogue from Swedish to Finnish. Regardless of the test languages, the Webtran software is in principle language independent, and tentative tests have been carried out with Estonian, French and English as targets. We also discuss here the embedding of Webtran to the overall catalogue production process.

1 Introduction

A controlled language (CL) is a subpart of a human language limited to a specific domain of discourse (Kittredge 1987). It is characterised by a limited vocabulary and restricted syntax. Controlled means that ambiguities are minimised in the texts. This enables practical and cost-effective solutions, e.g., for machine translation and text mining.

Nowadays, due to the fast development of information society, there is a growing interest in using simplified language to author source text (Joscelyne 1998). The approach has been successfully used to improve the quality of translation, as well as the readability and maintainability of the original texts (Kittredge 1987, Adriaens and Macken 1995, Douglas and Hurs 1996, Schwitter and Fucchs 1996, van der Eijk 1998, Whitelock and Kilby 1995). Currently, the truck company Scania is implementing ScaniaSwedish for the preparation of truck maintenance manuals in controlled Swedish (Almqvist and Sågvall-Hein 1996, Sågvall-Hein 1997).

The service providers in Internet may benefit considerably of the CL technology, which could lower the costs per client in the multilingual services. The CL-technology is particularly suited for building multilingual online mail-order catalogues. While the structure similarity of product descriptions may be tiresome for a human translator, there is a good reason to use an automatic translation system (Hutchins and Somers 1992). Moreover, the domain specificity of product descriptions lends very much to the use of controlled languages. By speeding up the catalogue maintenance process the technology shortens time-to-market and improves the overall competitiveness of the service provider. Reduced translation costs open visions for networking with other providers to increase repertoire.

We have designed a generic CL translation software, Webtran. A mail-order company is currently adapting it in their catalogue production process to translate product descriptions from Swedish to Finnish. The sample text in Table 1 illustrates the style of the descriptions. They have a specialised vocabulary and a noun phrase dominated syntax. The texts often lack a clean sentence structure and main verbs. Very few articles or pronouns are used. The descriptions consist mostly of noun phrases, which may

have quite complicated structure. Ellipsis is often present with conjunctive structures. The product descriptions contain mainly factual technical information.

Jacket
With feathery polyamide filling material. Bright colours and shiny surface. A lace and a stopper on the removable hood. A rib at the end of the sleeve. The lower edge has a lace with a stopper. A strong plastic zip on the front and two pockets with a zip. The lining and the outer surface are 100% polyamide. Washing 40°. 154 - 3905 lemon-yellow 154 - 3906 cherry.

Table 1: A sample product description.

With the translation software, the original catalogue needs to be maintained in only one language. Accurate translations can be provided in real time for customers with other languages. The translation process can be embedded in the overall service system so that its functioning is completely transparent to the end-user during the mail-order session. The approach requires that the catalogue maintenance process must include a language check to guarantee that the grammar of the controlled language is fulfilled.

In a multilingual system, post-editing is an expensive task, as it requires checking of every text in the target language, while pre-editing requires only one check of the source text. CL checking tool helps the writer at the pre-editing phase to adapt the source text to the allowed vocabulary and syntax. In a CL system, pre-editing is a necessity and enables fully automatic accurate translation later on without post-editing.

In this paper we first introduce the formalism we have developed for the modelling of controlled languages. After that we discuss the adaptation of the controlled language software in a catalogue service on Internet. This includes embedding the software into the catalogue maintenance and online service processes. We present our language modelling methodology, the modelling tools, and the experiences with the test catalogue. We also outline the machine learning methods we have implemented for grammar acquisition from sample texts.

2 Augmented Lexical Entries

In the Webtran approach, Augmented Lexical Entries (ALEs) are used to carry the linguistic information needed to both define and translate controlled languages (Lehtola et al. 1998). The formalism can be characterised by the following:

1. Describing simple phenomena is simple.
2. Complicated phenomena can also be described.
3. Declarative and intuitive notation.
4. A uniform way of representing phenomena on the different levels of language.
5. Bilingual or multilingual non-directed entries.
6. Automated or machine supported language modelling available.

