HELSINKI UNIVERSITY OF TECHNOLOGY
Department of Computer Science and Engineering
Laboratory of Computer and Information Science

Aarno Lehtola

Grammar Formalism for Controlled Language Machine Translation: Augmented Lexical Entries

Licentiate's thesis, that has been given for review in Espoo, March 19, 2004.

Supervisor:	Prof. Timo Honkela
Instructor:	Prof. Seppo Linnainmaa

HELSINKI UNIVERSITY OF TECHNOLOGY Department of Computer Science and Engineering	ABSTRACT OF LICENTIATE’S THESIS
Author Aarno Lehtola	Date 19.3.2004 Pages 81
Title of thesis Grammar Formalism for Controlled Language Machine Translation: Augmented Lexical Entries	
Professorship Computer and Information Science	Professorship Code T-61
Supervisor Professor Timo Honkela	
Instructor Professor Seppo Linnainmaa, VTT Information Technology	

The thesis presents a formalism for specifying grammars for automatic controlled language translation. The described Augmented Lexical Entries (ALE) formalism was developed in the Webtran project that was funded by TEKES and carried out at VTT Information Technology in 1997-1999. One of the two major results of the project was the controlled language machine translation system Webtran, which is based on the presented ALE formalism.

Controlled languages are disambiguated sublanguages of human languages. They are characterised by specific use domain, selected vocabulary and simplified syntax. They have benefits such as accuracy and clarity of expression, which make them very usable in tasks where faultless and efficient communication are crucial, like in technical maintenance manuals, medical epicrises, weather reports etc. In this thesis, controlled languages have been used in commercial product descriptions in order to make them multilingually accessible by automatic translation with minimal or zero post editing. The approach is called “write-once-publish-many”.

The ALE formalism is declarative and intuitive so that a professional translator can use it. It has enough expressive power for the targeted commercial product descriptions. It has been found suitable for human assisted machine learning of translation grammars. Moreover, it has been tested and found suitable for translating in the directions Swedish→Finnish, Finnish→English, Finnish→French. Small experiments have also been carried out to translate into Estonian and Norwegian.

The Webtran system and the ALE formalism have been in production use at Ellos Postimyynti Oy since spring 2000, with an annual amount of around 2000 translated catalogue pages and 10000-15000 product descriptions. An independent survey by CSC Scientific Computing Ltd found that already after one year of use time savings of more than 30% had been achieved. Nowadays, the translators of Ellos maintain the ALE based grammars themselves.

Keywords: Machine Translation, Controlled Language, Grammar Formalism

TEKNILLINEN KORKEAKOULU
Tietotekniikan osasto

LISENSIAATINTUTKIMUKSEN
TIIVISTELMÄ

Tekijä Aarno Lehtola	Päiväys 19.3.2004 Sivumäärä 81
Työn nimi	Kieli Englanti

Grammar Formalism for Controlled Language Machine Translation: Augmented Lexical Entries
Kielioppiformalismi kontrolloidun kielen konekääntöä varten: Augmented Lexical Entries

Professuuri Informaatiotekniikka	Professuurin koodi T-61

Työn valvoja
Professori Timo Honkela

Työn ohjaaja
Professori Seppo Linnainmaa, VTT Tietotekniikka

Tämä opinnäyte käsittelee kielioppiformalismia, joka on kehitetty kontrolloitujen kielten automaattisen käännöksen määrittelyyn. Kuvattu Augmented Lexical Entries (ALE) -formalismi on kehitetty TEKESin rahoittamassa Webtran-projektissa, joka toteutettiin VTT Tietotekniikassa 1997-1999. Eräs projektin päätuloksista oli kontrolloidun kielen käännösjärjestelmä Webtran, joka perustuu käsiteltävään ALE-formalismiin.

Kontrolloidut kielet ovat ihmisten käyttämien kielten alikieliä, joista monitulkintaisuus on eliminoitu. Niille on ominaista tietty aihepiiri, valikoitu sanasto ja yksinkertaistettu rakenne. Kontrolloiduista kielistä saavutettavia etuja ovat ilmaisun täsmällisyys ja selkeys, jotka tekevät ne hyvin käyttökelpoisiksi tehtävissä, joissa virheetön ja tehokas kommunikaatio ovat keskeistä kuten teknisissä huoltokäsikirjoissa, potilaskertomuksissa, säätiedotuksissa jne. Tässä opinnäytteessä kontrolloituja kieliä sovelletaan kaupallisiin tuotekuvauksiin, jotta ne voitaisiin muokata automaattisen kielenkäännön avulla monikielisesti saataville suoraan tai minimaalisella jälkitoimitustyöllä. Lähestymistapaa kuvataan englanniksi termillä “write-once-publish-many”.

ALE-formalismi on deklaratiivinen ja intuitiivinen niin, että ammattikielenkääntäjä osaa sitä käyttää. Sen ilmaisuvoima on riittävä kohteena oleviin kaupallisiin tuotekuvauksiin. Se soveltuu käännöskielioppien ihmisavusteiseen koneoppimiseen. Lisäksi sen on testeissä todettu soveltuvan automaattiseen kielenkääntöön suunnissa ruotsi→suomi, suomi→englanti ja suomi→ranska. Pieniä kokeita on tehty myös kääntämiseksi eestiksi ja norjaksi.

Webtran-järjestelmä ja ALE-formalismi ovat olleet tuotantokäytössä Ellos Postimyynti Oy:ssä keväästä 2000 lähtien. Vuotuiset käännösmäärät ovat olleet noin 2000 tuotekuvaston sivua ja 10000-15000 tuotekuvausta. CSC Tieteellinen laskenta Oy:n tekemän puolueettoman selvityksen mukaan yli 30% aikasäästöt saavutettiin jo vuoden käytön jälkeen. Nykyisin Elloksen kielenkääntäjät ylläpitävät itse ALE-pohjaisia kielioppejaan.

Avainsanat: konekäännös, kontrolloitu kieli, kielioppiformalismi

Preface

The reported Augmented Lexical Entries formalism was developed in the Webtran project that was carried out at VTT Information Technology in 1997-1999. The project developed technologies for multilingual WWW content production. The major results were the controlled language machine translation system Webtran and the ontology based user interface for cross-lingual information retrieval CONE.

The author wishes to thank Professors Seppo Linnainmaa and Timo Honkela for making valuable comments concerning the work and encouraging it in many ways. Special acknowledgements are due to Jarno Tenni for being the main software architect of the Webtran machine translation system that is based on the presented formalism and that has been in production use at Ellos Postimyynti Oy since spring 2000. He has also given constructive comments about the earlier versions of this work. Tuula Käpylä and Lena Carlsson deserve warm thanks for contributing to the new Java based revision of the system, and Paula Silvonen as well for maintaining the version, which is in production use. The author is also grateful to Kristiina Jaaranen, who was the first linguist to use the formalism and to evaluate its usability for specifying grammars for automatic translation of product catalogues from Swedish to Finnish. Although Dr Catherine Bounsaythip and Kuldar Taveter were concentrating on the ontology based cross-lingual information retrieval interface of the project, their co-operation was very important in bringing conceptual modelling related ideas to the formalism. The idea was born to use domain ontologies, such as product models, to constrain the language. This idea was further elaborated in the follow-up Mkbeem project. However, this thesis is written in retrospect and is limited to the formalism developed in the Webtran project.

Earlier co-operation with Dr Harri Jäppinen/Arnola, Dr Esa Nelimarkka, Dr Heikki Hyötyniemi, Kimmo Kettunen, Eero Lassila, Matti Yli-Lammi and Juha Niemistö at Sitra Foundation provided important knowledge and experience of human language processing formalisms. Several in-depth discussions with colleagues Dr Alain Leger, Dr Johannes Heineicke and Alain Cozannet from France Telecom R&D have brought insight to the relationship of human language and ontologies. The possibility to be a visiting researcher in Eurotra-DK in autumn 1987 brought the author important insight to transfer based machine translation system formalisation. Prof. Bente Maegaard, Dr Annelise Bech, Dr Hanne Ruus and Dr Poul Andersen are well acknowledged for making the visit possible and the many fruitful discussions.

Tekes, Ellos Postimyynti Oy, and TietoEnator Oyj funded the Webtran project. The author wishes to thank the members of the management group, Matti Sihto from Tekes, Antti Mälkönen, René Holm, and Kristiina Mokkila from Ellos, Eeva Palosuo and Maaret Virtanen from TietoEnator, for encouraging the project team to narrow the work focus to controlled language processing instead of unlimited language and thus to achieve practically functional and innovative solutions. The author is also very grateful to Johan Grandell, Kirsi Villo-Moen and Irma Sinerkari from Ellos for the successful implementation of the Webtran machine translation system into production use. Support received from VTT's group manager Markus Tallgren and Lauri Seitsonen is also well acknowledged.

Last but not least the author is grateful to his family members Sisko, Lauri and Tuomas for their support and patience during the long-lasting work process.

In Rajamäki, March 19, 2004

Table of Contents

1. Introduction

In the year 2000 native English speakers became outnumbered by native users of other languages in the Internet population. Since then the trend has continued and in September 2003 the others comprised already 64,4 % of the population [WWW-GlobalReach 2004]. The linguistic diversity is huge among the population. Even within Europe there can easily be counted over 60 languages among which even the smaller ones, like the almost 200000 Icelandic speaking Internet users, comprise potential customers groups for international eShops. There is a remarkable need for cost effective IT solutions for enabling multilinguality in consumer Internet trading.

Machine translation (MT) technology has been quickly finding new uses in WWW information services. MT has already been successfully integrated both in information retrieval from multilingual text material, and in translating documents into the language of the user. For instance, [Kikui & al. 1996] describes a cross-lingual URL search system (called TITAN), which originally enabled the user to query in Japanese or in English. Results were returned in their original languages with headers translated into the user language. Titan has been later on adapted to cover also Korean and Chinese. AltaVista search service has been already for several years embedding SYSTRAN MT system [WWW-AltaVista 2004, WWW-Systran 2004]. AltaVista provides automatic translations of the found web pages from English to 8 languages and vice-versa. Nowadays there are several other services available for WWW page translation. One may test their quality through the test page of Humanitas International [WWW-Humanitas 2004]. Only one of the services, namely InterTran, claims to support Finnish. Experimentation shows its Finnish support quite modest. Moreover, there are several inexpensive desktop translation programs that can be used to translate WWW contents. The cheapest ones are no more than electronic dictionaries and consequently their translations are just rather poor word-by-word or phrase-by-phrase replacements. Most of translations done by the above mentioned general-purpose systems are only draft-quality. Systems that have been specifically developed for particular target domains have been more successful.

1.1 Webtran project

The Webtran project of VTT started in 1997 with its goal to develop methods and software to support multilinguality in WWW services [Lehtola & al. 1998b, 1999a]. Already in the beginning, a few very important requirements were recognised in co-operation with the user partners Ellos Postimyynti Oy and TietoEnator Oyj. The developed technology must be:

1. embeddable, reliable and efficient,

2. accurate,
3. transportable,
4. easily maintainable, and
5. cost-effective.

The technology must be possible to embed into online services, as well as, into backoffice service maintenance processes. Thus, the application programming interfaces need to be lucid. Moreover, the technology must be reliable and provide good performance times. The response times need to be quick enough for online use. Accuracy is crucial because the technology is meant to be used for commercial activities where imprecise or faulty information would lead to dissatisfied clients or in worst cases to claims against the company. The solutions must be transportable to different operating environments. The technology needs to be portable so that it can be transported to different domains of use. The software and the knowledge resources must be easily maintainable. As for the linguistic knowledge bases, the solutions must be such that the current company personnel can take care of their maintenance. Cost-efficiency was very important because the marginal cost of providing multilinguality must not exceed the benefits gained through larger clientele.

Translation of product information and catalogues was considered as one of the most expensive preparatory internationalisation activities of eShops. Thus, the first of the objectives of the Webtran project became development of machine translation that would be specialised in product descriptions and would either require minimal post-editing or produce reliable translations fully automatically even online [Lehtola & al. 1998a, 1999b]. The second objective of Webtran project became to develop cross-lingual information retrieval for restricted domains [Bounsaythip & al. 1998, Taveter & al. 1999]. The machine translation grammar formalism described in this thesis is one of the results of the Webtran project. The described technology has been in production use at Ellos since spring 2000 and the thesis has been written in retrospect.

1.2 Suitability of general-purpose MT

General-purpose machine translation systems always need human resources to produce reliable high-quality output. Bad quality of results is a common problem when such systems are used in special domains without adapting them. Deepening domain knowledge is necessary for obtaining accurate translations. General-purpose machine translation systems may use too large language specifications and vocabularies, which cause difficulties through conflicting rules. The systems recognise and try to solve many unnecessary ambiguous choices. Besides, general-purpose systems likely lack specifications needed to translate the special terms and idioms of the target domain. Moreover, the general-purpose systems often are rather syntax limited and do not execute semantic feasibility checking, which in many use cases would be required.

(a)	Source:	**Jouluinen mekko, rypytetty hameosa, röyhelöhelma ja sivuilla solmittavat nauhat.**
	SYNTAX:	Christmas smock, the gathered skirt part, frill hem and on the sides tie laces. (program asked to disambiguate word **sivu** (*page* or *side*)
	TranSmart:	*Of Christmas dress, rypyttää skirt part, frill hem and tapes which are tied on the pages.*
	Actual meaning:	Christmas style dress, with ceased skirt part, frill hem and knottable laces on the sides.
(b)	Source:	**Päällä kaksi säädettävää solkea.**
	SYNTAX:	On two adjustable solkea.
	TranSmart:	*Buckle adjustable with head two*
	Actual meaning:	Two adjustable buckles on top.
(c)	Source:	**Suorat lahkeet, joiden suissa väljä joustin.**
	SYNTAX:	Direct you peak, whose mouths loose spring.
	TranSmart:	*Straight legs in the mouths of which I yielded large.*
	Actual meaning:	Straight pant legs, whose cuffs have elasticity.
(d)	Source:	**Kaikki nahan ja synteettimateriaalien hoitoon tarvittava kätevästi yhdessä.**
	SYNTAX:	*All leather and the synteettimateriaalien to care necessary handily together.*
	TranSmart:	All skin to be needed synteettimateriaalien for care and handily together.
	Actual meaning:	Everything necessary handily together for taking care of leather and synthetic materials.
(e)	Source:	**Kauniisti viimeistelty turkki.**
	SYNTAX:	Nicely finished Turkey.
	TranSmart:	*Turkish that has been beautifully finished.*
	Actual meaning:	Nicely finished fur coat.

Table 1.1: Examples of translations of mail-order catalogue texts by two commercial general-purpose machine translation systems.

Tests were done in 1998 to evaluate how general purpose machine translation systems perform with the language of women's clothes and accessories product descriptions of Ellos. The tested translation direction was from Finnish to English as there are no general-purpose systems for the direction from Swedish to Finnish. The tests revealed how the wide scope of the used systems complicates the translation process and makes it error prone. They also demonstrate ambiguities on the level of word forms, word meanings, sentence structures and translation correspondences. The general-purpose systems used were SYNTAX of Blue Ball Ltd., nowadays Hot Ball Ltd. [WWW-HotBall 2004], and TranSmart of Kielikone Ltd. [WWW-Kielikone 2004].

In Table 1.1 are shown some examples of the translation tests. Underlines have been manually added afterwards to pinpoint some of the problems. In example (a) the clothing terms cause difficulties. For instance, in Finnish **nauha** can mean either *lace* or *tape,* but in this context only the first one is semantically correct. Moreover, the proper translation of **sivuilla** requires semantic context information. In example (b) the second system misinterprets **päällä** as *with head* although the original sentence does not mean instrument but place. In example (c) both systems make serious translation errors, first one in the beginning of the sentence and the second one in the end. In both cases it is hard to observe what has went wrong in their analysis. Example (d) illustrates word-ordering difficulties, as **kaikki** (all or everything depending on context) does not refer to *leather*, but to the more remote word *necessary*. Both systems made quite disorderly translations. In example (e) the difficulty lies in word **turkki**, which may mean Turkish language, country of Turkey, fur or fur coat. Both systems choose automatically geographic interpretations leading into humoristic translations.

The sample translations also need to *"understand"* what is translated, especially when there is no human post editing embedded in the processing. General-purpose machine translation systems are rather syntax oriented. In the case of the sample texts much better accuracy can be reached by controlling the language and doing semantics based machine translation. At the time of the tests SYNTAX had a lexicon of about 80000 words and TranSmart around 65000 words. Nowadays both have much larger lexica, but one can doubt whether that would add accuracy. Likely the challenges for solving homonyms have increased as well.

1.3 Controlled languages

Controlled languages are disambiguated human languages characterised by specific use domain, selected vocabulary and simplified syntax. Controlled languages are evolvable. Controlled languages have benefits such as accuracy and clarity of expression, which make them very usable in tasks where faultless and efficient communication are crucial, like in technical maintenance manuals or in medical epicrises. Control is used to eliminate ambiguities and difficult syntactic and semantic complexities such as anaphora. Such restrictions make these languages suitable to automatic language processing, like machine translation [Huijsen 1998, Joscelyne 1998, Knops & Depoortere 1998]. Even reliable fully automatic machine translation (FAMT) can be reached. Information resources produced in controlled languages are applicable for automatic interpretation, e.g., in context of data mining and knowledge discovery. Moreover, mono or cross-lingual information retrieval can greatly benefit from such information resources.

Sublanguage is a nearby concept that has been used in this thesis synonymously to controlled language. Lehrberger [Lehrberger 1982] characterises sublanguage by the following properties:

1. limited subject matter,
2. lexical, semantic, and syntactic restrictions,
3. deviant rules of grammar,
4. high frequency of certain constructions,
5. text structure, and
6. use of special symbols.

In-company language is yet another nearby concept. In this thesis it can be considered to be a controlled language that is restricted to be used in a particular company. The company maintains it and it belongs to the company's intellectual properties and culture. In electronic commerce strong outlets such as Ellos and Ikea can establish such in-company languages as part of their brand building. During our co-operation with Ellos it has become clear that the language used in their product catalogues is one specific in-company language. Moreover, it has two dialects as there are two brands that are targeted to different client groups with their own catalogues, namely Ellos for mature and Josefssons for younger clients.