augmented_lexical_entry ::= **[** entry_*name*
pattern..
opt_message
opt_repair **]**

entry_name ::= *name* . *number_index*

name ::= *hierarchical_name_w_dots_betw_parts*

pattern ::= **[** *opt_language_id constituent_def..* **]**

opt_message ::= ε | **[message** *string_w_opt_binding* **]**

opt_repair ::= ε | **[repair** *string_w_opt_binding***]**

constituent_def ::= *constituent_def******

constituent_def ::= *constituent_def***..**

constituent_def ::= < *constituent_def..* >

constituent_def ::= *opt_regent_mark*
opt_lexeme
opt_binding
opt_feature_constraint

opt_language_id ::= ε | *ISO_std_lang_identifier* |
~ *ISO_std_lang_identifier*

ISO_std_lang_identifier ::= **ee** | **en** | **fi** | **fr** | **se** | ⊕

opt_regent_mark ::= ε | ^

opt_lexeme ::= ε | *lexeme* | *tag* | *name*

opt_binding ::= ε | *binding*

opt_feature_constraint ::= ε | **{** *feature..* **}**

binding ::= **(** *variable_name* **)** | **(^)**

feature ::= *feature_value* | *property_type binding*

Table 2: The syntax of the augmented lexical entries.

The general form of the augmented lexical entries is shown in BNF notation in Table 2. Nonterminals are in italics. The symbol ε means empty and ⊕ denotes any ISO standard language code. The number of languages used in an ALE is not restricted. An ALE can be mono- or multilingual. A monolingual entry defines either an

allowed or prohibited language expression without any translation information. In the latter case, it may contain an interactive message and a repair instruction for the user of the checking tool. Monolingual language definitions could be used, for instance, to ensure consistent language when producing manuals in one language only. A multilingual online catalogue service would also contain multilingual entries to provide the translation relations.

(a)

```
[footwear.word.27
   [se allväderskänga]
   [fi jokasäänkenkä]
   [en all weather shoe] ]
```

(b)

```
[price.tax.4
      [se inkl. moms]
      [fi sis. alv]
      [en incl. VAT] ]
```

(c)

```
[cloth.material.composition.3
   [se  ^(A){product} i tag_percentage(X)
        (B){material}]
   [fi ^(A){product} tag_percentage(X)
        (B){material ptv}]
   [en  ^(A){product} of tag_percentage(X)
        (B){material}]
]
```

(d)

```
[cloth.property.1
   [se  (A){adj clothProp gender(B) number(B)}
        ^(B){noun cloth}]
   [fi  (A){adj clothProp case(B) number(B)}
        ^(B){noun cloth}]
   [en  (A){adj clothProp}
        ^(B){noun cloth}] ]
```

Table 3: Examples of augmented lexical entries: (a) a simple word correspondence, (b) an idiomatic surface expression, (c) a generalised entry with numeric value preserved, and (d) a generalised entry with semantics and interdependence of the words denoted.

An ALE contains a *pattern* for each language it covers. The ISO language codes are used to mark these language-specific *patterns*. The number of languages in an ALE is not limited. A *pattern* also contains the *constituents* of the corresponding language expression in their matching order. If the specified *constituents* are bounded by angle brackets, they may appear in any order. A *constituent* followed by an asterisk (*) can have zero to more occurences and a constituent having two consecutive dots (..) right after can have one to more instances. *Constituent* definitions can specify surface form words, they may be *bindings* and/or a set of morphological or semantic *feature constraints*, or they may refer to other entries. A *binding* is a reference to another constituent stated either in terms of a *variable name* scoped lexically by the entry, or stated using the caret (^) referring to the *constituent* marked as the *regent* and scoped by the parse context.

Tables 3, 4, 5 and 6 contain examples of ALEs. Later on the small letters refer to these examples. In their basic form ALEs are elementary correspondence templates between surface expressions. For instance, the entry (a) is just a simple word correspondence definition and (b) specifies translation for a specific idiom of the controlled language. These entries match just the presented single patterns in the source texts and support all translation directions. They illustrate the characteristic 1 of the formalism "Describing simple phenomena is simple".

In their more complicated form the ALEs can specify generalised patterns of adjacent expressions that will be treated in the further processing as single units. These generalised entries can not be associated with any particular word but with a class of words. This class is specified by *feature constraints* written in curly brackets. The example (c) translates expressions like "shirt of 100% cotton" ("skjorta i 100% bomull" in Swedish and "pusero 100% puuvillaa" in Finnish). It specifies the semantic categories of the words and the preservation of the percentage figure using a *variable*. In ALE formalism, *variables* have a capital character in their beginning. They share the single-binding behaviour with Prolog variables and carry constituents as their values.