Controlled language approach has been successfully used to enhance the quality of translation [Kittredge 1987], and also readability, understandability, and maintainability of the original texts. The idea is old and one of the earliest examples is from the 1930's, when Odgen developed "Basic English" that contains 850 words and a few inflection and derivation rules [Odgen 1932]. Controlled languages got into industrial use as computer based documentation authoring started. Examples include TITUS system, which was designed for storing and translating abstracts on textiles, using many restrictions on vocabulary and syntactic structures [Hutchins & Somers 1992]. Also, the success of the machine translation system TAUM-MÉTÉO is explained by the restricted vocabulary and telegraphic style syntax used in the translated weather forecast bulletins [Hutchins 1986, Colmerauer 1992, Whitelock & Kilby 1995]. In Finland a system was developed for Finnish to Swedish translation of weather forecasts [Blåberg 1989]. The European Association of Aerospace Industries (AECMA) maintains the standardised AECMA Simplified English for authoring aerospace manuals. Also several other organisations use simplified language [Adriaens & Macken 1995, Douglas & Hurs 1996, Schwitter & Fuchs 1996]. Caterpillar Inc. has its "Caterpillar Fundamental English" [Caterpillar 1974] and General Motors Inc. its "Controlled Automotive Service Language" [Means & Godden 1996] for authoring maintenance manuals in a controlled way. Uppsala University has been developing together with Scania ScaniaSwedish [WWW-Scania 2004] for the preparation of truck maintenance manuals in a controlled Swedish [Almqvist & Sågvall-Hein 1996, Sågvall-Hein 1997]. Another non-English based controlled language is the

Siemens "Dokumentationdeutsch" in Germany [Schachtl 1996]. In addition to the mentioned ones, several other examples of controlled languages exist.

1.4 Goal and overview of the thesis

The goal of this thesis was to develop a knowledge representation formalism that can be used to define controlled language specifications. The formalism was designed to be used in the controlled language machine translation system that was under development in VTT's Webtran project. Related academic works are Jarno Tenni's M.Sc. thesis [Tenni 1999], which handles machine learning methods for learning grammars in the formalism, and Kristiina Jaaranen's M.Sc. thesis [Jaaranen 2000a], which handles the modelling of a grammar for translating product description articles of Ellos' mail-order catalogues from Swedish to Finnish using the formalism. Tenni also was the main software architect of the grammar modelling workbench and the machine translation system.

Chapter 2 of this thesis contains a survey of grammar formalisms, which have been used in machine translation systems. The section also presents taxonomy of properties, which can be used to evaluate these formalisms. Chapter 3 describes the developed formalism and presents examples about how it can be used. The thesis assumes the general architectural idea of separating the knowledge representations from the algorithms that are used to apply them for the processing tasks. The presented formalism is declarative and can be implemented in several ways. The practical importance of this separation is based on the fact that the implementation can vary depending on the requirements and available resources. The software is outlined that was implemented for Ellos' catalogue translation purposes. It involves hierarchic compilation of the grammars in order to gain runtime efficiency. A reference algorithm is described for language translation based on knowledge specifications represented in the formalism. The formalism is compared to the solutions studied in Chapter 2. There are also presented experiences from supervised machine learning of language specifications in the formalism and experiences from using the formalism and the system in production use. Chapter 4 concludes the thesis by outlining future research goals for improving the formalism and the machine translation methodology. Appendix A explains terms used in this thesis, Appendix B provides annotated bibliography of the Webtran project, and Appendix C contains example translations done by the Webtran system.

1.5 Syntax notation

In this thesis syntax of various formalisms is described using BNF notation with the following conventions. The rewrite operator (::=) separates a nonterminal from the right hand side of a rule, which describes its structure. Alternative constructs on the right hand side of the production rules are separated by bars (|). Nonterminals are in italics. A nonterminal followed by an asterisk (*) can have zero to more occurrences and one having two consecutive dots (..) right after can have one to more instances. Terminals are in boldface. The symbol ε means empty. To avoid unnecessarily detailed BNF some leaf level nonterminals have been given descriptive names (e.g. *hierarchical_name_with_dots_between_parts*) or they have been described verbally in the text.

2. Review of some existing MT formalisms

2.1 A brief history of MT

The idea of automatic language translation is indeed not a new one. The use of mechanical dictionaries to overcome the barriers of language was first suggested in the 17th century. In the 1930's two patents on mechanical translation were applied independently. George Artsrouni demonstrated in France in 1937 a paper tape based device that found the equivalents of words in another language. Russian Petr Smirnov-Troyanskii outlined in his patent a three phase translation process involving basicly what nowadays is called analysis, transfer and synthesis. The first and the last phase were to be done manually by humans and the transfer was to be carried out by the machine. John Hutchins has written excellent presentations of the history of machine translation [Hutchins 1986, Hutchins 1995].

The emergence of "electronic brains" and the article of Dennis Weaver [Weaver 1955] made machine translation a central target for computer scientists. Large-scale research of the topic gradually started. Yehoshua Bar-Hillel made a comprehensive survey of machine translation current state in 1960 [Bar-Hillel 1960]. At that time there were about 220 researchers (full-time equivalent) in the world developing automatic language translation in at least 22 organisations. In the spirit of the cold war the research concentrated around translation of English and Russian. Bar-Hillel presented also some sceptic views about the feasibility of the applied approaches and examples of fundamental theoretical difficulties hindering a quick breakthrough in the technology. He presented the famous "the box was in the pen" example, which necessitates deeper semantic analysis to solve that the pen means here a playing ring of a child, i.e. playpen.

Nowadays new software products for automatic translation or for supporting interactive translation work are announced frequently. A listing of such products is given in the *Compendium of translation software* compiled and maintained by John Hutchins and published by the European Association for Machine Translation [WWW-EAMT 2004]. It gives more than 140 companies, with brief descriptions of the products, the language-pairs supported, dictionary sizes when known, etc.

2.2 Basic models of machine translation

Machine translation systems have been developed using different models. Main models used are *direct*, *transfer-based*, *interlingua-based* and *corpus-based* (see e.g. [Nirenburg 1987]).

Direct-translation systems contain tailored approaches for each supported language pair. The earliest systems were of this type and the approaches were rather straightforward with lexical replacements of source words with target language correspondents and special procedures for revising the word orders to comply the grammar of the target language etc.

Transfer-based systems use an intermediate phase for translating source language specific structural representation (outputs of analysis process) of original texts into the target language specific structural representations, which are used as input of a generation/synthesis process. This approach is used for languages that are relatively close to each other, like European languages. The benefit is that in a multilingual cross-translation system the analysis and generation parts need to be designed only once and the major work goes for the language pair-wise transfer rules. Usually the analysis does not go deeply into the semantics of the texts, but the transfer rather operates on syntax level. The Eurotra project, which developed cross-translation system between nine official EU languages, relied on transfer approach.

Interlingua approach analyses the source text into some interlingual semantic presentation that is used as input of the generation process. This approach has been found useful for translating between languages that are not near relatives, like between Indo-European and Asian languages. The target sentence structure may differ considerably from the origin. As an example, the Japanese Mu system assigns source sentence entities with deep cases such as *subject, object, recipient, origin, partner, opponent, time, duration, space, space_trom, space_to, space_through, source, goal, attribute, cause, tool, material, component, manner, condition, purpose, role, content, range, topic, viewpoint, comparison, frequency, circumstance, quantity, choice, property, whole, part, owner, author, trend, means* etc.

Main corpus-based approaches include both *translation memories* (TM) reusing possible translations stored in the system memory, and *example-based machine translation* (EBMT) techniques, using automatically discovered translation templates, including generalised structure representations of phrases with feature conditions.

Figure 2.1 illustrates the differences of direct, transfer and interlingua approaches. It also includes the target positioning of the Webtran system. While Webtran was anticipated for the preparation of multilingual sets of texts from source language pivot texts, and the target languages are European, it was natural to consider a transfer-based approach. However, as the controlled language has a close relationship to the domain ontology, there is also semantics involved in the approach. To help cope with the knowledge acquisition bottleneck and to take benefit of the large database of sample translations, the developed hybrid approach includes also features from example-based machine translation.

With respect to the required user interaction, MT systems can be classified into fully automatic MT (FAMT), human aided MT (HAMT) and machine aided human translation (MAHT or MAT). Interactive systems have been developed since 1970's (e.g. [Kay 1980, Melby & al. 1980]).

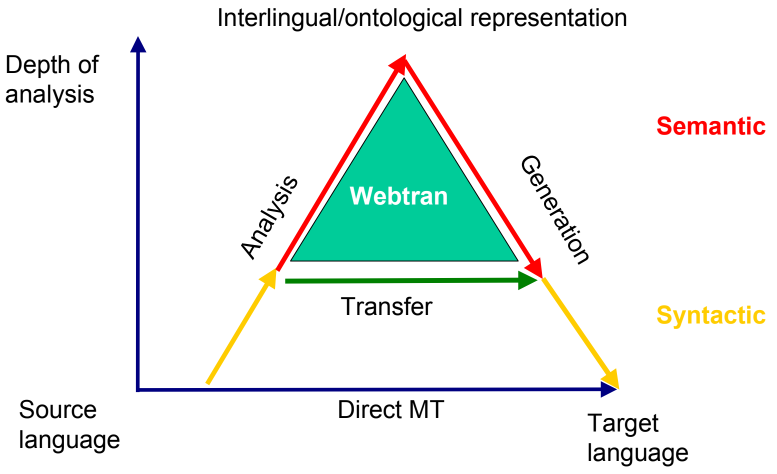


Figure 2.1: The machine translation triangle with the target positioning of Webtran.

2.3 Background to linguistic formalisms

There are several formalisms developed for the specification of linguistic knowledge. Usually the formalisms are task specific. One way to classify the formalisms is by using a stratificational language processing model. This way formalisms can be divided, e.g., into categories like phonology level, morphology level, lexical level, (sentence) syntax level, and semantic level. However, there is no general consensus or exact definition on what those levels are, what are the constructs processed, and what are the knowledge description formalisms. There are different schools among researchers and developers. Depending on the actual goals of the systems the decomposition models may be quite different, like the models in Hearsay project [Erman & al. 1980] and in Kielikone project [Jäppinen & al 1988b, Lehtola & al. 1988a, 1988b].

In the area of word-form morphology several finite-state approaches have been proposed and implemented, like [Jäppinen & al. 1983, Jäppinen & Ylilammi 1986, Blåberg 1994]. The Two-Level Model (TWOL) of morphology [Koskenniemi 1983] has been widely accepted as a standard and word formations of several languages have already been specified according to it.

Syntax parsing is an area where several linguistic and computational theories have been proposed. These include transformational grammars [Chomsky 1957, 1965], transition network grammars (outlined e.g. in [Winograd 1982], ATN in [Woods 1970]), and unification grammars and other constraint-based grammars [Schieber 1986, 1992]. Usually the grammars specify derivations of acceptable language constructs. Sometimes it is advantageous to make the grammars over-generating and in some approaches there have been added way to specify non-acceptable constructs, like the killer-rules used in the Eurotra project. The algorithmic issues in implementing the accompanied parsing automata and constraint satisfaction facilities have been widely studied.

If we consider machine translation systems, there are many formalisms proposed for representing transfer between structures on different levels. On the whole, the linguistic community has been active in the past fifty years in inventing language formalisms that also have computational dimensions (see [WWW-ACL 2004, WWW-LingDataCons 2004]). Usually the theories have been targeted to some specific needs and there has not yet appeared any absolute philosopher's stone. Thus for Webtran MT system it was reasonable to consider the requirements for its language formulations, to look among the available approaches for ideas and to develop a formalism specifically suitable for it.

2.4 Properties of formalisms in MT systems

A language specification formalism can have the following properties:

1. Language independence
2. Linguistic felicity
3. Nondirectionality / Invertibility
4. Expressiveness
5. Declarativity
6. Uniformity
7. Non-stratification
8. Monotonicity
9. Reusability
10. Multidimensional patterns
11. Complex feature structures
12. Error tolerance

A formalism may have the following implementation related properties:

13. Implementability
14. Computational complexity
15. Deterministicity
16. Portability

Language independence: This means that the formalism can be used for specifying any language without modifying it. In case of Webtran it was necessary to cover at least languages officially used in the European Community countries and languages among their native minorities. Languages imported by immigrants, e.g. Asian languages, were not in the focus.

Linguistic felicity: The degree to which descriptions of linguistic phenomena can be stated directly as the modeller would wish to state them [Shieber 1986].

Nondirectionality / Invertibility: This means that specifications used for analysis should apply also for generation, or that specifications used for transfer from language A to B should apply also for transfer from B to A. For instance the widely used two-level model of morphology complies with this requirement [Koskenniemi 1983].

Expressiveness: Expressiveness or expressive power means the capability of a grammar formalism to cover wide scale of language issues. As for a human language syntax formalism this includes not only the elementary requirement of supporting syntactic specification of concatenations of language constructs, but also specifying of the following (for a comprehensive survey of the linguistic phenomena, cf. [Flickinger & al. 1987]):

- **Agreement:** Agreement relationships are required between words of sentences. For instance, in Finnish in a noun phrase the adjective must have the same case and number as the noun. In French the gender of the article and the noun need to agree. The survey of Flickinger & al. lists for English six types of agreement.
- **Co-ordination:** Co-ordination refers to handling the interdependencies of the conjunctive sentence parts with each other and with the rest of the sentence. For instance, in the sentence "Markus manages and teaches programmers" the two verbs form a conjunctive subpart with Markus as the subject and programmers as the objective. Both of the verbs need to agree in number and person with the joint subject. The survey Flickinger & al. lists for English 11 types of co-ordination.
- **Ellipsis:** Ellipsis refers to omitting to repeat a sentence part to gain more compact expression. For humans it is easy to understand such sentences, but for computers this phenomena needs to be explicitly specified. Ellipsis may also involve ambiguity that needs to be solved based on context knowledge. For instance, in the sentence "Old men and women were dancing" it is not clear if also the women are old. In translation it is often possible to retain and transfer the ellipsis to the target language.
- **Anaphora:** Pronominal anaphora refers to handling pronominal references in both intra and extra sentential scope. As an example, in the sentence "The manager evaluated her staff" the pronoun "her" may refer to the "manager" or to somebody mentioned in the preceding context.
- **Long-distance dependencies:** Long distance dependencies are effective between sentence parts that are neither consecutive nor in the same subtree of the syntax parse tree. In other words, some elements of the structure is

constrained by features of another part of the structure that is not its parent or one of its children, but appears at a distant place in the tree. The short example “Which baby did the girl want to kiss?” contains a long-distance dependency between the verb “kiss” and its objective “baby”. Augment Transition Network (ATN) grammar uses HOLD register and Lexical Functional Grammar (LFG) special variables to specify long distance dependencies [Winograd 1983].

- **Negative evidence:** Negative evidence refers to the capability of specifying also non-constructs of the specified language. This capability allows deliberate use of over-generative grammars together with negative rules to prune illegal instances. The motivation is to improve the cognitive economy of the knowledge base. This technique has been used, e.g., in Eurotra projects language formalisms, which use killer rules for pruning overly generated constructs [Arnold & al. 1986, Arnold & des Tombe 1987, Varile & Lau 1988].

Declarativity: This means that specifications do not involve algorithms or other information that effects directly in their computational use. Algorithms and specifications are separated. Unification based and constraint based grammar formalisms usually are declarative [Shieber 1986, Shieber 1992]. Augment Transition Network (ATN) grammar is an excellent example of a non-declarative formalism while it factually describes the parsing process control as an automaton [Woods 1970, Winograd 1982]. Some formalisms are basically declarative, but provide for expert users the option of calling procedures from the grammars thus weakening their declarativity. E.g. *Definite Clause Grammar* belongs to this category, as it lets the modeller include side-effective Prolog procedure calls to the syntax rules [Pereira & Warren 1980]. As grammars grow larger, it is very beneficial, if they are declarative. If the grammars embed a lot control knowledge, they may become intractable as the complexity and the size of the language models grows. Often declarativity also makes possible optimisations in algorithmic implementations and enables the use of the same knowledge description for more than one task.

Uniformity: Uniformity means the ability of a formalism to provide high expressive power without resorting to tailored ad hoc means to handle the most difficult phenomena. Both the simple phenomena and the most complicated phenomena in a grammar can be expressed with the same uniform expression syntax. Unification grammars, like the Lexical Unification Grammar (LFG, [Winograd 1983]), provide a good example of uniformity.

Non-stratification: Stratification refers to the separate modelling and processing of different levels of language, e.g. word formation, lexical knowledge, syntax, logical analysis (e.g. anaphora, linguistic quantifiers), semantics with respect to ontologies etc. In the classical approach each level gets its own knowledge description in the most suited formalism [Winograd 1983]. The language analysis model of the Kielikone project is an example of stratificational approach

[Jäppinen & al. 1988b]. This model includes a task-oriented specification formalism for each level [Lehtola & al. 1988a, Lehtola & al. 1988b]. However, stratification may complicate the language modelling, as the different levels of knowledge specifications need to be compatible so that the overall pipeline would work properly. Thus, adding or revising a definition in one level often leads to updating other levels, as well. Non-stratification simplifies model acquisition and maintenance. A non-stratified formalism lets specify phenomena on different levels of language in the same knowledge specification.

Monotonicity: If adding a new rule to a language description leads to some other part of the description to become void, we are dealing with a non-monotonic knowledge description. A formalism that embeds control knowledge or side-effective operations is more susceptible to non-monotonicity and the knowledge descriptions are harder to maintain.

Reusability: Knowledge descriptions are reusable, if they can be ported to other formalisms and uses easily retaining their details. It has been observed that resources, which are encoded in a very rich formalism, are hardly reusable in lean formalisms. This suggests avoiding rich expressiveness in order to gain reusability.

Multidimensional patterns: Multidimensional patterns allow expressing constraints in several dimensions. Lexical, morphological, syntactical, functional and semantic properties of constructs can be referred in same condition patterns.

Complex feature structures: Complex feature structures refer to the possibility that feature structures have internal structure. Such structures can be very effective in expressing locally ambiguities, e.g., with mutually disjunctive substructures. For instance, the dependency grammar specification formalism of the Kielikone project supported complex feature structures that could be used to present compactly a set of ambiguous choices [Nelimarkka & al. 1984, Lehtola & al. 1985, Valkonen & al. 1987].

Error tolerance: One distinctive property in human language processing systems is their tolerance to ill-formed input. In a dialog with a computer it is most annoying if the system rejects user inputs due to minor grammatical flaws. Most frustrating for a user is rejection of correct input due to deficiencies in the language models of the system. Engineering tolerance to the language specifications requires from the used formalism certain means. For instance, the use of over-generative specifications together with negative evidence is one way to engineer error tolerance.

Implementability: The richer a formalism is the more difficult it is usually to implement. Missing or partial implementations are not unusual, especially for linguistics motivated works. The users may be forced to rely on the future to bring efficient and full implementations for them. There is also linguistic research that uses strict formalisation for mediating knowledge to the research community, but practical implementation is not a goal or even possible, like [Jämsä 1986].

Computational complexity and deterministicity: Theoretically Koskenniemi's two-level-morphology model has been proved to be NP-complete [Barton & al. 1987] meaning, that the worst case time or space requirement is not polynomially limited in the size of the input, but grows steeply. In practice this does not prevent the use of the two-level-model in real-world applications. In fact, it is the most widely used morphology solution nowadays. The way of handling ambiguities has large effects to computational performance and makes a central difference in implementation approaches. For instance, in syntax analysis there have been proponents of deterministic parsing and those who consider that all choices are important to check in order to reveal true ambiguities. The deterministic parsing advocates have achieved good performance while the other group has needed to solve difficult complexity problems. One can claim that the deterministicity assumption is infeasible as human language is fundamentally ambiguous and making deterministically right choices is therefore not possible. Limiting the length of one time input of processing is one way to cope with exponential growth of execution time. Another very pragmatic way for handling worst cases is to use time cutters in the processing algorithm after having found some solution/s.

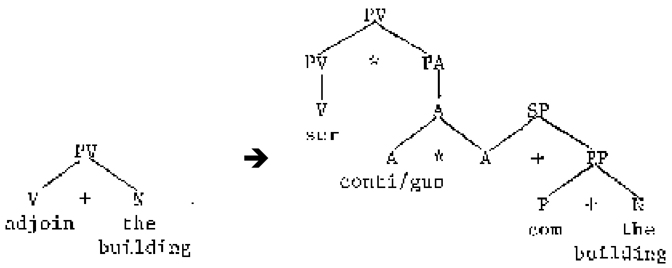
Portability: Portability simply means that the implementation can be easily ported to wide scale of operating platforms. Big computing resource requirements are in some cases one practical obstacle for portability.

The following sections outline specification formalisms that have been used in major machine translation projects or systems. Formalisms of Interactive Translation System, ALP Systems Ltd, TAUM system, Eurotra project and Metal system will be studied in more detail. Their methodology is quite sophisticated and relatively well documented in several publications. The well-known system Systran is omitted from the review. Being developed since the late 1960’s it lacks several features of modern MT. In Systran linguistic data and algorihms are not strictly separated, operations performed by different modules are rarely discrete, and the linguistic strategy is relatively primitive [Whitelock & Kilby 1995]. One difficulty in making this kind of survey is that not enough technical detail is available in each case from public resources. Machine translation companies rely on industry secrets, when they are protecting their intellectual property rights.

2.5 Interactive Translation System and ALP Systems

Brigham Young University in Provo, Utah, was a place for active machine translation research in the 1970’s. The original goal was to develop an Interactive Translation System (ITS) to translate from English initially to French, Spanish, Portuguese, German and Chinese [Hutchins 1986]. As often in good projects a number of good approaches were developed. Eventually the development continued in two competing companies, Weidner Communications Corporation (1977-) and ALP Systems (1980-). Both became famous machine translation companies in their time. However, only the latter company made public enough its specification formalism so that it can be reviewed.

ITS was transfer-based one-to-many translation system, with online user interaction during the analysis and transfer for reducing ambiguity. For presenting the transfer rules, a modification of transformational grammar, called junction grammar, was used [Melby & al. 1980]. Figure 2.2 illustrates how a transformational transfer rule looked like. The formalism describes procedurally how so-called J-trees, resembling phrase structure trees, are transformed. Distinctive feature of junction grammar is the treatment of relative clauses by distinguishing ‘subjunction’ (*“the fact that John came …”*) and ’interjunction’ (*“the book that John read …”*), and the coding of ambiguities in the J-trees. In practical use ITS was soon found to be too elaborate. Knowledge of junction grammar was a prerequisite for users of ITS. In extensive testing translation of one page around 250 words) of unrestricted texts required user interaction around 30 minutes per target language, hardly less than draft quality manual translation would have taken. MT was considered to be attractive if the average per page would have been under 10 minutes. Alan Melby developed during the early 1980’s a new more efficient version of ITS in the form of a translator’s workstation [Melby 1983].



```
ADJOIN
1 &s map
2 ser conti/guo com [l.v
3 unir .v
4 &w("1") c =pa;ser;conti/guo;com
  =PA
1 &s p verb;pa;prep
2 /* verb becomes copula with pred adj
     modified by PP */
3 &s *,f (,ad) verb;pa
4 &s m (,ad)
5 &s f,do (do
6 &t *,t (ad,,,do) ;prep
```

```
EXPLANATION:
1-3: Interact on the two translation choices
given.
4: When "1" is returned from the MAP action
code ("ser conti/guo com" was chosen), call
dictionary subroutine = PA, passing the 3
parameters shown.
(=PA)
1: Associate the 3 names given with the
parameters passed.
2: Comment
3: Build a predicate adjective, setting the
verb and predicate adjective to the values
("ser" and "conti/guo") of the respective
parameters.
4: Move any insertions on the verb to the
predicate adjective.
5: Find the direct object of the verb (if
any).
6. If direct object is found then build a
prepositional phrase modifying the predicate
```

Figure 2.2: An example transfer rule in the junction grammar formalism.

ALP Systems developed a new formalism called PeriPhrase to replace the above described junction grammar notation [Beesley & Hefner 1986]. This specification language was much higher-level than its predecessor was. The idea was to use the same formalism to specify both the syntax analysis rules and the transfer rules. The general rule form is simply **pattern => rewrite.** The pattern may refer to phrase categories or partial phrase structure trees and contain feature constraints and repetition constrains. The rules can be context-sensitive. Below are a few examples:

(a) DET ADJ N => NP[1, 2, 3].

(b) DET(number#plural) *(ADJ 0-1COMMA) N(number=singular, Y:=case) => NP[…](number:=singular, case:=Y).

(c) DET N_V => 1 2:=N.

(d) NP[DET, *ADJ, N] => 1[2, 4, 3]

(e) DET *AJD N; check_flag(X) => NP[…]; print_message, set_flag(Y, Z).

(f) DET ADJ N N; action(X) =>
choose(X) { NP[1, N[2, N[3, 4]]] | NP[1, N[N[2, 3], 4]] }.

Rule (a) simply conflates all mentioned pattern components under a new father node, thus forming a noun phrase structure. On the rewrite side numbers are used to refer to the categories in the pattern. DET means determiner, ADJ adjective, N noun and NP noun phrase. Rule (b) is a more complicated variant of a noun phrase. Number of the determiner is constrained to not equal to plural. Repetition can be expressed by prefixes, which include Kleene star for zero or more repeat items, 1+ for one or more items, and 0-1 for optionality. Rule (b) expects zero or more groups consisting of an adjective and optional comma. After these groups a noun in singular is required. Its case is assigned to the variable Y. On the rewrite side a new NP is formed from all recognised constructs. That is denoted by three dots. The rewrite inherits its case from the subordinate noun. Rule (c) is an example of solving noun-verb homographs, like “walk” in the phrase “the walk”. It is explicitly marked that a noun instead of a verb follows a determiner. Rule (d) is an example of a transfer rule. It changes ordering of noun phrase subparts so that adjectives are moved after the noun, like in French language. Rules (e) and (f) demonstrate how the formalism allows external procedure calls. The last example suits for expressions like “the small car factory”, which is syntactically ambiguous. The “choose” procedure invokes user interaction so that the user is asked to solve, which of the two alternative interpretations in the curly brackets is semantically right. In general, possibilities to call external procedures or use global flags to control processing are not elegant solutions in a language formalism as they break declarativity. In addition to these presented features, PeriPhrase language also provides WILD patters that match anything. The pattern <N & ADJ> is an exclusion pattern that requires that neither N nor ADJ is found. The pattern {DET | ADJ | N} is and OR pattern that requires that one of the enumerated set of possibilities is found. It is important to note, that PeriPhrase is oriented more towards fixed word order languages than languages with relatively free word order like Finnish. PeriPhrase does not contain any special means for compactly expressing different orderings.

PeriPhrase rules are stored in packets that are activated based on the state of analysis. The rules of a packet can be parametered to be matched left-to-right or right-to-left. When the traversal orders of rules are specified explicitly, the processor searches down inside tree structures when trying to match patterns. Otherwise, pattern matching is limited to roots of the trees already formed.

2.6 TAUM

The TAUM project of University of Montreal (Traduction Automatique de l´Université de Montréal, 1965-1981) was successful in developing syntactic transfer based machine translation. The major outcome was MÉTÉO system that was fully automatically translating weather reports from English into French. The system was installed to operative use in 1976. The MÉTÉO system used a formalism known as Q-system (Q = Quebec) that was developed by Alain Colmerauer and that expressed translation as manipulation of linguistic strings and trees. The syntax of Q-system rules is as follows [Colmerauer 1992, Whitelock & Kilby 1995, Hutchins 1986]:

```
rule ::= pattern == pattern  / condition .
pattern ::= tree  |  tree + pattern
tree ::= variable | label | label(list)
list ::= tree | tree , list | tree , etiquette , list
set_of_trees ::= tree | tree , set_of_trees
variable ::= symbol*
etiquette ::=  /
condition ::= list = list | list =/ list |
               set_of_trees -DANS- set_of_trees |
               set_of_trees -HORS- set_of_trees |
               condition -ET- condition |
               condition -OU- condition | ε
```

The weather expression “sunny with showers today” gets analysed into the tree MET((C(ADJ(SUNNY),CMOD(P(WITH),N(SHOWERS))), T(TODAY)). The compositional analysis is illustrated as a chart in Figure 2.3. Charts are working memory structures that are used to maintain the state of processing. The Q-rules transform such charts to new ones. The chart may correspond to an analysis, to a sentence synthesis or to an intermediate state of manipulation. The *patterns* of the Q-rules specify paths to be found (LHS) from the chart or to be added (RHS) to the chart. If a rule set is preceded by keyword –INV-, the rules are nondirected/invertible.

The translation in TAUM-Météo has five basic stages: morphological analysis of English, syntactic analysis of English, transfer, syntactic generation of French, and morphological generation of French. All stages are specified using the Q-system formalism. For example, the rule STOOD==SW(STAND)+ED(PST) defines morphology for the word form "stood", namely that its stem word SW is "stand" and that its time form is past tense ED(PST). The rule DET(V*)+N(X*)==NP(N(X*),DET(V*)) forms a noun phrase tree from a determiner and a noun. The variables V* and X* get bound to the verb and noun and transfer them to the newly constructed tree. Variables can be present in the rules but not in the tree strings that are in the working memory. Variables are referred in the optional conditions. The possessive form 's is treated in the sample rule NP(U*)+'S+A*(V*)==NP(U*)+'S+ART(DEF)+A*(V*) / A* -DANS- ADJ,N,QUANT –ET- I,YOU,HE,SHE,IT,THEY –HORS- U*. The rule applies for cases where a noun phrase tree not constituted by personal pronoun follows by 's and an adjective tree, noun tree or quantifier tree. -DANS- denotes set inclusion and –HORS- exclusion. The rule produces a string of trees marked up with ART(DEF) to be considered in the transfer. The Q-system rule AN+ADDITIONAL==CIRC(INV(ADDITIONALITY),/,*DEG) contains an *etiquette* "/" that marks the list members on its right side to be interpreted as features of the member on its left side.

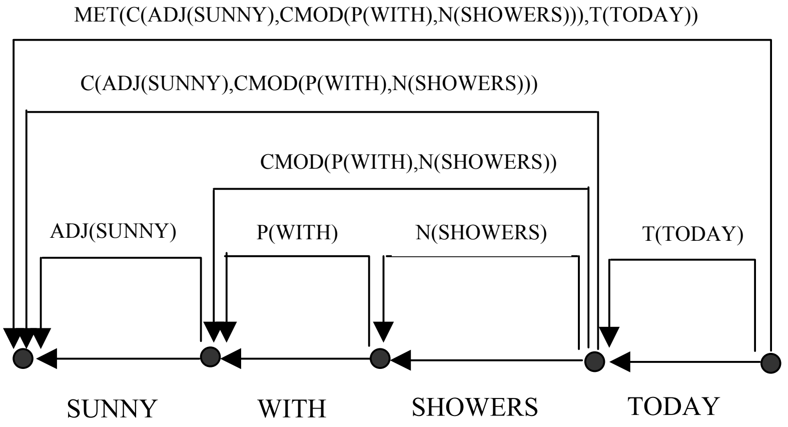

MET(C(ADJ(SUNNY),CMOD(P(WITH),N(SHOWERS))),T(TODAY))
C(ADJ(SUNNY),CMOD(P(WITH),N(SHOWERS)))
CMOD(P(WITH),N(SHOWERS))
ADJ(SUNNY)
P(WITH)
N(SHOWERS)
T(TODAY)
SUNNY
WITH
SHOWERS
TODAY

Figure 2.3: Taum-Météo's syntactic analysis of "sunny with showers today" as a chart.

Q-system rules are compiled into internal data structures [Whitelock & Kilby 1995]. After that there is an interpreter executing a chart traversal algorithm, which traverses the chart in such a way that every path through the chart is matched once against the rule base. New paths may be added by the executed rules. The pattern matcher examines paths of increasing length starting from their rightmost edge. The result of applying a grammar is independent of the order in which the rules of the grammar are applied. The system does not check whether a grammar terminates. It is possible to end up in an endless process. Q-systems have computational power of Turing system.

The Q-system specification of TAUM-Météo used semantic features to denote weather forecast constituents. Such semantic features were used as measure, place, direction, duration, punctual time, possibility, month, day-of-week etc. The system did rather semantics oriented machine translation in its restricted domain.

The Q-system formalism does not provide means for easy copying of features from node to node [Hutchins 1986, Whitelock & Kilby 1995]. Neither does it have a flexible means for enumerating different ordering alternatives. Moreover, passing control between chained Q-systems requires use of “tactical” features (a kind of flags) in the trees, which is a rather confusing way of coding. There is no way of specifying directly that some component of a tree must be at an arbitrary vertical distance from other component. This has been found to cause difficulties, e.g., by severely restricting the possibility of lexical disambiguation from the surrounding structure.

TAUM-Aviation was a follow up project (1977-1980) to develop machine translation system to translate maintenance manuals of the CP-140 aircraft. It was a continuation for the successful TAUM-Météo. The language of manuals was no more as constrained as in the case of weather forecasts. The Q-systems formalism was no more used in every processing step. Instead specialised approaches were introduced to cope with the complexity. For instance, the morphological analysis of English was coded directly as a Pascal program that strips suffixes, adjusts the remaining root forms and assigns morphological features. The program made use of some external linguistic data in the form of a table of special morphological forms. Syntax analysis was based on a variant of Augmented Transition Network formalism [Woods 1970], called REZO. Like ATN it is based on an extended finite-state transition network. The transitions can be labelled with non-terminals to denote calls to subnetworks. They can also include arbitrary tests and arbitrary structure building operations. Finally, last but not least inelegantly there can be retained intermediate values in globally accessible registers to affect later analysis. This sort of machine oriented procedural specification of language syntax can be justified from the viewpoint of computational efficiency. However, nowadays the main focus is in declarative formalisms that provide much better clarity. Taum-Aviation used Q-systems only in structural transfer and syntactic generation. The lucidity of the TAUM-Méteo was lost. The project managed to produce a working prototype, but its development was discontinued. This was decided after finding in tests that the human effort needed for post-editing the translation results was intolerable – twice as much as would have been needed for human translation [Hutchins 1986, Whitelock & Kilby 1995].

2.7 EUROTRA

Eurotra project (1982-1993, EC funded by 37,5 million ECUs, around 160 workers located in 20 centres) was set up to develop a machine translation system that would cross-translate EU documentation between nine official EU languages, namely English, Danish, Dutch, French, Greek, Italian, Portuguese, and Spanish. Although the project did not reach its initial goal, it stimulated human language processing research in the EC countries, developed interesting grammar frameworks for machine translation and fertilised ideas that found continuations in new projects and start-up companies. Figure 2.4 illustrates the transfer architecture of the system and names the intermediate representations of the analysis and generation.

First will be described the <C,A>T formalism, which was used when the author visited the Eurotra-DK team in autumn 1987. Later on the practical use of the formalism proved a bit too complicated and a new more streamlined version, called E-Framework, was introduced [Bech & Nygaard 1988]. Team in Saarbrücken developed another formalism called CAT2 [Sharp & Streiter 1992, Sharp 1994]. Although the Eurotra project itself did not produce a working system, a number of researchers involved continued to work on the theoretical approach developed. PaTrans system developed in University of Copenhagen for English-Danish translation of patents is one concrete result of Eurotra project [Hansen 1994, Ørsnes & al. 1996]. PaTrans uses simpler transfer than was the original goal in Eurotra. The system has been commercialised.

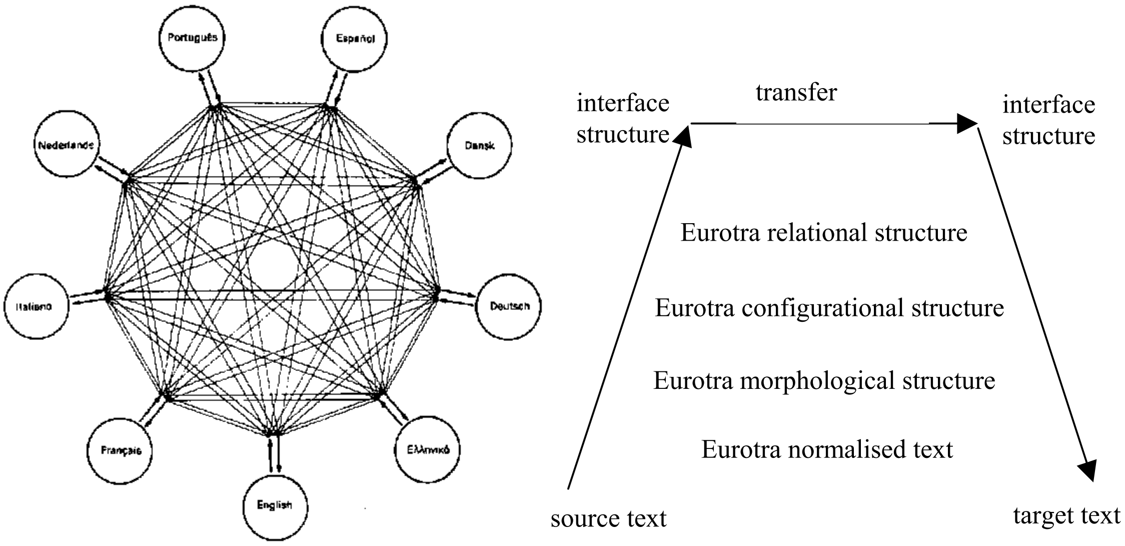


Figure 2.4: The transfer architecture of Eurotra with the analysis and generation intermediate structures..