The example (c) is one to illustrate the characteristic 4 of the formalism "A uniform way to represent phenomena on different levels of language". Syntactic and semantic constraints can be presented in the same rules. During processing these are considered simultaneously. The approach differs from the so called "stratificational processing models", where language processing is divided into consecutive phases along to the language levels, e.g. morphology, sentence parsing, logico-semantic analysis, transfer etc. In our understanding the stratified models bring extra complexity into the language modelling, as a linguist doing modelling would need to carefully thread together the levels vertically while specifying the grammar.

The example (d) covers expressions of cloth properties, such as "comfortable blouse" ("bekvämt linne" in Swedish and "miellyttävä pusero" in Finnish). For example, in its Swedish pattern the rule specifies constraints for two consecutive words; an adjective and a noun. The adjective must belong to the semantic category *cloth property* and the noun to the semantic category *cloth*. The adjective must be inflected in the same number and gender as the noun.

When the source part of an entry is found in the text, the rule controls the formation of corresponding constructs in the target language. The word translations are not defined explicitly but are retrieved from a separate domain specific lexicon. In the example (c) in Table 3 the last word of the Finnish pattern is the only one that is inflected (case overridden to *partitive*), the rest of the words appear in their nominative form and preserve the number of their correspondent. The variables are bound to whole constructs and can be used for specifying word order reversals, if such were needed.

If ALEs are properly defined, they can be used non-directionally to cover all translation directions in a single entry. Similar non-directional reading also appears, e.g., in unification grammars, like lexical-functional grammar (Shieber 1986). The entries in Table 3 function in multiple directions.

When more descriptive power is needed the entries can also capture hierarchical sentence structures by specifying a dependency grammar. In the entries the words marked with a caret will be considered the regents of their idiom. While a dependency parse tree is constructed, the marked word is the root of the corresponding subtree and will have the other words of the idiom as its subordinates.

By marking the regents the grammar is turned into a forest of partial dependency parse trees of depth one. The use of such grammar employs parsing algorithms that derive the parse tree fulfilling the given constraints.

The entries in Table 4 generalise the entry (d) to cover also conjunctive lists of cloth properties. The entries (e) and (f) rewrite and partition the entry (d) into two entries. *Bindings* referenced using the caret, get bound to the regent constituent of the construct. This way the rules (e) and (f) include the interdependencies of constituents stated in the entry(d). The entries (g) and (h) specify language independently the recursive structure of conjunctive lists.

The entries in Table 4 also illustrate the idea of explicitly marking the regent constituents. In traditional dependency grammar the topology of the parse tree is implicitly defined in the relation specifications. In case of a long conjunctive list, the result would be a deep parse tree, which complicates further processing. In fact often there is a separate tree flattening processing added. Similar phenomena happen also with phrase structure grammars where the production rules specify the tree topology. In our approach all language expressions fulfilling the entries in Table 4 produce a parse tree of depth one. The entries in Table 4 demonstrate the characteristic 4 of the formalism "Complicated phenomena can also be described".

```
(e)
[cloth.property.2
   [se  property.expr{clothProp}
        ^(B){cloth}]
   [fi    property.expr{clothProp}
          ^(B){cloth}]
     [en  property.expr{clothProp}
          ^(B){cloth)] ]
```

```
(f)
[property.expr.1
     [se  (A){adj prop gender(^) number(^)} ]
     [fi (A){adj prop number(^) case(^)} ]
     [en  (A){adj prop} ] ]
```

```
(g)
[property.expr.2
   [property.expr.2  tag_comma  property.expr.3]]
```

```
(h)
[property.expr.3
   [property.expr.1 {conjAND} property.expr.1] ]
```

Table 4: Examples of rule references: (e) and (f) partition the entry for cloth properties into two entries, and (g) and (h) generalise this using language independent entries to cover lists of properties delimited by commas and a conjunctive.

All of the entries this far evidence the characteristic 3 of the formalism "Declarative and intuitive notation". The entries of in the first table are easy to understand and to write also by professional translators. The ALEs provide a constraint programming way of specifying the grammars.