2.7.1 <C,A>T formalism

The first formalism used in Eurotra was called the <C,A>T model [Arnold & al. 1986, Arnold & des Tombe 1987, Varile & Lau 1988]. In its name C refers to constructors, A to atoms, and T to translators. Constructors can be viewed as non-leave nodes in trees and Atoms as leave-nodes. Translators specify tree transformations, i.e. how new trees/subtrees are assembled from the parts of old trees. The constructors describe partial trees, atoms their leaves, and translators relations between trees. Figure 2.5 illustrates the generic framework with TL_S and TL_T referring to the source and target language texts, RL_n referring to the n^{th} *representation level* and G_n denoting its *generator*. The horizontal arrows denote *translators* between the representation levels. The idea is to set up a pipeline that gradually translates the trees towards the target language text. To formalise the processing steps in each phase, a consistent declarative formalism with semantics based on unification is used.

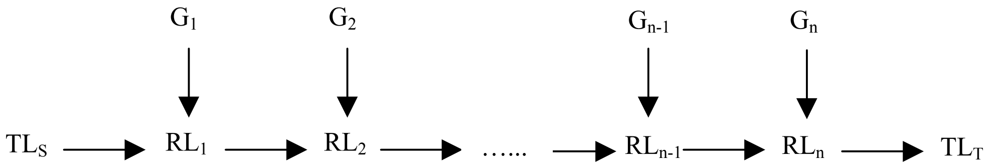


Figure 2.5: Intermediate Representation Levels (RLs) and Generators (Gs) of the <C,A>T framework illustrated

Correct representations are ensured for each level by its generator, which is defined in terms of three types of rules [Arnold & al. 1986, Arnold & des Tombe 1987, Varile & Lau 1988]. The syntax of the formalism as in autumn 1987:

1. B-rules, which are augmented context free structure **b**uilding rules.
   The syntax of B-rules is:

```
B-rule ::=  rule-id = ( name, { featspec .. last_fspec } ) |
            rule-id = rootspec.[ argspec .. last_argspec ]
rootspec ::= arspe
argspec ::= last_arsgpec, | ε
last_argspec ::= arspe | *arspe | ^arspe | ε
arspe ::= name | ( name, { featspec .. last_fspec })
featspec ::= last_fspec, | ε
last_fspec ::= attribute = value |
               attribute == value |
               attribute ~= value | ε
```

   The example `cpp = pp.[^(adv, {scat==mot}), prep, np]`
   defines a prepositional phrase consisting of an optional (^) adverb with semantic category mot, a preposition and a noun phrase. When launched this rule builds a new tree with pp as the root with the mentioned subordinate structures.
   The example contains == operator which tests that the named attribute is already present in the feature description of the construct and is having the given value. Other possible operators would have been = which has unification semantics and ~= which states that the named attribute should not have the given value. Kleene star * is used for expressing zero, one or more instances. Different variants of a construct can be either specified as distinct rules or using disjunctive operator ";" inside rules.

2. F-rules (also called gentle-rules), which define **f**eature percolation principles in trees. Their syntax resembles B-rules with the addition of variables accepted in the rules. These are denoted by capital letters, in case of being anonymous by question marks. The example `agnc = (np, {nc=X}).[*, (n, {nc=X}), *]` defines that a noun phrase (np) gets its number and case (nc) from its noun (n). Here the = operator functions like in unification and X can be considered as a once bound value carrier.
3. K-rules, which control over-generation and abolish illegal constructs. There are two types of K-rules, strict-rules and killer-rules. Strict-rules specify what conditions an object must fulfill in order to survive. The object is retained if in any single match unification succeeds. The second type of rules defines negative instances of objects. These are called killer-rules. The syntax of both types of rules resembles the syntax of F-rules. The killer –rule `s_mod = s.[*, (mod, {pform==of}), *]` defines that sentences cannot be modified by prepositional phrases denoted by "of".

Although syntactically similar, the different semantics of the rule types and subtypes is parametered to the rule compiler. This produces different operational semantics for them.

Information is translated from one level to the following level by the translators that are said to be compositional, direct (i.e. "one-shot") and simple. Translation is compositional when the translation of a complex expression is some (reasonably straightforward) function of the translation of the basic expressions it contains, plus the translation of their mode of combination. A central problem in the Eurotra project was to keep the transfer components as simple as possible because the number of required components grows exponentially with the number of languages covered. The translation rules are in principle limited to lexical component. They have their own syntax:

4. T-rules (or translation rules) look like the following one taken from Italian analysis component:
   ti7 = (s, {coord=yes}).[$S1! (s, {coord=no}), $CONG! conj, $S1! (s, {coord=no})]
   => coord($CONG, $S1, $S2)
   The rule builds a new tree with root "coord" and conjuction as the first subtree head, and the subclauses as the following ones. The variables starting with "$" get bound to the subtree roots of the original syntax level s tree.

2.7.2 E-Framework

When grammars in the <C,A>T formalism grew to have substantial linguistic coverage, the rules describing the mapping between levels became highly complex and numerous due to the interdependence between the linguistic phenomena triggering structural changes in a representational tree. The goal of the E-Framework became to simplify this by providing a mapping system that requires only partial target level tree descriptions [Bech & Nygaard 1988, Crookston 1990]. The E-Framework consists of generators, which interpret grammars, and translators, which interpret *t-modules*. A grammar defines a level of representation and a t-module states the relation between source level trees and their corresponding partial descriptions to be completed by the generator at the target level. The translator of the E-Framework is quite weak as it can only provide guidance for the construction of target objects, but it cannot build them itself. This is a major deviation from the <C,A>T formalism. The route from representation level A to B is **reprA** → translator → **descriptorAB** → grammar → **reprB** (boldface used for representations).

Translators are defined in terms of t-rules. Their general form is:

```
t-rule ::=  srctree  =>  tgtdescriptor
srctree ::=  identifier:featurebundle[stree .. srctree]
stree ::= srctree,
featurebundle ::= { featspec .. last_featspec }
featspec ::= last_featspec,
last_featspec ::= attribute = value | ε
tgtdescriptor ::= identifier | ε |
                  identifier [ descriptor .. tgtdescriptor ] |
                  identifier < descriptor .. tgtdescriptor > |
                  identifier ( descriptor .. tgtdescriptor )
descriptor ::= tgtdescriptor,
```

Curly braces delimit feature bundles, square brackets denote immediate dominance, and angle brackets just dominance. Precedence is unspecified inside parentheses, while outside them precedence is as written. Identifiers are written in capital letters and they function in the rules as value carrying variables. The t-rule S:{cat=s}[V:{cat=v},SUBJ:{cat=np}] => S<(V,SUBJ)> applied to the source level representational tree representing the sentence “the woman works” {cats=s}[{cat=v, lu=work}, {cat=np, defness=definite}[{cat=n, lu=woman}]] produces the descriptor:

```
{cat=s, …} < ( {cat=v, lu=work, …},
               {cat=np, defness=definite, …}
                   < {cat=n, lu=woman, …} > ) >
```

The “subtrees” in the descriptor are thus unordered. In the next step the grammar rules are applied to further specify the tree. The grammars are defined in terms of g-rules which are hierarchic feature patterns denoting trees and used by applying unification. Unification with the g-rule

```
{cat=np, defness=D, …}
  [  !{cat=detp, defness=D, …},
      {cat=n, …}
     ^{cat=pp, … } ]
```

produces the subtree

```
{cat=np, defness=definite, …}
  [ {cat=detp, defness=definite, …},
     {cat=n, lu=woman, …} ]
```

to replace the np-subtree of the descriptor. In the g-rule the "^" prefix denotes that the following feature bundle is optional while the "!" prefix means that the feature bundle must be added if not present. Variables are written in capitals. Thus a detp construct was inserted into the tree. Further rule {cat=vp, ..}[{cat=v, …}, ^{cat=np, …}] builds a vp-subtree, the rule {cat=s, …}[{cat=np, …}, {cat=vp, …}] standardises the ordering of the tree into direct word order and the rule {cat=detp, defness=D, …}[{cat=det, defness=D, …}] adds the determiner to the dept-subtree. The overall tree transformation from the source level representational object into target level representation is depicted in Figure 2.6. Linguistically these tree transformations belong to the synthesis of the machine translation system. The source representation has canonised ordering of constituents and the target representation is made to follow the structure of the target language.

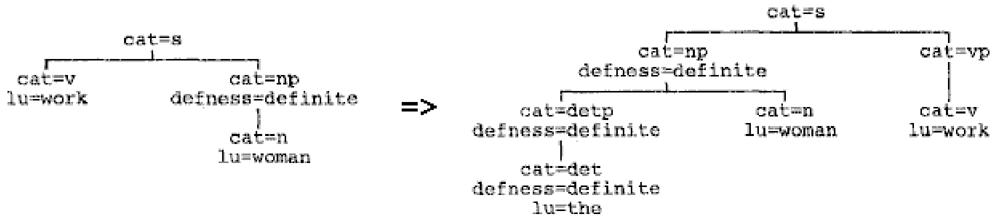


Figure 2.6: Source tree and the result after applying the t-rule and the grammar rules.

2.8 LRC MT System / Metal / Litras

The Linguistic Research Center (LRC) of the University of Texas began working on MT in 1961. The work was first based on interlingua paradigm and influenced by transformational grammar. Difficulties in transformational parsing and interlingua development lead to taking a new approach in 1979, which resulted in the LRC Machine Translation System, later on renamed to Metal and commercialised in 1989 as Litras. The performance of the system was reported to be very good, both in terms of linguistic capabilities and computing speed. It became very famous after Siemens Corporation started to commercialise it for German-English translation. There are many general descriptions of the system [Slocum 1983, Slocum & Bennett 1985, Bennett & Slocum 1985, Hutchins 1986, White 1987, Whitelock & Kilby 1995]. Unfortunately none of the references gives a comprehensive and precise description of the underlying formalism. The following description has been gathered from pieces of examples found in the references. Next the system is referred by name Metal, as it has been its most widely used name.

METAL has a modularised transfer design, with monolingual and bilingual/transfer dictionaries, a bottom-up chart parser, fail-safe heuristics, and batch processing with post-editing on PC workstations. Metal shares with Systran the property of being specified near the executable program code. In its case, the language specifications are formulated in LISP language syntax so that they can be directly interpreted for execution. Due to this close relationship, there is no longer in METAL a strict separation of algorithmic and linguistic data. This inelegant approach has been chosen in order to get good computational performance.

```
(a)    (THIS CAT (DET)
             ALO   (this)      (these)
             NU    (SG)        (PL) $
             PLC (WI WF)
             ... )
(b)    (     NN          NST         N-FLEX
             0           1           2
             (LVL 0)     (REQ WI)    (REQ WF)
             TEST  (INT 1 CL 2 CL)
             CONSTR (CPX 1 ALO CL)
                         (CPY 2 NU CA)
                         (CPY 1 WI)
             TRANSF (XFR 1)
                         (ADF 1 ON)
                         (CPY 1 MC DR) )
```

Figure 2.7: Examples of language specifications in Metal.

Examples in Figure 2.7 demonstrate language specifications in Metal. The expression (a) contains part of the lexicon entry for word “this”, which is said to have category determiner (DET), two actual surface string representations (ALO, allomorph) “this and “these”, which have respectively singular (SG) and plural (PL) as their number and jointly (e.g. after $) word initial (WI) and word final (WF) as their placement (PLC) features, e.g. words can only occur as unbounded forms. The system uses a set of feature types and feature values that are referred to by two or three character names. This makes the grammars look like assembly language code. Another characteristic is the column wise associating of information from a line to the respective items on the previous line, which is also used in the German syntax rule (b). It states that a noun (NN) consists of a noun stem (NST) and an inflected nominal ending (N-FLEX). The rule is a top-priority rule (LVL 0). Its column tests state that the first element must be word-initial and the second word-final. The TEST part states that the intersection (INT) of values of feature CL of the first element and second element must be non-empty. If rule application gets this far, a tree will be constructed along the CONSTR part by copying (CPX) to NN all features associated with the stem form except for the allomorph (ALO) and inflectional class (CL). The first CPY copies the grammatical number (NU) and case (CA) features of the inflectional ending and the second CPY the WI feature of the stem form. The TRANSF part is activated during transfer phase and it derives corresponding English structure from the German one by copying and adding features. Apart from the expressions in Figure 2.7 the presentation formalism provides also syntactic tools to state case frames for verbs and transformation rules for modifying syntactic structure, both along the same LISP programming style as in the examples.

Metal uses bottom-up parser, which applies rules along to their numerical levels (LVL) by starting with rules at lower level before applying any at a higher level. This makes rule application partially ordered, yielding more efficient processing but also reducing declarativity as control strategy information gets coded into the rules.

2.9 Summary

This chapter handled different approaches of machine translation. Section 2.4 discussed various properties of language specification formalisms. Formalisms used in a set of major machine translation projects or systems were presented. Table 2.1 summarises some of their characteristics. The evaluation is based on the references and on the author's general understanding of language specification. Precise evaluation would have required hands-on experiences from personal use of each formalism for some weeks. This was not possible to organise. Thus, the scoring uses a rough scale "*good – moderate – poor"* and additional verbal explanations. In some cases the judgement is totally based on the author's reasoning. This is signalled by the word *"likely"* in front of the scoring.

In addition to the characteristics included in Table 2.1 the following observations can be done of the reviewed formalisms:

- All the reviewed formalisms are language independent in principle. However, only E-Framework provides in its t-rules means for compactly expressing different orderings. That would be important in case of languages that have relatively free word order, like Finnish.
- Only the Q-system formalism of TAUM supports nondirected/invertible rules.
- Q-system and the formalisms of Eurotra imply stratificational processing model although the same descriptive instruments are used in all phases. With the junction grammar and the knowledge representations (KR) of Metal the stratification is even more explicit. However, PeriPhrase appears to be less directed to stratification.
- It is easy to see that formalisms involving negative evidence have diminished monotonicity. However, with the rest of the formalisms evaluation of monotonicity is very difficult. Likely the Q-system formalism of TAUM has the best qualities with respect to monotonicity.
- All of the formalisms support multidimensional patterns. The formalisms of Eurotra support complex feature structures as well.
- The error tolerance of Metal has been reported to be very good. In Eurotra use of over-generative grammars together with the killer-rules could in principle provide an effective way for engineering error tolerance.

	ITS: Junction Grammar	ALPS: PeriPhrase	TAUM: Q-System	EUROTRA: <C,A>T	EUROTRA E-Framework	METAL: Knowledge representa-tion in Lisp
Linguistic felicity	Likely poor due to its procedurality and intercon-nectivity of KRs	Likely good due to its simplicity.	Likely moderate due to its linguistic limits and need to use "tactical" features	Likely moderate due to intercon-nectivity of its levels and the alternative ways to repr.	Likely quite good.	Likely moderate due to its procedurality.
Expressiveness	Good	Good	Moderate, suitable for restricted syntax with no anaphora etc.	Good, rich set of instruments	Good	Good
Declarativity	Poor, procedural	Moderate, due to side-effective procedure calls and global flags	Good	Good	Good	Poor, mixing algorithms and knowledge descriptions
Uniformity	Moderate, tree trans-form prog-ramming	Good, using PeriPhrase rules in all phases	Good, using Q-system rules in all phases	Moderate, many rule/ expression types	Good, better than <C,A>T	Moderate, many expression types
Reusability	Poor	Likely good due to simple rules	Likely good due to simple rules	Likely moderate	Likely moderate	Likely poor
Implementability	Poor, in practical tests	Good	Good	Moderate, heavy to process	Good	Good, boasts about efficiency

Table 2.1: Summary of some characteristics of the reviewed MT formalisms.

3. Augmented Lexical Entries

This chapter presents the Augmented Lexical Entries (ALE) formalism. The first section introduces the philosophy behind it. The following section describes the syntax of the ALE formalism. Section 3.3 gives examples of how ALEs can be used. Section 3.4 discusses about ways of implementing ALEs. Section 3.5 compares ALEs to formalisms presented in Chapter 2. Section 3.6 reports of the experiences from applying automated grammar discovery with ALE formalism. Webtran machine translation system is one implementation of the ALE formalism. Section 3.7 tells about experiences from using it in production use. The chapter ends with a short summary.

3.1 Introduction

Augmented Lexical Entries are used to carry the linguistic information needed to both define and translate controlled languages [Lehtola et al. 1998, 1999b]. The target languages are used to describe entities of a restricted domain, like mail-order clothes or rental cottages. The target texts usually have great semantic homogeneity rather than syntactic homogeneity. Because of the semantic homogeneity the conceptual aspects of the texts become more relevant than the structural ones also in translation. For example, Steimann [Steimann & Brzoska 1995, Steimann 1998], who used conceptual model for representing a medical language with a dependency grammar had taken similar bias. In the Mikrokosmos KBMT system [Beale & al. 1995], the concept representation, called *Text Meaning Representation*, serves as an interlingual component.

Choice was made to use dependency grammar instead of phrase structure grammar as the basis of ALE formalism for two reasons. Firstly, dependency grammar has been found very suitable for analysing relatively free word order languages, such as Finnish. In Finnish the words themselves carry a lot of semantic information in their forms thus relaxing the ordering of words. Swedish language, subpart of which, namely Ellos' Swedish, was the first source language used for Webran, also has inflectional word forms. Secondly, syntactic dependency relations coincide rather closely with semantic relations holding between the same entities (e.g. [Hajicova 1987]). This has been noticed also by some research groups, which are developing text mining and using dependency analysis as a processing step [Faure & al. 1998, Faure & Nedellec 1999, Maedche & Staab 2000a & 2000b]. The Mikrokosmos MT system uses dependency-directed processing for semantic analysis and treating ambiguities [Beale & al. 1995], as well.

Dependency grammar can be written in concordance to the semantic relations found from a conceptual domain model. The parsing would then produce a dependency tree, which would reflect also the semantic relations defined in the conceptual model. The approach in Webtran became to use conceptual models, when defining a new controlled language. For instance the lexical semantic categories are based on it. Figure 3.1 presents a small conceptual model of women's clothes as covered in mail-order product descriptions of Ellos.