The ALE formalism does not take any position to which algorithm is used to fulfil the constraints. In fact, multiple algorithms may be used. The hierarchic naming convention enables to modularise the grammar and to use different control strategies in different sets of entries. Many practical strategies and algorithms have been published for dependency parsing. Elementary two-way finite-automata are considered for dependency parsing in (Nelimarkka 1984). The article (Jäppinen 1986) formalises dependency grammar in terms of partial trees of depth one and presents an algorithm for those. The article (Valkonen 1987) employs a blackboard mechanism for the book keeping of the partial constituents when parsing with two-way finite automata. Non-deterministic dependency parsing is handled in (Jäppinen 1988 and Arnola 1998). The approach we are implementing is based on two-way automata and application of proper ordering in recognising the hierarchic structure. We are also investigating a hybrid approach where the strategy would change along the properties of the entries. Such a hybrid approach has been implemented for context-free grammars (Hyötyniemi and Lehtola 1989). For run-time use, the Webtran software compiles the ALEs into Prolog clauses, which in turn can be compiled using a Prolog compiler. During the compilation various automata optimisations are possible.

```
(i)
[correct.ellos.3
   [~se (kardborrstängning(A)]
   [~se (kardborreförslutning(A)]
   [~se (kardborrknäppning(A)]
   [~se (kardborreknäppning(A)]
   [message Use the correct synonym
            “kardborrestängning” instead of
             word(A)]
   [repair  kardborrestängning(A)]
```

```
(j)
[correct.ellos.7
   [~se storlekar(A)tag_size(X)]
   [~se stl(A)tag_size(X)]
   [message  Word (A) is not allowed
             in this context]
   [repair storlek(A)tag_size(X)]
]
```

```
(k)
[correct.ellos.8
   [~se (i (A){property} tag_comma
        (B){property} (C){model}]
   [message  Sentence structure not allowed.
             Use word "och" instead of ","]
   [repair   i (A){property} och
             (B){property} (C){model}] ]
```

Table 5: Examples of correction entries: (i) correct version of synonyms should be used, (j) a prohibited word in the context, and (k) conjunctive word instead of a comma should be used.

The controlling of the language is important, but it is a difficult task. For this purpose the ALE formalism provides notation for specifying also prohibited language expressions. These correction entries can include *message parts* and *repair parts*, which specify user interactions for the checking tool. They thus instruct the author to map from a natural language to a controlled language. The correction rules cannot have full coverage of the natural, unrestricted language. If the machine could understand unrestricted language, there would not be any need for the controlled language. But there are still some commonly repeated mistakes that can be corrected, like which one of a set of synonyms should be used, or in which context the words should be used.

Table 5 contains examples of correction entries. The entry (i) specifies the correct synonym to be used in the catalogue. The entry (j) indicates that the use of words "*storlekar*" and "*stl*" is not allowed in the beginning of size number/list, but instead the word "*storlek*" should be used. If the prohibited sentence structure is found in the checking phase, the message "*word storlekar is not allowed in this context*" is shown to the writer with the repair suggestion. The user can then accept the replacement "*storlek*". The entry (k) specifies that a conjunctive word is obligatory in the end of a list instead of a comma.

Our checking tool handles both sentence structure and synonym usage. Corrections are specified mainly to repetitive errors. Unique errors are pointed out without repair suggestion as observed in the ordinary processing with positive entries.

.

```
(l)
[description.cloth
     [  ^description_heading
        < cloth.model cloth.material >
        cloth.washing
        product_code_and_colour..
        cloth.size..
        price.. ] ]
```

Table 6: Specifying structure of a cloth description.

Table 6 demonstrates the application of ALEs for describing conceptual structure or document syntax of product description articles. The entry can be used to check the semantic admissibility of a cloth description. For specifying conceptual models we have an ontology editor (Kankaanpää 1999).

3 Embedding Webtran into a Catalogue Service on Internet

Figure 1 shows the architecture of the multilingual product catalogue system, which we have implemented as a test case for our CL technology. Webtran Software consists of three parts:

1. *Webtran Modelling Tool* is used by the designers of a controlled language, e.g. professional translators, to specify approved vocabulary and terminology, sentence structures, and translation correspondences (see Section "Language Modelling"). It also includes automated learning methods which make the language definition process easier (see Section "Supervised Machine Learning Methods").
2. *Webtran Checking Tool* is a controlled language grammar checker used by the editors while maintaining the text database to check the syntactic and semantic admissibility of the input product descriptions.
3. *Webtran Translation Engine* is a fully automatic machine translation program used either during the database maintenance or in real-time as the end-users access product descriptions.