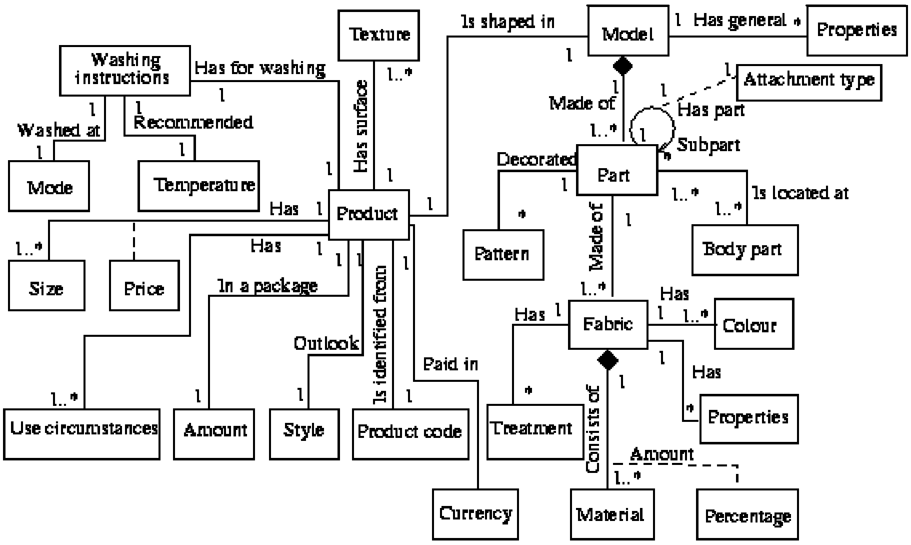


Figure 3.1: A small conceptual model of the domain of women's clothes.

The Augmented Lexical Entries formalism can be characterised by the following:

1. Describing simple phenomena is simple:

   The great majority of the knowledge to be described is supposed to be rather simple. Ordinary language translators should be able to specify the regular things using simple expressions and afterwards to test that they are functioning. This requirement sounds obvious, but it is not met in all systems. For instance, the stratification in the Eurotra approaches leads often in distributing a phenomenon into several knowledge description parts, e.g. tailoring a path through the translation model. ALEs try to provide an explicit presentation that localises one phenomenon into one place.

2. Complicated phenomena can also be described:

   The assumption is that only small minority of the grammar knowledge is complicated. Modelling of the complex phenomena may require a language engineering professional. Complexity is allowed to lead to complicated grammar expressions. ALEs strive to be a simple notation with high expressive power.

3. Declarative and intuitive notation:

   The notation should contain minimally procedural information thus relieving the user of the notation from thinking how the expressions relate to the overall computational process. Clarity and compactness of the notation is enhanced, for instance, by allowing reference to features directly by their values thus making the expressions compact. The conventional way would have been to refer by <*feature_name,feature_value*> pairs. Reference straight by value is possible, if all acceptable values are unique and belong to a single category.

4. A uniform way of representing phenomena on the different levels of language:

   All features of constituents, whether they are lexical, syntactic or semantic, should be possible to access in all grammar expressions. There should not be, e.g., any processing model based hindrance against the use of the most effective constraints when found necessary. ALEs endeavour to enable flexible and uniform use of different types of constraints.

5. Monolingual, bilingual or multilingual non-directed entries:

   In order to further elaborate the idea of uniformity, the notation should be generally applicable for presenting relationships between both constituents of a single language (= syntactico-semantic dependency relations) and constituents of several languages (= translation relations). Such entries can serve as a knowledge base for monolingual analysis, monolingual synthesis, and bilingual or multilingual translation. ALEs strive to be versatile in this respect. Moreover, they can be specified to be invertible if required.

6. Suitable for fully automated or machine supported language modelling:

   Machine learning is an important technology for coping with the knowledge discovery bottleneck of machine translation, which is due to the overwhelming amount of details that need to be specified for a translation system. Solely relying on manual work would make the task extremely expensive. The notation should be suitable for automated discovery of translation grammars based on parallel bilingual corpora. To be well suited, it should provide means for presenting translation relations starting from surface level phrasal correspondences up to generalised more abstract correspondences. ALEs aspire to provide such expressive means that would suit well for the learning task.

The next section describes the syntax of ALE formalism. After it, there is a section illustrating the use of ALEs by examples. That section will make references to the six characteristics described above.

3.2 Syntax

The general form of an augmented lexical entry is shown in BNF notation in Table 3.1. The symbol ⊕ denotes any ISO standard language code. Optional nonterminals start with *opt_* .

```
augmented_lexical_entry ::= [      entry_name
                                   pattern..   opt_message   opt_repair   ]

entry_name ::= name . integer_index

name ::= hierarchical_name_with_dots_between_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 ::=   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_reference..  }

binding  ::= (  variable_name  )  |  (^)

feature_reference  ::=  feature_value | feature_type    binding
```

Table 3.1: The syntax of the augmented lexical entries.

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 use of language, when manuals are maintained in one language only. A multilingual online catalogue service would also contain multilingual entries to provide the translation relations.

An ALE contains a pattern for each language it covers. The ISO language codes are used to mark these language-specific patterns. In case of specifying negative instance of a language its ISO code is preceded by negation operator (~). The number of languages in an ALE is not limited. A pattern also contains the constituents (see Appendix A) of the corresponding language expression in their matching order. If the specified constituents are bounded by angle brackets, they may appear in any order. 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 surface form word can be specified by giving its lexical key (*lexeme*) and in case of an inflected form its set of morphological feature values. Interactive messages and repair patterns consist of alphanumeric strings that may contain bindings (*string_w_opt_binding*).

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. In ALE formalism, variables have a capital character in their beginning (*variable_name*). Variables are important in forcing constraints among constituents and specifying translation relations. They carry constituents as their values. A variable can be bound only once to its value, when an augmented lexical entry gets selected through matching. Variables are scoped by the entries.

Constituents may have feature constraints specified as sets of features. Features are used to denote required lexical, syntactic or semantic properties. Features are referred to directly by their values, which necessitates that there is only one known type or category for each used value. Feature values (*feature_value*) and feature types (*feature_type*) are alphanumeric strings that do not start with capital characters. In order to be able to specify agreement between constituents, there is provided a way to refer to a feature value of a constituent using its feature type and variable reference.

Reference to a larger lexicon is considered implicit in augmented lexical entries. In order to avoid need to specify thousands of trivial ALEs relating just single words to their correspondents in other languages, an elementary bi/multilingual lexicon of simple word correspondences is assumed to exist. This lexicon can be in its simplest form a table with columns representing different languages and rows words with their features and their translations. Thus in ALEs it is possible to concentrate into constituents of several words and into generalised constituents that leave the actual words open and just specify allowed categories and other features. Following the controlled language paradigm, the lexicon should provide only non-ambiguous translations for its words.

The nonterminal *tag* represents tag names, which depend on the implementation of the ALE formalism into some particular use. Tag marks are given by the text tokeniser to special characters, numeric quantities, typographic and other mark-ups etc. For instance, in the case of the product catalogues of Ellos the following tag names are used in grammars: tag_centig, tag_comma, tag_dot, tag_end_of_product, tag_fahren, tag_fieldseparator, tag_html, tag_image, tag_lengthunit, tag_minus, tag_number, tag_parclose, tag_paropen, tag_percentage, tag_plus, tag_price_end , tag_prodcode, tag_quotationmark, tag_size, tag_slash, and tag_tab.

The rules are equipped with hierarchical names consisting of name strings concatenated with dots (*hierarchical_name_with_dots_between_parts*) and ending into an integer (*integer_index*). The integer is used for numbering different instances of rules that are sharing the same name. The names can be used to relate the rules to the conceptual domain model. They can be used to cluster the knowledge base so that it can be maintained more easily. The names can be also used for activating particular parts of the knowledge base while translating. This may be practical, for instance, when document syntax already tells, what subdomain a paragraph of text handles. The document syntax may be marked, e.g., by using XML.

3.3 Examples

Tables 3.2, 3.3, 3.4 and 3.5 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 Section 3.1 “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. Variables 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 Section 3.1 “A uniform way to represent phenomena on the 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.

(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 3.2: 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 interdependency of the words denoted.

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 3.2 the last word of the Finnish pattern is the only one that is inflected (case overridden to partitive indicated by "ptv"). 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 3.2 function in multiple directions.

The characteristic 6 of Section 3.1 "Suitable for fully automated or machine supported language modelling" is demonstrated by the entries in Table 3.2. It is easy to see how the surface level entries can be used for translation memory type of learning. Generalisations, like in the entries (c) and (d), could be deduced automatically based on sets of simpler entries.

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 3.3 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 3.3 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 3.3 produce a parse tree of depth one. The entries in Table 3.3 demonstrate the characteristic 2 "Complicated phenomena can also be described" mentioned in Section 3.1.

```
(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 3.3: 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.

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 human language to a controlled language. The correction rules cannot have full coverage of unrestricted human language. If the machine could understand unrestricted language, there would not be any need for the controlled language on the whole. 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.

```
(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 3.4: 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.

Table 3.4 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. The word “och” in Swedish equals to “and” in English.

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 3.5: Specifying structure of a cloth description.

Table 3.5 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 these conceptual models there is an ontology editor as part of VTT's CONE (COnceptual NEtwork) software [Kankaanpää 1999].

All of the presented entries evidence the characteristic 3 "Declarative and intuitive notation" mentioned in Section 3.1. The entries in Table 3.2 are easy to understand and to write also by professional translators. The ALEs provide a constraint programming way of specifying the grammars.

3.4 Algorithms and Implementation

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. Dependency parsing as a bottom-up process of recognising linguistic binary relations was considered in [Lehtola 1984, Nelimarkka & al. 1984]. The approach used was to apply elementary two-way finite-automata consisting of *Left-Stack-of-Constituents*, *Current-Regent-Register* and *Right-Stack-Of-Constituents*. The constituent in the *Current-Regent-Register* seeks for valid binary relations with respect to the tops of the stacks. In case of a valid relation the found dependent is added to the partial dependency tree where the regent is a root and the dependent is removed from the stack. The regent from this onwards represents also its dependent. Only roots of partial dependency trees (initially depth=0) are considered. The automaton is capable for moving its focus both leftwards and rightwards. Also the algorithm [Covington 2001] focuses on one binary relation at a time. It uses as a working memory *Headlist* and *Wordlist* and can move its focus to the right.

The article [Jäppinen & al. 1986a] formalises dependency grammar in terms of partial trees of depth one and presents an algorithm for those. The article [Valkonen & al. 1987] employs a blackboard mechanism for the book keeping of the partial constituents when parsing with two-way finite automata. Deterministic dependency parsing is handled in [Jäppinen & al. 1988a, Lassila 1989] and in [Arnola 1998].

The current implementation of the Webtran software compiles the ALEs for run-time use into optimised Prolog clauses, which in turn can be compiled using a Prolog compiler. Its coverage is described in the user guide [Jaaranen 1999]. We have also investigated a hybrid approach, where the strategy would change along the properties of the entries. Such a hybrid approach has been implemented earlier for context-free grammars [Hyötyniemi & Lehtola 1989]. The VTT's ALE compiler and the applied parsing and translation algorithm are not handled in this thesis. However, a simple "reference algorithm" applicable for the ALE MT formalism is described next. Its functioning is illustrated with an example.

3.4.1 Reference algorithm

As described earlier, by marking the regent constituents an ALE grammar is turned into a forest of partial dependency parse trees of depth one. The dependency relations of an entry are expected to hold between adjacent constituents. Naturally, these constituents may have further structure, i.e., they may present phrases or sentences. As mentioned, two-way-automata formed by lists or stacks have been popular in implementing dependency analysis. The reference algorithm also uses the idea of moving focus in both directions. In case of a successful match the whole covered list segment is reduced into a new dependency tree with the regent of the matched entry as the root and all the rest of the covered constituents as subordinates.

The reference algorithm considers all constituents to be dependency trees. In the starting state, all of these trees have depth 0 and consist solely of the root constituent, representing one input text token together with its lexical information. The working memory structure, which is used for book keeping during translation, is called constituent chart. The chart includes following types of relations between constituents: *concatenation relation* → , linguistic *dependency relation* ⇢ , *translation relation* ↶↷ , and *derives-constituent relation* ⤑ . Constituents chained by concatenation relations are said to be on the same path. Each path has an entry point and an exit point. The algorithm is presented together with an example, which assumes the rules A and B, and the initial constituents given in Table 3.6.

```
(A)          [cloth.property.100
                   [en     < (A){adj model} (B){adj colour} >
                           ^(C){noun cloth}]
                   [fr     (A){adj model gender(C) number(C)}
                           ^(C){noun cloth}
                           (B)(adj colour gender(C) number(C)} ] ]
(B)          [cloth.material.100
                   [en     ^(A){noun cloth} of (B){noun material} ]
                   [fr     ^(A){noun cloth} en (B){noun material} ] ]

{ lexeme(long) syntcat(adj) semcat(clothmodel) }
{ lexeme(white) syntcat(adj) semcat(colour) }
{ lexeme(skirt) syntcat(noun) number(sg) semcat(womencloth) }
{ lexeme(of) syntcat(prep) }
{ lexeme(cotton) syntcat(noun) number(sg) semcat(textilematerial) }
…
```

Table 3.6: Rules A and B and the initial constituents of the example.

The following algorithm assumes that the translated language can be analysed into projective dependency trees and that each word has only one regent.

Translate (Text)

1. Segment the text into a list of tokens.
2. Build the initial constituent chart from the list of tokens by associating lexical information to the tokens. Insert precedence relations between the constituents to form a path (Figure 3.2).
3. Matching of rules until no more rules match:
   - 3.1 Select entry *e* from the ALE grammar such that its source language pattern can be matched with a segment of consecutive constituents of one of the paths and that *e*'s instantiation is unique.
   - 3.2 Add to the constituent chart the instantiated constituents of the matched rule. Mark bypassing of the covered part of the path with a path segment consisting of either the regent of the rule (Figures 3.3 and 3.4), or with the sequence of the instantiated constituents if no regent has been marked
   - 3.3 Retain an association from the added constituents to the matched rule.

4. Select the shortest path through the chart and for each constituent on that path do the following:
   4.1 For each ALE instantiation *i* accessible directly or through *derives-constituent* relations do the following:
      4.1.1 Construct and add to the chart the target language constituents and their interrelations according to the target language pattern of *i*.
   4.2 Link the leave constituents of the formed new tree with the concatenation relation
5. Generate surface presentations of all target constituents that are linked with concatenation relation.
6. Return the formed translation.

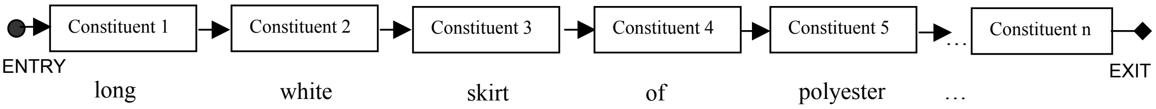


Figure 3.2: The initial constituent chart consists of concatenation relations of constituents formed by incorporating lexical information to the tokenised input.

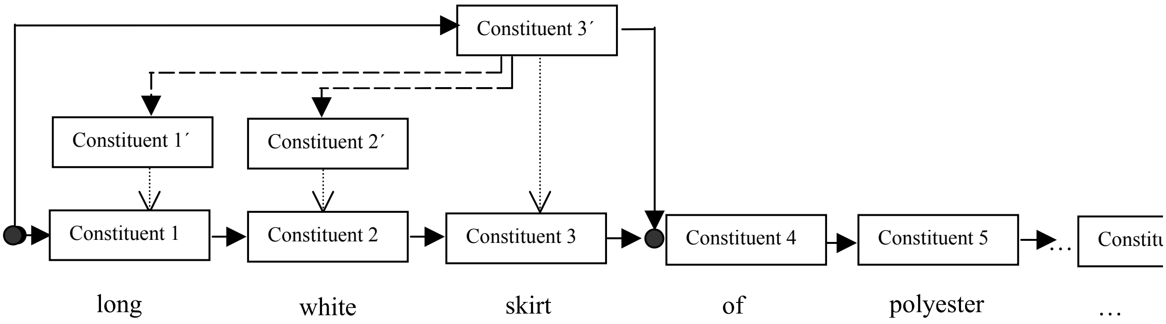


Figure 3.3: The constituent graph after instantiation of rule A.

The algorithm refers to the elementary operation of matching and instantiating ALEs. A few words are necessary to explain what happens there. Applying an ALE to a path segment yields matching each of its constituent definition with a path constituent in the order specified by the ALE. A constituent specification matches with a constituent, if its optional lexeme part matches and if the constituent fulfils the feature constraints specified in the ALE. When an ALE is matched its variables are bound to the matched constituents. A variable can be bound only once. Features of constituents can be tested using variables as in ALE (A) of Table 3.6. There is checked that the two adjectives agree in number and in case with their regent noun. Variables are also used to transfer information between the language specific patterns of the ALE. What is implicit in transfer is the handling of lexical correspondences. The lexemes in the source language side are used to retrieve corresponding words in the target languages from a simple tabular multilingual lexicon. The reference in an ALE to another ALE can be handled like a macro in programming languages. In that case the resolved combined ALE is used for the matching.

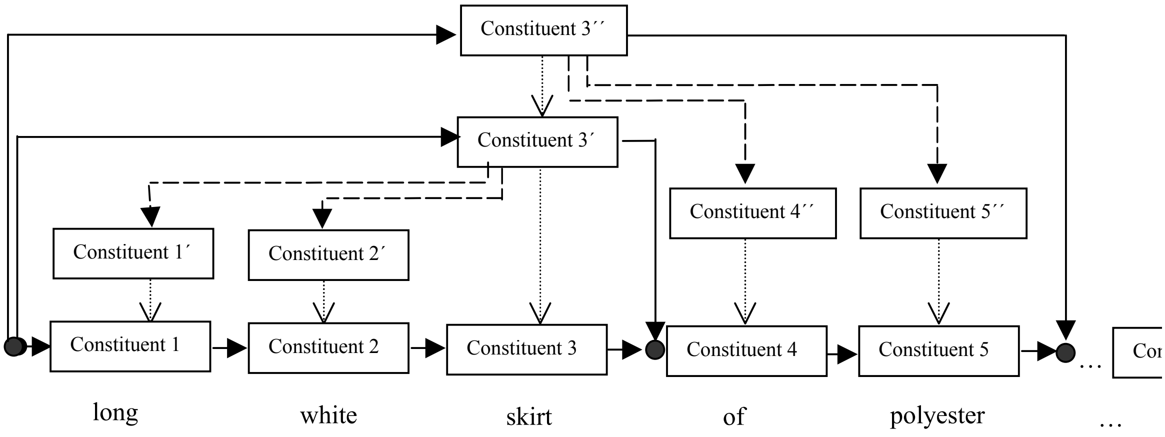


Figure 3.4: The constituent graph after instantiation of rule B.

The analysis part of the algorithm makes exhaustive search with an uninformed control strategy and gives each ALE equal priority for testing. Step 3 is the place where most of the optimisations could be done. As currently described, the processing time would increase very steeply, when the length of the input grows. Exhaustive search can be justified with the need of uncovering all ambiguous analyses. However, considering that the controlled language paradigm strives to eliminate ambiguities, this sort of search might be unnecessarily heavy. An easy way of raising efficiency would be to modularise the ALE grammar using the entry name facility. During the analysis different groups of ALEs would be made active following their linguistic expectancies. The idea would be to use similar hierarchic classification as described in [Lassila 1989]. E.g. first entries related to noun phrases, prepositional phrases and later those having active verbs. An interesting idea would be to bootstrap the rule selection into more informed by machine learning from the successful analyses proper orderings of rule selections and their probabilities.

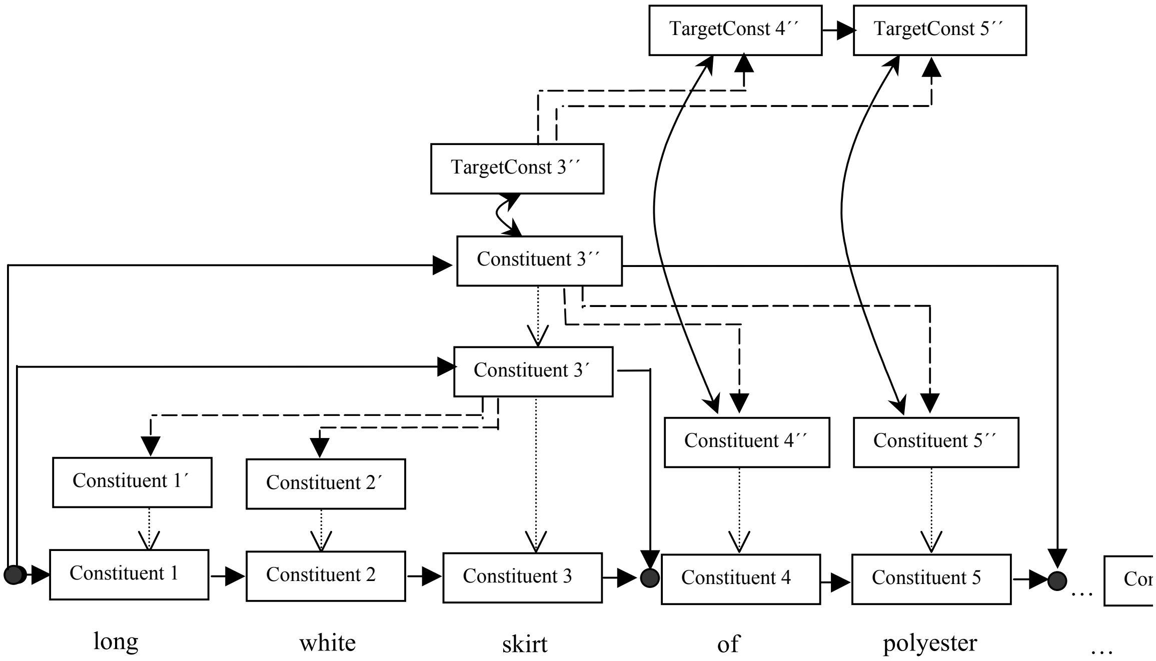


Figure 3.5: The constituent graph after transfer of the constituent part covered by rule B.

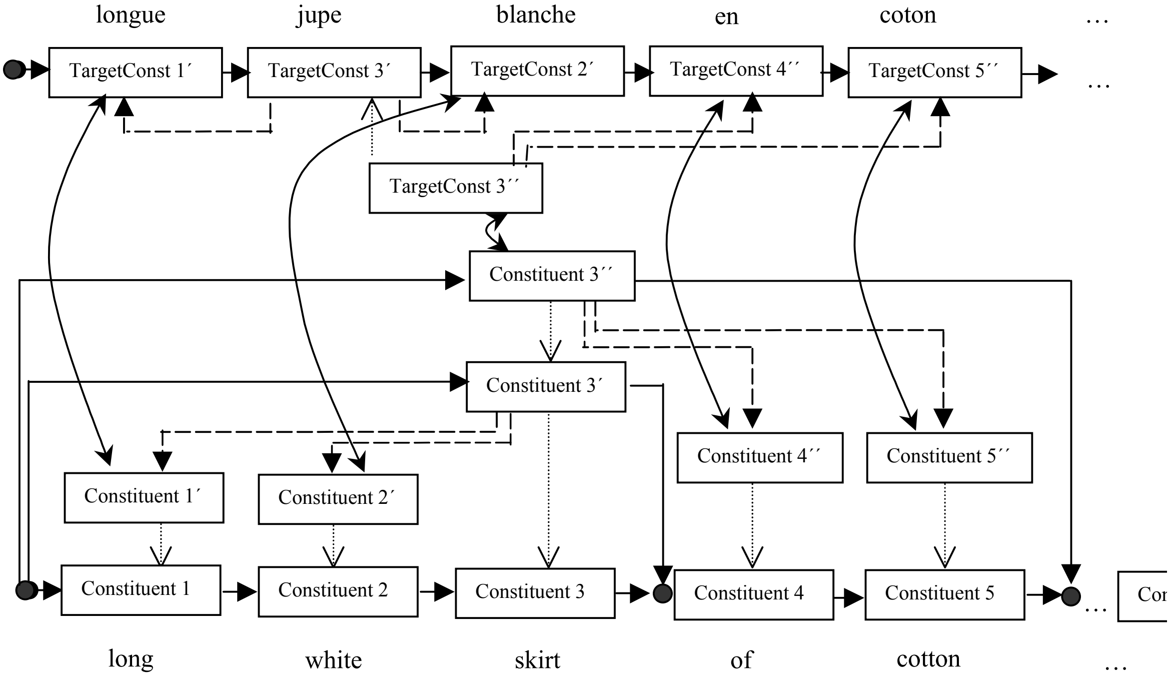


Figure 3.6: The constituent graph with the part covered by rule A transferred and concatenation established.

3.5 Comparison

Augmented lexical entries formalism owes to many of the formalisms handled in Chapter 2 or developed during earlier research work of the author and desribed in [Lehtola 1984, Nelimarkka & al. 1984, Lehtola & al. 1985, Jäppinen & al. 1986a, Valkonen & al. 1987, Lehtola & Honkela 1988, Lehtola & al. 1988a, Lehtola & al. 1988b, Jäppinen & al. 1988a, Jäppinen & al. 1988b, Hyötyniemi & Lehtola 1989]. The next list summarises some of the properties of the ALE formalism and mentions the formalisms that have been used as examples:

- **Use of dependency grammar instead of phrase structure grammar:** The author's experiences in earlier Kielikone project have motivated the choice. Dependency grammar makes the basis of Kielikone's syntax solutions. Dependency structure is close to semantic structure and thus facilitates semantics oriented machine translation. Moreover, dependency grammar has been found better suited for relatively free-word order languages with inflectional words. None of the reviewed MT formalisms in Chapter 2 is based on dependency grammar.
- **Multipurpose rules applicable for both monolingual syntax analysis and translation:** Ideas adapted from PeriPhrase (cf. Section 2.6) to dependency grammar based approach.
- **Nondirectionality / invertibility of entries:** It is possible to make ALEs that can translate in both directions. From the reviewed formalisms of Chapter 2 also the Q-system formalism can be used nondirectionally.
- **Single-binding of variables:** Ideas adapted from logic programming and unification based grammars [Pereira & Warren 1980, Shieber 1986, 1992]. In ALEs the variables get bound to constituents. The features referred through variables are always features of the root constituent of the partial dependency tree.
- **Reference to features directly by their values:** This idea has not been widely used in NL formalisms although it makes specifications clearer and more compact. The model was adapted from the DPL formalism of Kielikone project [Lehtola 1984, Lehtola & al. 1985].
- **Negative evidence:** The idea of supporting negative evidence emerged from the Eurotra project (cf. Section 2.8). However, it was not implemented to effect more generally than to the text checking purposes. The implementation is not as procedure oriented as in PeriPhrase (cf. Section 2.6).
- **Single type of rule:** To enforce uniformity in the grammars, it was decided to strive to just one type of rule limiting the choice of the grammar modeller. In that respect ALE formalism resembles the Q-system formalism (cf. Section 2.7). This is in contrast to e.g. the <C,A>T formalism of Eurotra (cf. Section 2.8).

- **Declarativity:** Declarativity became obvious property of the formalism, as procedural specifications are hard to maintain, when they grow large. Most approaches are nowadays declarative. ALEs do not have procedure calls, unlike PeriPhrase.
- **Non-stratification:** Stratification introduces a hierarchical processing model to a grammar. This was important to avoid in ALE formalism as it was anticipated that stratification would complicate machine learning. An ancestor with this characteristic is PeriPhrase.
- **Relaxation of word orderings:** The idea to mark with angle brackets those constituents that may be in any mutual order saves effort as there is no need to explicate all the possible combinations. This property of the ALE formalism was designed combining ideas from the FUNDPL formalism of Kielikone (Jäppinen & al. 1986, Valkonen & al. 1987) and the GPSG formalism (Gazdar & al.1985).

The augmented lexical entries formalism has some characteristics that cannot be traced from the reviewed or other well-known formalisms:

- **Possibility to combine multiple languages into on single entry:** None of the reviewed MT formalisms supports unlimited number of languages in a single grammar expression. This property of the ALE formalism is motivated by the fact that the entries are related to the conceptual structure of the discourse domain. Also the language specific knowledge can usually be organised in the same way and it becomes natural to specify at the same time for multiple languages the linguistic side of some part of the conceptual structure.
- **Flattening of tree structures:** If regent marks are omitted from ALEs, there is not marked up hierarchic relation but rather a “horisontal” relations. As an outcome there is derived a parse tree which is shallow. This is useful in processing iterative constructs, which in traditional approaches lead into deep trees that need to be flattened for further processing purposes. The author has not observed any other language specification formalism with this property.
- **Suitability for machine learning:** Simple form of machine learning has been employed in translation memory systems, which replace recognised surface level texts with their correspondents picked from a example base. However, automatic learning of generic translation rules is a much more difficult task. None of the reviewed formalisms in Chapter 2 is prepared for machine learning of generic rules. Compared to them ALE formalism is unique.

Augmented Lexical Entries formalism presumes that the texts can be compositionally translated. It does not contain means for handling anaphora or long-distance dependencies. These restrictions have not caused difficulties in applying it to the product description texts. Moreover, controlled languages do not usually accept use of anaphora or long-distance dependencies.

3.6 Experiences from supervised machine learning of ALEs

Automatic learning approaches are central in solving the language specification bottleneck of machine translation. Example-based machine translation (EBMT) strives to learn translation rules from pre-existing translations. EBMT has been much researched since 1980's [Nagao 1984, Simard & al. 1992, Chen 1993, Frederking 1994, Fururose 1994, Jones 1994, McLean 1994, Mahesh & Nirenburg 1995, Sato 1995, Cicekli & Güvenir 1996, Collins 1996, Langlais 1997, Tjong 1996, Tiedemann 1997, Carl & Way 2003]. Applicability for automated grammar learning from sample translations became one very important requirement for the ALE formalism as well. This importance was due to the chosen approach of starting controlled language modelling with an empty model. This approach was taken in order to control the coverage of the model itself and to ensure its consistency. The more usual approach would have been to adapt a pre-existing language model for the particular translation domain. The traditional approach was considered neither to be well controllable nor to guarantee high enough quality. The learning methods are used for two purposes: creating ALE-rules and defining new words for the lexicon. Figure 3.7 shows the overall architecture of the supervised translation grammar learning system.

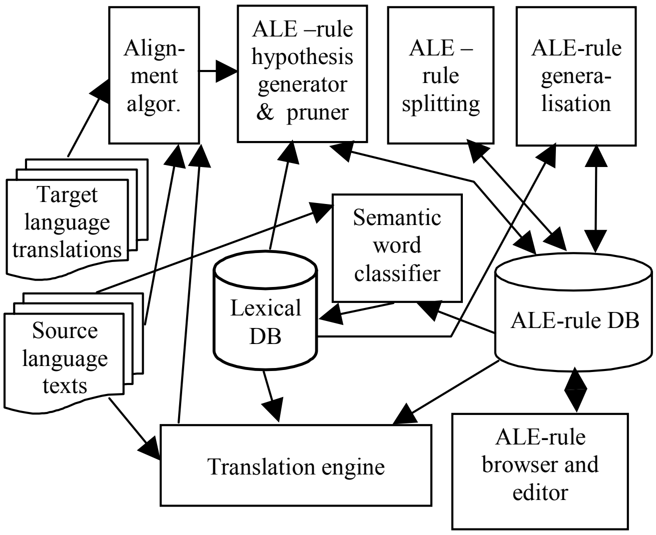
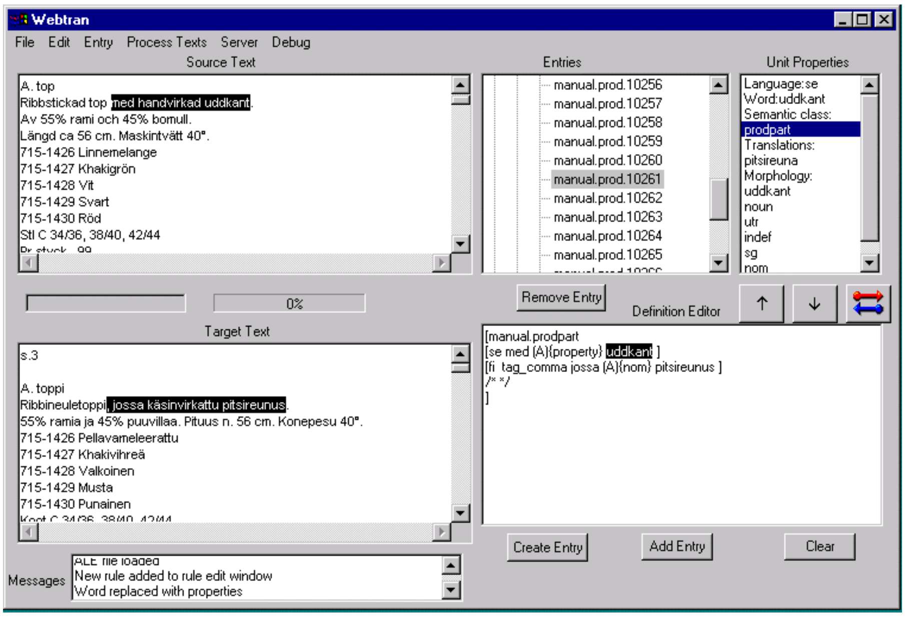


Figure 3.7: Architecture of the translation grammar learning system.

Webtran facilitates discovery of translation grammars by four learning methods. New ALEs are created using the first three methods and the fourth method is for adding new words to lexicon:

1. **Alignment**: The sentence alignment is inspired by the one proposed by Gale and Church [Gale & Chuch 1991]. 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.
2. **Split:** The aligned sentence correspondents are typically very long and too specific to be used as such for CL definition. For this reason, the aligned sentences are split 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.
3. **Generalisation:** The generalisation method extends the usage of ALEs. It generalises the repeating patterns in the language and reduces the number of ALEs needed for language definition. The surface word forms that are generalised are concluded from the lexicon. The generalisation method modifies the rules, so that some words in the rules are replaced by variables with grammatical and semantic feature restrictions and thus the rules can be used in a wider context. 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. This attribute-oriented phrase generalisation is a variant of the method of Han et al. [Han & al. 1993].
4. **Semantic word classifier:** The existing controlled language definition can be applied for inferring semantic information for new words found in text samples. This is carried out by the semantic word classifier method. 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 to every pair of text excerpt and general-entry. 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 approved by the language specifier.



```
A. top
Ribbstickad top med handvirkad uddkant.
Av 55% rami och 45% bomull.
Längd ca 56 cm. Maskintvätt 40°.
715-1426 Linnemelange
715-1427 Khakigrön
715-1428 Vit
715-1429 Svart
715-1430 Röd
Stl C 34/36, 38/40, 42/44
```

```
manual.prod.10256
manual.prod.10257
manual.prod.10258
manual.prod.10259
manual.prod.10260
manual.prod.10261
manual.prod.10262
manual.prod.10263
manual.prod.10264
manual.prod.10265
```

```
Language:se
Word:uddkant
Semantic class:
prodpart
Translations:
pitsireuna
Morphology:
uddkant
noun
utr
indef
sg
nom
```

```
s.3

A. toppi
Ribbineuletoppi, jossa käsinvirkattu pitsireunus.
55% ramia ja 45% puuvillaa. Pituus n. 56 cm. Konepesu 40°.
715-1426 Pellavameleerattu
715-1427 Khakivihreä
715-1428 Valkoinen
715-1429 Musta
715-1430 Punainen
```

```
[manual.prodpart
[se med (A){property} uddkant ]
[fi  tag_comma jossa (A){nom} pitsireunus ]
/* */
]
```

```
ALE file loaded
New rule added to rule edit window
Word replaced with properties
```

Figure 3.8: The user interface of the language modelling tool

The translation engine can be used with an incomplete rule base to reveal those text excerpts that are not covered. The learning process can then be focused on these excerpts. All four learning methods are human assisted, i.e., their results are checked and expanded by a human. The user can verify and edit the rules using the **ALE-rule browser and editor**. The user interface of the language modelling tool is depicted in Figure 3.8.

The supervised machine learning of initial ALE grammars was found feasible in a test with 109 product description articles of Ellos and their sample translations. The achieved sentence alignment accuracy was over 98%, which was a good result because the material was not properly punctuated. Moreover, the rule splitting method was found functional for creating smaller rules. Table 3.7 shows number of the entries splitted, the new entries generated, the truly unique ones, and finally result of the manual evaluation.