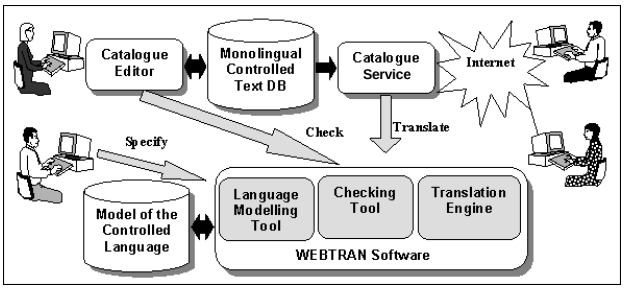

Figure 1: Architecture of the multilingual product catalogue test system.

Webtran has been applied in providing multilingual views to product descriptions of women's clothes on the WWW (Figure 2). In the pilot system, product descriptions are maintained in one CL only (a sublanguage of Swedish). The end-users get their translated descriptions through the Information Service in the language of their preference. The first target language is Finnish and preliminary tests have been done with Estonian, French and English as well. Table 7 shows product descriptions in five languages, translated by Webtran using the current language specifications. The original text is in controlled Swedish. This far the prices are not converted.

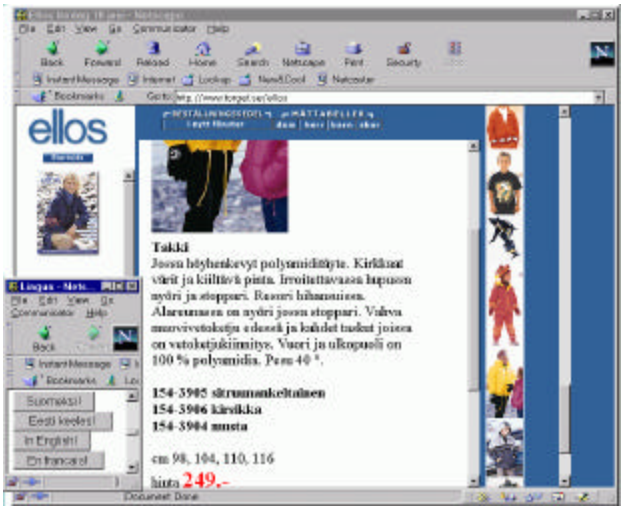

Figure 2: The WWW interface of the multilingual catalogue test system.

The original frames and links on the left-hand side of the screen have remained unchanged (in Swedish) as the pilot is used for demonstrating the translation capability and it will not be seen in the final end-user interface. So the translation has been done in real time, and the user can now continue to browse other products and get the interesting ones translated.

Swedish (source language)	Cardigan Rak modell med snygg mönsterstickning med broderier på framstycket. Ribbstickad krage och kant i ärmslut och nederkant. Längd ca 64 cm. Kvalitet av 70% akryl/30% ull. Handtvätt. 156-3556 Gråmelange Storlekar 34/36, 38/40,42/44 Pr styck 449,-
Finnish	Neuletakki Suora malli, jossa tyylikäs kuvioneulos ja brodeeraukset etupuolella. Ribattu kaulus ja reuna hihansuissa ja alareunassa. Pituus n. 64 cm. Neulos 70 % akryyliä / 30 % villaa. Käsinpesu. 156-3556 harmaameleerattu koot 34/36, 38/40, 42/44 hinta 449,-
English	Cardigan Straight model with stylish figure knitting with embroidery on the front. Rib collar and edges in cuffs and in hem. Length approx. 64 cm. Fabric 70 % acryl / 30 % wool. Hand wash. 156-3556 melange grey sizes 34/36, 38/40 , 42/44 price 449,-
French	Cardigan Modèle droit avec maille raffinée avec broderies devant. Col polo et avec bord côtes aux manches et à la base. Longueur env. 64 cm. Maille 70 % acrylique / 30 % laine. Lavage à la main. 156-3556 mélange de gris tailles 34/36, 38/40, 42/44 prix 449,-
Estonian	Trikoojakk Sirge mudel millel on stiilne musterömblus ja muster esiküljel. Traageldatud krae ja käiste ja hölmade ääred. Pikkus u. 64 cm. Kangas 70 % akrüüli / 30 % villa. Käsipesu. 156-3556 hallikas suurused 34/36, 38/40, 42/44 hind 449,-

Table 7: Sample translations of a cloth description from controlled Swedish into four other languages.

4 Language Modelling

The first task with the pilot system was the controlled language modelling for the domain of mail-order catalogue article, as found in the sample catalogues. The modelling included the definition of the vocabulary and the allowed sentence structures. Also the correction entries were created to guide the authors in writing according to the controlled language specification. Disambiguation needs

caused extra constraints to be stated for the created CL. The initial language model was created manually.

In the pilot system, the definition process focused on product descriptions of women's clothes. It was found out that similar or almost similar phrase and sentence structures were repeated throughout the catalogue. Even though the language used in the descriptions is rather simple, the problem is that it is not controlled in any way, and ambiguities are possible.

The actual definition work started by entering bilingual (Swedish, Finnish) ALEs that only contained phrase templates in surface form. After processing all the women's clothes descriptions (over 100 in number) in one catalogue, the language definition had over 700 ALEs. There were enough entries to translate most of the women clothes descriptions from Swedish to Finnish.

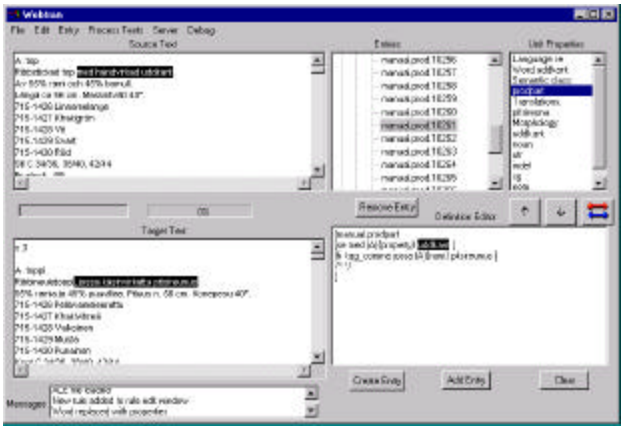

Figure 3: The user interface of the language modelling tool.

The next step was to generalise these ALE entries in order to limit the amount of possible new entries. Generalised entries matched both the older and newer catalogue and at the same time the overall number of entries diminished.

The following task with the second catalogue was to find the parts that did not comply with the existing controlled language definition. Even the smallest variation in words or sentence structures in the new text would cause the surface form entries to be incompatible. The generalisation started with the most implicit examples, where the words to be generalised had definite semantic classes. For instance, the semantic class of colours was used for generalising phrases to cover all colour choices. Gradually, more semantic classes were added, which allowed more generalisations in the entrybase. Generalisation by using semantic classification is efficient as it can be done gradually and the language definition can be kept operating all the time. The semantic classification is simple as the words are classified into just one class, which is presented by the class name.

Figure 3 illustrates the five pane browsing and editing interface of Webtran Language Modelling Tool. In the source text window and the target text window the corresponding texts excerpts as found by the alignment algorithm are inverted. The tool supports quick working by enabling modeless moving between the panes along the moving of the modelling focus.

5 Test Experiences

The original entries (containing the 700 surface form ALEs) were based on "autumn 97"- catalogue. This entrybase was generalised by using web catalogue "autumn 98". The entries were then tested with two new catalogues ("autumn 98", "spring 99") which had not been used while creating the controlled language definition. As the examples in Table 5 show, the generalisation of the entrybase enables the system to translate acceptably even texts that had not been used in the definition process whereas the surface form entries would not have been capable of handling such texts.

Some of the errors in translations are repetitive and it is easy to diminish the error rate notably with only a few new entries. For instance, if two new most-needed entries are added to the generalised entrybase, the number of errors in the words of autumn 98 catalogue drops to 320 (4,4%) and the average number of error per sentence drops to 0,39.