NAME	ENTRIES SPLITTED	SPLIT PARTS	UNIQUE PARTS	CORRECT PARTS (%)	INCORRECT SPLIT (%)	FAULTY ORIGINAL (%)
Test Set 1	23	46	33	33 (100)	0	0
Test Set 2	30	60	37	33 (89,2)	2 (5,4)	2 (5,4)
Test Set 3	35	100	54	50 (92,6)	4 (7,4)	0
Test Set 4	61	158	66	60 (90,9)	5 (7,6)	0
Test Set 5	60	151	69	63 (91,3)	6 (8,7)	0

Table 3.7: Results from rule splitting tests.

The semantic word classification method found classification to 261 new words, 90 of which were suggested to the user and 60 were found correct. The original number of found classifications is high because many irrelevant and ambiguous classifications were found. However, the semantic word classification method was found to facilitate lexicon acquisition well.

When the very first set of the material had been processed, there were over 700 generalised and surface form rules. After processing of the second set of the material in a couple of days, the amount of rules decreased rapidly and after a few sets of new material there were a bit over 300 (mostly generalised) rules. Also the time used in the process became shorter after each new set of texts. As the rules become more and more generalised during the process and at the same time also more suitable for the new material, only the actual definition of the new words is needed. This does not take more than a couple of workdays of a professional translator (in a case of product descriptions). The sizes of the sets varied between 265 to 339 sentences both in Swedish and Finnish.

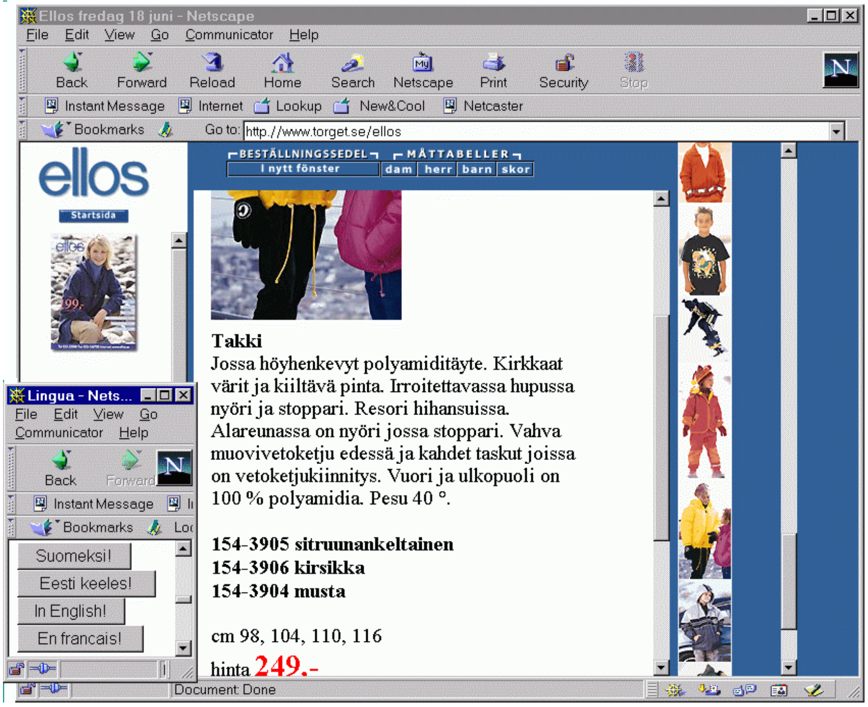
The ALE formalism was found well suited for automatic learning of grammars, as was necessitated in the original requirements. ALE-rule generalisation was tested with the entries from alignment and split methods. Automatically generated generalised entries were used to translate new material that had not been used either in manual processing or in learning phase and translation results were evaluated. The results are summarised in Table 3.8. An automatically learned grammar was able to translate over three out of four sentences in a text excerpt of women's clothes of a totally new catalogue. More details of the learning methods can be found in [Tenni 1999, Tenni et al. 1999].

CLASSIFICATION	**Sentence count**	**%**
Correct	133	76,9
Inflection error(s)	13	7,5
Faulty/partially translated	27	15,6
Total	173	100,0

Table 3.8: Results from translating new material with an automatically generated grammar.

3.7 Experiences from use in catalogue translation

Webtran was originally trialed for providing multilingual views to product descriptions of women's clothes on the WWW. In the trial product descriptions were maintained in one controlled language only (a sublanguage of Swedish). The product descriptions of the original Swedish Ellos' Internet shop were captured for Webtran, which carried out their online translation to the language selected by the user. The trial user interface is depicted in Figure 3.9. The first target language was Finnish and preliminary tests were done with Estonian, French and English as well.



Takki
Jossa höyhenkevyt polyamiditäyte. Kirkkaat värit ja kiiltävä pinta. Irroitettavassa hupussa nyöri ja stoppari. Resori hihansuissa. Alareunassa on nyöri jossa stoppari. Vahva muovivetoketju edessä ja kahdet taskut joissa on vetoketjukiinnitys. Vuori ja ulkopuoli on 100 % polyamidia. Pesu 40 °.

154-3905 sitruunankeltainen
154-3906 kirsikka
154-3904 musta

cm 98, 104, 110, 116

hinta 249.-

Figure 3.9: The WWW interface of the multilingual catalogue test system.

Swedish = pivot 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 3.9: Sample translations of a cloth description from Ellos' Swedish into four other languages.

Table 3.9 shows, how the trial system translated a product description from controlled Swedish into four languages. The prices were not converted as pricing is not a straightforward currency exchange task. It concerns also several other considerations. Appendix C contains a few more examples translated by Webtran.

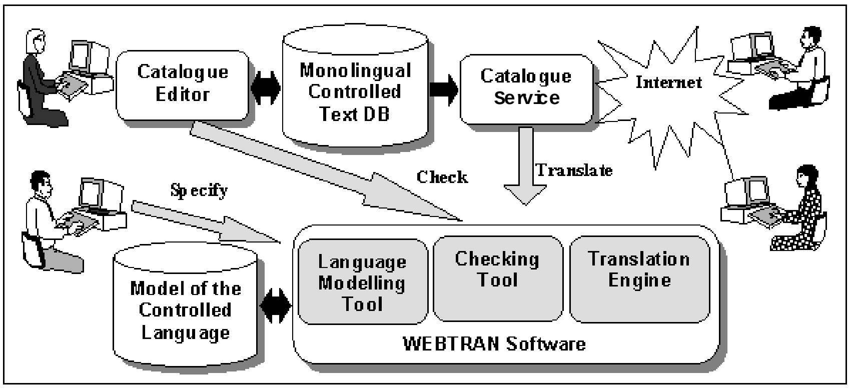


Figure 3.10: The target architecture of the multilingual product catalogue system.

Figure 3.10 describes the architecture that was considered as the goal in the beginning of the Webtran project. In this architecture the product catalogue is maintained in a controlled language in a text database. Specific tools are provided for maintaining the language model, for checking that new product descriptions comply to the language specifications, and for automatic translation. The benefit would be the saving of lots of work, because the product descriptions need to be maintained in just one language. The descriptions could then be translated fully automatically to multiple languages. This kind of architecture proposes a workflow and would very likely require re-engineering of work processes in the adapting organisation. In case of a multinational corporation the re-engineering would involve several suborganisations and the task would get even more complicated.

In practice, the Webtran technology was easier to get into production use at Ellos by adapting it to the company processes. The production use could be started gradually first with just one language pair, and with an option of enlarging the linguistic coverage gradually, when the organisation is ready for it. Figure 3.11 describes the situation at Ellos after one year of production use. Norwegian, German, Russian and Estonian could be potential next target languages.

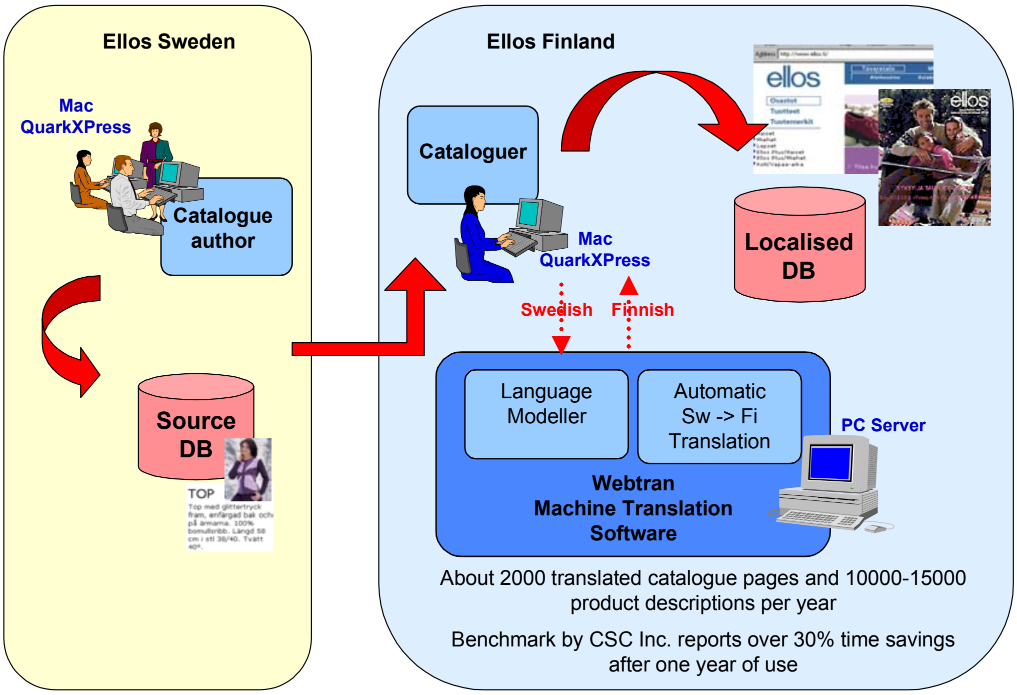

Ellos Sweden
Mac
QuarkXPress
Catalogue author
Source DB
TOP
Ellos Finland
Cataloguer
Mac
QuarkXPress
Swedish
Finnish
Localised DB
ellos
Language Modeller
Automatic Sw -> Fi Translation
Webtran Machine Translation Software
PC Server
About 2000 translated catalogue pages and 10000-15000 product descriptions per year
Benchmark by CSC Inc. reports over 30% time savings after one year of use

Figure 3.11: Embedding of Webtran in the catalogue production process of Ellos since spring 2000.

Experiences from the use of Webtran show that an automatic translation process is faster than phrase-lexicon assisted manual translation, which was the starting situation at Ellos [Jaaranen & al. 2000b]. This requires that an in-company language model is created to control and support the language used within the company. A company benefits from defining of such a model as:

1. source texts will have a uniform and explicit way of expressing information,
2. source texts with systematic use of terminology and linguistic structures can be analysed and translated automatically with minimal post editing, and
3. the company needs to maintain only one pivot version of the source text facilitating a variety of ways for multilingual publishing.

According to the experiences from production use of Webtran, post-editing of automatic translation results only needs a minimal amount of human resources, especially when compared to entire process of manual translation [Jaaranen & al. 2000b]. Depending on the degree of control exerted on the original pivot documents, the post editing may be totally avoided. An in-company language model also constitutes central knowledge property in the company, when authoring and translation tools are taken into everyday production process. The in-company language becomes part of the company culture.

It has been found that the language modelling work for Webtran requires both linguistic and translation expertise as the rules involve both syntax and semantics [Jaaranen 2000a, Jaaranen & al. 2000b]. Definition of terminology and its translation requires also a deep knowledge of the company domain and the sublanguage to be modelled. Creating of a language model from zero is a full-time job even for an expert on linguistics. Depending on the time invested in modelling work and the level of requirements set by the company language, it takes a couple of months to build up a functioning foundation for the language model. It can then be edited and adapted along the implementation of the rest for the Webtran system into production.

Besides the role of the translation engine, Webtran has been a practical tool for terminology management in form of bilingual lexicon and language checking rules that conduct and control the use of different terminological variants [Jaaranen & al. 2000b]. During the first year of production use, terminology refinement took around two to three weeks of work of the main user. After Webtran got fully implemented, it was found sufficient that the main user updates the lexicon and ALEs once per week for an hour. ALEs of the translation grammar require maintenance seldom.

Processing of a 134 page Christmas catalogue takes around one hour after which it is ready for post-editing and proof-reading [Jaaranen & al. 2000b]. Effort required depends on how well controlled the source text has been. The only parts that actually need post-editing are usually longer sentences with other information than restricted facts about the product contents. These “advertising” elements are always reformulated manually to fit with the local culture and style.

The embedding of the translation to the used Quark XPress desktop publishing program saves time of the translator as the layout and typology of the original texts are maintained during the translation process.

The EUROMAP project of EU made an independent survey about the experiences of Webtran system in the production use at Ellos [Loimaranta 2000, WWW-Euromap 2004]. The process of teaching translators how to handle the texts with Webtran took a while. "It is a rather complex system, it takes more than a couple of hours to really understand how Webtran works. The rules database can be demanding, especially when you try to track down why a specific expression is translated incorrectly." said Kirsi Villo-Moen, manager of translation services at Ellos. After using Webtran for over a year and translating several catalogues with it Kirsi Villo-Moen was happy with the results: "The translation time has decreased by over 30 percent. We do not have a complete database of controlled language expressions for the product descriptions. We do have a language description that is already used in the translation process." At the time of the survey the grammar for Swedish→Finnish translation consisted of over 300 ALEs.

As the technology was taken into use only in the Finnish subsidiary of Ellos and the overall document production process was not similar with the architecture of Figure 3.10, the original source texts were not controlled language. This caused additional work, because the texts needed to be checked by a human translator. Further savings would be achieved, if also the mother company in Sweden would start to use Webtran and the controlled language rules would be applied already in Sweden.

The structural similarity of product descriptions in a catalogue can be tiresome and tedious for human translators. Webtran was found to bring welcome relief to this. In addition to speeding up the translation process, Webtran has made the whole translation process more interesting for the translators working at Ellos. Translators can focus on the marketing texts in the catalogues, which are obviously more challenging than the repetitive product descriptions. "With the implementation of Webtran our translators' motivation has increased. On the whole our working environment has improved", said Antti Mälkönen, managing director at Ellos [Loimaranta 2000].

3.8 Summary

Chapter 3 presented a new language definition formalism that has been particularly developed for controlled language translation purposes. This Augmented Lexical Entries (ALE) formalism consists of rule like entries that may be bilingual or multilingual. They can be defined to be non-directed/multidirectional. What those entries actually represent are phrase and sentence structures in terms of partial dependency trees and possibly their translation relations in terms of tree transformations. A grammar, which is sufficient for machine translation between two language, consists of a set of ALEs and a bilingual lexicon of word correspondences. Neither of them should contain ambiguities.

Both surface form entries and generalised entries are available. For facilitating the discovery of entries, human assisted machine learning methods have been developed and tested. Although the learning bears some resemblance with translation memory type of approaches, the generalised entries make clear distinction. In tests, the machine learning approach has been found functioning well and very useful. Formalism itself has been found suitable for automated or machine supported language modelling.

The ALE formalism is intuitive to the extent that it is possible for professional translators to maintain the language definitions themselves. This has been found in practice after ALE based technology been in production use at Ellos.

The ALE formalism provides a uniform way of representing phenomena on different levels of language and it does not force any specific processing model. It is declarative and constraint based. Specifications made with it can be implemented in several ways as functioning programs. VTT’s Webtran machine translation system is one possible implementation. A reference algorithm was presented for using ALEs in machine translation.

The one who implements ALEs may choose a hierarchic processing model in the fashion of Eurotra, if that is found necessary. ALEs can be grouped into clusters using their hierarchical naming feature and activating different clusters in different phases can pipeline processing.

The choice of dependency grammar as the background linguistic theory makes incorporation of semantic information relatively easy. With augmented lexical entries it is possible to obtain near relationship to the corresponding conceptual structures.

4. Conclusions

This thesis presented a formalism for specifying grammars for automatic controlled language translation. The described Augmented Lexical Entries (ALE) formalism was developed in the Webtran project that was carried out at VTT Information Technology in 1997-1999 and funded by Tekes, Ellos Postimyynti Oy, and Tietoenator Oyj. One of the two major results of the project was the controlled language machine translation system Webtran. Its implementation is based on the ALE formalism. The formalism has been found suitable for human assisted machine learning of translation grammars. Moreover, it has been tested and found suitable for translating in the directions Swedish→Finnish, Finnish→English, Finnish→French. Small experiments have also been carried out between Finnish→Estonian, Finnish→Swedish, and Finnish→Norwegian. In Webtran project, the domain of the translated texts was description of clothing products. Later on the translation has been tested successfully also in the domain of vacation cottage descriptions. The Webtran system and the ALE formalism have been in production use at Ellos since spring 2000, with an annual amount of around 2000 translated catalogue pages and 10000-15000 product descriptions. Independent survey found after one year of use that timesavings of more than 30% had been achieved. It is very likely that the system generates even more savings now after its running-in.

The EU project Mkbeem [WWW-Mkbeem 2004] continued in its research the use of ALEs. It extended their application to the meaning extraction, which is the process of relating human language inputs to domain ontologies. In Mkbeem the meaning extraction determines for an input text the so-called Ontological Formula (OF) in terms of the CARIN description logic language [Levy & Rousset 1998]. OFs serve as the interlingual presentation in the Mkbeem system. It is used both in producing a multilingual product database and in solving its cross-lingual queries. The ALE formalism proved to be versatile enough to be adapted to this use as well. More information about these developments can be found in [Leger & al. 2000a, 2000b, Gomez-Perez & al. 2001, Lehtola & al. 2003a, 2003b, 2003c].

4.1 Future research topics

The required language modelling work still forms a bottleneck for adapting the controlled language machine translation technology to new domains and new languages. In order to facilitate the adaptation the Webtran technology would benefit from incorporating handling of probabilities. Naturally that would also change the ALE formalism, as the entries would need to include probabilities. This far the text alignment has been tailored for the particular language pairs. More generic approach would be needed. It could be based on vocabularies and grammars of the languages. The machine learning process could be further automated and equipped with automated knowledge base validation facilities. Self-estimation of translation quality would be a very welcome new feature as well.

References

Adriaens, G. and Macken, L. (1995): "Technological Evaluation of a Controlled Language Application: Precision, Recall and Convergence Tests for SECC", *The Sixth International Conference on Theoretical and Methodological Issues in Machine Translation (TMI-95)*, pp. 123-141.