In the maintenance phase of the entrybase, the most needed entries might be added, but only after confirming that the entrybase does not already include an entry that could express the same thing. Webtran Checking Tool with it's checking rules, is used for this text control. The entrybase maintenance phase should mainly be used for adding new words like product types and brand names. This way the language remains controlled and the translation quality is preserved.

6 Supervised Machine Learning Methods

We have also developed human-assisted learning methods, which help the language specifier to create a controlled language definition. The learning methods are supervised, i.e. a human reviews the results of the methods before they are entered into the language definition.

Two types of methods are used. The first class includes ALE creation and usage expansion, and the second class extends the lexicon.

Three methods belong to the first class. The first one is the sentence alignment which is inspired by the one proposed by Gale and Church. Sentence alignment uses existing bilingual material to extract sentence correspondents. The extraction method uses sentence length and word properties to calculate matching probabilities for sentence pairs in the example material. A matrix is then created from these matching probabilities, and the optimal sentence correspondence is found out by using dynamic optimisation with the matrix.

The aligned sentence correspondents are typically very long and too specific to be used as such for CL definition. For this reason, the second method is used to split these aligned sentences into phrases according to the current entrybase. This split method finds suitable cut point words from existing language definition and splits the sentences from the cut points if the same amount of cut points are found in the sentences in both languages. These split entries replace the original, long sentence in the definition, thus allowing wider range of phrases to appear in the language.

The third method is the ALE generalisation, which extends the usage of ALEs. It generalises the repeating patterns in the language and diminishes the number of ALEs needed for language definition. The surface word forms that are generalised are concluded from the lexicon. The generalisation here means replacing surface word forms with their grammatical and semantic properties. The degree of generalisation, i.e. how many properties are included, depends on the number of words in lexicon that match the property set. This generalisation allows one general entry to represent multiple sentences in the language definition when the basic structure of the original sentences is the same.

Currently, the only method in the second class is the semantic classifier. It operates on new material with new words that should be added to the lexicon. The method helps the language specifier by finding all the new words in the text and then by making suggestions for the semantic properties of the words. These suggestions are determined by trying to fit all new non-translated text excerpts to existing rules. When a match is found, the features that enable the match are saved. This comparison is performed all text excerpt-general entry-pairs. After the comparison, the suggestions are evaluated in order to find the suggestions that are minimally ambiguous and appear repeatedly. These suggestions are then further evaluated and accepted by the language specifier.

The learning methods are presented in detail and with their impact on language definition in (Tenni 1999, Tenni et al. 1999).

7 Conclusions

Test experiences of Webtran have been positive in translating mail-order product articles. The first steps are now underway to embed the Webtran software into the catalogue production process of Ellos Corporation and to take it into everyday use. So far the language models have covered mainly women's clothes and their coverage will widen in the near future.

The CL modelling was started with an empty grammar in contrary to the adapting of a general-purpose machine translation system to a new domain. This was regarded to be a very important modelling choice as with a CL grammar it is necessary to exactly control the coverage of the grammar. Pruning and extending a pre-existing language model could not have achieved this.

Altogether, language modelling is a crucial bottleneck in adapting CL technology to new uses. The cost and difficulty of this task must not prevent the use of CLs. The role of the language-modelling tool has been growing all the time while we have been doing practical implementation of Webtran in production use. Now the tool has also been delivered to the piloting mail-order company and their translator team is starting to maintain their precious organisational knowledge, the language model, by themselves.

From research point of view, we are in the future going to investigate ways of developing our methodology for translating less controlled languages, as well. This involves ways of applying probability theories together with the machine learning to predict accuracy of translation. Moreover, we would need extensions to the formalism for handling the so called *distant dependencies* to solve bindings of faraway constituents. These we can avoid in the strictly controlled languages.

Acknowledgements

We would like to thank Eeva Palosuo and Maaret Virtanen from Tieto, Antti Mälkönen, René Holm and Irma Sinerkari from Ellos Oy for supporting our work. We are also grateful to Prof. Seppo Linnainmaa and Juha Sorva for proof-reading this paper. The work has been funded by Tekes Technology Development Centre in Finland.

References

Adriaens G. and Macken L. (1995). "Technological Evaluation of a Controlled Language Application: Precision, Recall and Convergence Tests for SECC". In proceedings of the Sixth International Conference on Theoretical and Methodological Issues in Machine Translation.

Almqvist I. and Sågvall-Hein A. (1996). "Defining ScaniaSwedish - A Controlled Language for Truck Maintenance". In Proceedings of the 1st Int. Workshop on Controlled Language Applications.

Arnola, H (1998). “Processing of Dependency-Based Grammars”. In Proceedings of the Workshop on Processing of Dependency-based Grammars, COLING-ACL'98, Montreal.

Carter C. and Hamilton H. (1998). "Efficient Attribute-Oriented Generalization for Knowledge Discovery from Large Databases". In IEEE Transactions on knowledge and data engineering, Vol. 10, No. 2, March/April 1998.

Chen S. F. (1993). "Aligning Sentences in Bilingual Corpora Using Lexical Information". In proceedings of the 31st Annual Meeting of the Association for Computational Linguistics, Columbus, Ohio, pp. 9-16.

Douglas S. and Hurs M. (1996). "Controlled Language Support for Perkins Approved Clear English (PACE)". In proceedings of the 1st Int. Workshop on Controlled Language Applications (CLAW'96), pp. 93-105.

van der Eijk P. (1998). "Controlled Languages and Technical Documentation". Report of Cap Gemini ATS, Utrecht.

Hutchins W. J. and Somers H. (1992). "An Introduction to Machine Translation". Academic Press.

Hyötyniemi Heikki & Lehtola Aarno (1989) “A Metatool for Implementing Task-Oriented Formalisms”. IEEE International Workshop on Tools for Artificial Intelligence. IEEE Computer Society Press, Los Alamitos, California, pp. 182-188.

Joscelyne A. (1998). "A controlling interest? Simplified languages to meet the global communication challenge". In Le Journal, 10 June, 1998, http://www.linglink.lu/LeJournal/

Jäppinen H., Lehtola A. and Valkonen K. (1986). “Functional Structures for Parsing Dependency Constraints”. COLING86, Bonn.

Jäppinen H., Lassila E. & Lehtola A. (1988). “Locally Governed Trees and Dependency Parsing”. COLING88, Budapest, pp. 275 - 277.

Kankaanpää T. (1999). "Design and Implementation of a Conceptual Network and Ontology Editor". VTT Information Technology, Research Report TTE1-4-99, June 1999, 74 p.

Kittredge R. (1987). "The Significance of Sublanguage for Automatic Translation". In S. Nirenburg, editor, Machine translation: Theoretical and methodological issues, Studies in Natural Language Processing, Cambridge University Press, pp. 59-67.

Lehtola A., Tenni J., Bounsaythip C. and Jaaranen K. (1999). "Controlled Languages as the Basis for Multilingual Catalogues on the WWW". In proceedings of EMMSEC'99, Stockholm, Sweden, 21-23 June, IO Press, pp. 207-215.

Nelimarkka E., Jäppinen H. & Lehtola A. (1984). “Two-way Finite Automata and Dependency Theory: A Parsing Method for Inflectional Free Word Order Languages”. Proceedings of COLING84/ACL, Stanford, P. 389-392.

Schwitter R. and Fucchs N. E. (1996). "Attempto Controlled English – A Seemingly Informal Bridgehead in Formal Territory". In proceedings of JICSLP'96, Bonn, Germany, September.

Shieber, S. (1986) An Introduction to Unification-based Approaches to Grammar. Stanford: CSLI Lecture Notes 4.

Sågvall-Hein A. (1997). "Language Control and Machine Translation". In proceedings of the 7th International Conference on Theoretical and Methodological Issues in Machine Translation (TMI-97), Santa Fe (USA).

Tenni J. (1999). "Methods and a Tool for controlled language definition". VTT Information Technology, Research Report TTE1-3-99, June 1999, 81 p.

Tenni J., Lehtola A., Bounsaythip C. and Jaaranen K. (1999). "Machine Learning of Language Translation Rules". In proceedings of IEEE SMC'99, Oct. 12-15, Tokyo, Japan (in print).

Valkonen Kari, Jäppinen Harri & Lehtola Aarno (1987). “Blackboard-based Dependency Parsing”. IJCAI'87, Proceedings of the Tenth International Joint Conference on Artificial Intelligence, Milan, pp. 700-702.

Whitelock P. and Kilby K. (1995). "Linguistic and Computational Techniques in Machine Translation System Design". UCL Press Limited, second edition.