Almqvist, I. and Sågvall-Hein, A. (1996): "Defining ScaniaSwedish - A Controlled Language for Truck Maintenance", In: *Proceedings of the 1st International Workshop on Controlled Language Applications (CLAW'96)*, Katholieke Universiteit Leuven, Leuven, pp. 159-164.

Arnola, H (1998): “On Parsing Binary Dependency Structures Deterministically in Linear Time”. In: Kahane, S., Polguère, A. (eds.), *Proceedings of Workshop on Dependency-Based Grammars, COLING-ACL’98*, Montreal, pp. 68-77.

Arnold, D. J., Krauwer, S., Rosner, M., des Tombe, L., Varile, G. B. (1986): “The <C, A>, T Framework in Eurotra: A Theoretically Committed Notation for MT”. In: *Proceedings of the 11th International Conference on Computational Linguistics (COLING-86)*, pp. 297-303.

Arnold, D., des Tombe, L. (1987): “Basic Theory and Methodology in EUROTRA”, in Nirenburg, S. (ed), *Machine Translation: Theoretical and Methodological Issues,* Cambridge University Press, pp. 114-135.

Bar-Hillel, Yehoshua (1960), "The Present Status of Automatic Translation of Languages". In: Alt, F.L. (ed.), *Advances in Computers*, vol 1, Academic Press, New York, pp. 91--163.

Barton, G. E. Jr., Berwick, R. C., Ristad, E. S. (1987): “*Computational Complexity and Natural Language”.* MIT Press, Cambridge, 335 p.

Bech, A., Nygaard, A. (1988): “The E-framework: a formalism for natural language processing”. In: *Proceedings of the 12th International Conference on Computational Linguistics (COLING-88)*, pp. 36-39.

Beale, S., Nirenburg, S., Mahesh, K. (1995): “Semantic Analysis in the Mikrokosmos Machine Translation Project”. In: *Proceedings of the 2nd Symposium on Natural Language processing*, Kaset Sart University, Bangkok, Thailand, pp. 294-307.

Beesley, K. R., Hefner, D. (1986): “PeriPhrase: Lingware for Parsing and Structural Transfer”. In: *Proceedings of the 11th International Conference on Computational Linguistics (COLING-86)*, Bonn, Germany, pp. 390-392.

Bennett, W. S., Slocum, J. (1985): “The LRC Machine Translation System”. *Computational Linguistics*, vol. 11, no. 2-3, April-September 1985, pp. 111-119.

Blåberg, O. (1989): "Machine Translation from a Natural Sublanguage to Another". In: *Proceedings of the Scandinavian Conference on Artificial Intelligence - 89, SCAI '89*, IOS, Amsterdam, pp 811-818.

Blåberg, O. (1994): *"The Ment Model – Complex States in Finite State Morphology"*. Ph.D. thesis, Report RUUL 27, Department of linguistics, Uppsala University, 185 p.

Bounsaythip, C., Lehtola, A., Tenni, J. (1998): "Automatic Translation in Cross-Lingual Access to Legislative Databases". *Proceedings of the 8th DELOS Workshop on User Interfaces in Digital Libraries*, Stockholm, Sweden, 21-23 October 1998, pp. 33-38.

Brown, Ralf D. (1996): "Example-Based Machine Translation in the Pangloss System". In: *Proceedings of the 16th International Conference on Computational Linguistics (COLING-96)*, Copenhagen, pp. 169-174. (http://www.cs.cmu.edu/~ralf/papers.html#Brown96)

Carl, M., Way, A. (eds.) (2003): *"Recent Advances in Example-Based Machine Translation"*. Kluwer Academic Publishers, Dordrecht, 482 p.

Caterpillar (1974): *"Dictionary for Caterpillar Fundamental English"*, Caterpillar Corporation, East Peoria, Illinois.

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.

Chomsky, N. (1957): *"Syntactic structures"*. Mouton & Co., Hague, 116 p.

Chomsky, N. (1965): *"Aspects of the Theory of Syntax"*. MIT Press, Cambridge, 261p.

Cicekli, I., and Güvenir, H. A. (1996), "Learning Translation Rules From A Bilingual Corpus", In: *Proceedings of the 2nd International Conference on New Methods in Language Processing (NeMLaP-2)*, Ankara, Turkey, pp. 90-97. (http://www.cs.bilkent.edu.tr/~ilyas/pubs.html)

Collins, D., Cunningham, P., Veale, T. (1996): "An Example-based Approach to Machine Translation". *Proceedings of 2nd Conference of the Association of Machine Translation in the Americas (AMTA-96)*, Montreal, pp. 1-13.

Colmerauer, A., Roussel, P. (1992): *"The Birth of Prolog"*. Report of the Faculté des Sciences de Luminy, Marseille, 28 p. Also in: *ACM SIGPLAN Notices*, vol. 28, no 3, 1993, pp. 37-52.

Covington, M. A. (2001): "A Fundamental Algorithm for Dependency Parsing". In: *Proceedings of 39th Annual ACM Southeast Conference*, Univ. of Georgia, Athens, GA, 18 p.

Crookston, I. (1990): "The E-Framework: Emerging Problems". In: *Proceedings of the 13th International Conference on Computational Linguistics (COLING-90)*, pp. 66-71.

Douglas, S. and Hurs, M. (1996): Controlled Language Support for Perkins Approved Clear English (PACE). In: *Proceedings of the 1st International Workshop on Controlled Language Applications (CLAW'96)*, Katholieke Universiteit Leuven, Leuven, pp 93-105.

Erman, L. D., Hayes-Roth, F., Lesser, V. R., and Reddy, D. R. (1980): "The Hearsay-II Speech-Understanding System: Integrating Knowledge to Resolve Uncertainty". *ACM Computing Surveys*, vol 12, no 2, pp. 213-253.

Faure D., Nédellec C., Rouveirol C. (1998): "*Acquisition of Semantic Knowledge using Machine learning methods: The System ASIUM*". Technical report number ICS-TR-88-16, Laboratoire de Recherche en Informatique, Université Paris Sud, 19 p.

Faure D., Nédellec C. (1999): "Knowledge acquisition of predicate argument structures from technical texts using Machine Learning: the system ASIUM". In: Dieter Fensel, Rudi Studer (editors), *11th European Workshop EKAW'99*, Springer-Verlag, pp. 329-334.

Flickinger, D., Nerbonne, J., Sag, I.A., and Wasow, T. (1987): *Toward Evaluation of NLP Systems.* Technical Report, Hewlett-Packard Laboratories, distributed at the 24th Annual Meeting of the Association for Computational Linguistics (ACL), Stanford, 31 p.

Frederking, R., Nirenburg, S., Farwell, D., Helmreich, S., Hovy, E., Knight, K., Beale, S., Domashnev, C., Attardo, D., Grannes, D., and Brown, R. (1994), "Integrating Translations from Multiple Sources within the Pangloss Mark III Machine Translation System". *Proceedings of 1st Conference of the Association of Machine Translation in the Americas (AMTA-94)*.

Fururose, O., and Iida, H. (1994), "An Example-Based Method for Transfer-Driven Machine Translation". In: *Proceedings of TMI Conference 1994 (International Conference on Theoretical and Methodological Issues in Machine Translation)*.

Gale, W. and Church, K. (1991): “A Program for Aligning Sentences in Bilingual Corpora”, In: *Proceedings of the 29th Annual Meeting Association For Computational Linguistics*, Berkeley, CA, pp.177-184.

Gazdar, G., Pullum, G., and Sag, I. (1985): *“Generalised Phrase Structure Grammar”*. Harvard University Press, 276 p.

Gomez-Perez, A., Corcho Carcia, O., Fernandez Lopez, M., Lehtola, A., Taveter, K., Sorva, J., Käpylä, T., Toumani, F., Soualmia, L., Barboux, C., Castro, E.; Sallatin, J., Arbant, G., Bonnaric, A. (2001): *“Requirements, Choice of a Knowledge Representation and Tools”*. Public Report of MKBEEM project (EC IST-1999-10589), 93 p. (http://www.mkbeem.com/)

Gruber, T. R. (1993): “A Translation Approach to Portable Ontology Specifications”. *Knowledge Acquisition*, Vol. 5, No. 2, pp. 199-220.

Hajicova, E. (1987): “Linguistic meaning as related to syntax and to semantic interpretation”. In: Nagao, M. (ed.), *Language and Articial Intelligence. Proceedings of an International Symposium on Language and Articial Intelligence*, North-Holland Ltd., Amsterdam, pp. 327-351.

Hansen, V. (1994): “PaTrans - a MT system: development and implementation of and experiences from a MT- system”. In: *Proceedings of the First AMTA Conference*, pp. 114-121.

Huijsen W. (1998): “Controlled Language - An Introduction”. In: *Proceedings of the Second International Workshop on Controlled Language Applications (CLAW'98)*, Language Technologies Institute, Carnegie Mellon University, Pittsburg, p. 16-29.

Hutchins, W. J. (1986): “*Machine Translation: Past, Present, Future*”. Ellis Horwood Ltd., Chichester, 487 p.

Hutchins, W. J. and Somers, H. J. (1992): “*An Introduction to Machine Translation*”. Academic Press, 320 p.

Hutchins, W. J. (1995): “Machine Translation: A Brief History”. In: Koerner, E.F.K. and Asher, R.E. (eds.), *Concise history of the language sciences: from the Sumerians to the cognitivists*, Pergamon Press, Oxford, pp. 431-445.

Hyötyniemi H., Lehtola A. (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.

Johnson, R., King, M., Tombe, L. des (1985): “EUROTRA: A Multilingual System under Development”. *Computational Linguistics*, vol. 11, no. 2-3, pp. 155-169.

Jaaranen, K. (1999): “*Webtran Modeller -ohje*” (“Webtran Modeller Guide”, in Finnish). VTT Information Technology, 31 p.

Jaaranen, K (2000a): *“Kontrollerat språk vid maskinöversättning – en fallstudie av översättningsstrategier och subspråket i productbeskrivningarna hos postorderföretaget Ellos Postimyynti Oy*” (in Swedish). Master’s thesis, University of Helsinki, 71 p.

Jaaranen, K., Lehtola, A., Tenni, J., Bounsaythip, C. (2000b): “Webtran tools for in-company language support.” *Language Technologies for Dynamic Business in the Age of the Media*, Vereinigung fur Sprache und Wirtschaft, Köln, pp. 145 - 155.

Jones, D. (1994), "Non-hybrid Example-based Machine Translation Architectures". In: *Proceedings of International Conference on Theoretical and Methodological Issues in Machine Translation 1994 (TMI-94)*, pp. 163-171.

Joscelyne A. (1998): "A controlling interest? Simplified languages to meet the global communication challenge". *LE Journal*, 10 June 1998, 4p. (http://www.hltcentral.org/LeJournal/)

Jämsä, T. (1986): “*Suomen kielen yleisimpien verbien semantiikkaa*” (“Semantics of most common Finnish verbs”, in Finnish). Doctoral dissertation, University of Oulu, 182 p.

Jäppinen, H., Lehtola, A., Nelimarkka, E., Ylilammi, M. (1983): *“Morphological Analysis of Finnish – A Heuristic Approach”*. Research report no. 26, Series B, Digital Systems Laboratory, Helsinki University of Technology, 72 p.

Jäppinen, H., Ylilammi, M. (1986): "Associative Model of Morphological Analysis: An Empirical Inquiry". *Computational Linguistics*, Volume 12, Number 4, pp. 257-272.

Jäppinen, H., Lehtola A., Valkonen K. (1986): “Functional Structures for Parsing Dependency Constraints”. In: *Proceedings of the 11th International Conference on Computational Linguistics (COLING-86)*, Bonn, pp. 461-463.

Jäppinen H., Lassila E. & Lehtola A. (1988a): “Locally Governed Trees and Dependency Parsing”. In: *Proceedings of the 12th International Conference on Computational Linguistics (COLING-88)*, Budapest, pp. 275 - 277.

Jäppinen, H., Honkela, T., Hyötyniemi, H., Lehtola, A. (1988b): “A Multilevel Natural Language Processing Model”. *Nordic Journal of Linguistics*, no 1988:11, pp. 69 - 87.

Kankaanpää T. (1999): *"Design and Implementation of a Conceptual Network and Ontology Editor"*. VTT Information Technology, Research Report TTE1-4-99, Master’s thesis, Helsinki University of Technology, 74 p.

Kay, M. (1980): *“The Proper Place of Men and Machines in Language Translation”*. Report CSL-DD-11, October 1980, Xerox Corporation, 20 p. (one of the historical publications of Xerox PARC http://www.parc.com/about/history/pub-historical.html)

Kikui, G., Hayashi, Y. and Suzaki, S. (1996): “Cross-Lingual Information Retrieval on the WWW”. In: Spyropoulos, C. D. (ed), *Proceedings of MULSAIC 96 workshop, Multilinguality in Software Engineering: AI contribution* (in conjunction with ECAI 96), Budapest, 6 p.

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

Knops, U., Depoortere, B. (1998): “Controlled Language and Machine Translation”, In: *Proceedings of the Second International Workshop on Controlled Language Applications (CLAW'98)*, Language Technologies Institute, Carnegie Mellon University, Pittsburg, p. 42-50.

Koskenniemi, K. (1983): “*Two-level Morphology: A General Computational Model for Word-Form recognition and Production*”. Ph.D. thesis, Publication No. 11, Department of General Linguistics, University of Helsinki, Helsinki, 160 p.

Langlais, P. (1997): *“A System to Align Complex Bilingual Corpora”*. In report TMH-QPSR 4/1997, Kunglika Tekniska Högskolan, Stockholm, 7 p.

Lassila, E. (1989): “Parsing Finnish sentences by performing funcitonally defined sequential subtasks”. In: *Proceedings of the Scandinavian Conference on Artificial Intelligence - 89, SCAI '89*, IOS, Amsterdam, pp 835-840.

Leger, A., Michel, G., Barrett, P., Gitton, S., Gomez-Pere, A., Lehtola, A., Mokkila, K., Rodrigez, S., Sallantin, J., Varvarigou, T., Vinesse, J. (2000a): “Ontology domain modeling support for multi-lingual services in E-Commerce: MKBEEM”. In: *14th European Conference on Artificial Intelligence ECAI'00, Workshop on Applications of Ontologies and Problem Solving Methods*, Berlin, 4 p. (http://delicias.dia.fi.upm.es/WORKSHOP/ECAI00/19.pdf)

Leger, A., Lehtola, A., Villagra, V. (2000b): MKBEEM – Developing Multilingual Knowledge-Based Marketplace. *ERCIM News*, July 2001, pp. 50-52. (Reprinted in Research News of VTT Information Technology, December 2001, pp. 1 – 3.)

Lehrberger, J. (1982): “Automatic Translation and the Concept of Sublanguage”. In: Kittredge, R, Lehrberger, J. (eds.), *Sublanguage: Studies of Language in Restricted Semantic Domains*, de Gruyter, Berlin, pp. 81-106.

Lehtola, A. (1984): “Two-way Finite Automata in Parsing of Finnish Sentence Structures” (in Finnish, *“Kaksisuuntaiset automaatit suomen kielen jäsennyksessä”*). Master’s thesis, Helsinki University of Technology, 120 p.

Lehtola, A., Jäppinen, H., Nelimarkka, E. (1985): “Language-Based Environment for Natural Language Parsing”. *Proceedings of the 2nd EACL Conference*, Geneve, pp 98-106.

Lehtola, A. and Honkela, T. (1988): “AWARE - A Transformation System for Semantic Analysis of Natural Language”. *Finnish Artificial Intelligence Symposium 1988, STeP-88*. Helsinki, 15. - 18.8.1988, Finnish Artificial Intelligence Society and University of Helsinki, pp. 100 - 109.

Lehtola, A., Honkela, T., Hyötyniemi, H., Jäppinen, H. (1988a), "Task Oriented Knowledge Representation Languages for NLP-Systems*". The 3rd Int. Symp. Methodologies for Intelligent Systems, ISMIS '88*, Torino 12 - 15 Oct. 1988., North-Holland. New York, pp. 250 - 259.

Lehtola, A., Honkela, T., Hyötyniemi, H., Jäppinen, H. (1988b): “Knowledge Languages and Metatools for Natural Language Processing”. In *Proceedings of International Computer Science Conference '88. ICSC '88*. Hongkong, 19 - 12 Dec. 1988. Hongkong, IEEE Computer Society, pp. 273 - 280.

Lehtola A., Tenni J., Bounsaythip C. (1998a): “Definition of a Controlled Language Based on Augmented Lexical Entries”. In: *Proceedings of the Second International Workshop on Controlled Language Applications (CLAW'98)*, Language Technologies Institute, Carnegie Mellon University, Pittsburg, pp. 16-29.

Lehtola, A., Bounsaythip, C., and Tenni, J. (1998b): Controlled Language Technology in Multilingual User Interfaces. In: *Proceedings of the 4th ERCIM Workshop on User Interfaces for All (UI4ALL´98)*, Stockholm, 1998, pp. 73-78.

Lehtola, A., Tenni, J., Bounsaythip, C., and Jaaranen, K. (1999a): “Controlled Languages as the Basis for Multilingual Catalogues on the WWW”. In: Jean-Yves Roger, Brian Stanford-Smith and Paul T. Kidd (Eds.), *Business and Work in the Information Society: New Technologies and Applications*, IOS-Press, Amsterdam, pp. 207-213.

Lehtola A., Tenni J., Bounsaythip C., and Jaaranen K. (1999b): “WEBTRAN: A Controlled Language Machine Translation System for Building Multilingual Services on Internet”. In: *Proceedings of Machine Translation Summit VII `99* (MT Summit 99), September 13-17, 1999, Singapore, pp. 487 - 495.

Lehtola, A., Tenni, J., Käpylä, T. (2003a): “Multilingual cataloguing of product information of specific domains: case Mkbeem system”. In: *Proceedings of the joint conference combining the 8th international workshop of the European association for machine translation and the 4th controlled language applications workshop (EAMT-CLAW 03)*, European association for machine translation (EAMT), Dublin, pp. 79 - 86.

Lehtola, A., Heinecke, J., Bounsaythip, C (2003b): “Intelligent Human Language Query Processing in Mkbeem”. In: *Volume 4 of the Proceedings of HCI International 2003: Universal Access in CHI*, Lawrence Erlbaum Associates, Mahwah, pp 750-754.

Lehtola, A., Käpylä, T., Bounsaythip, C., Tallgren, M. (2003c): “Multilingual and ontological product cataloguing tool - user experiences”. In: Paul Cunningham, Miriam Cunningham & Peter Fatelnig (eds.), *Building the Knowledge Economy: Issues, Applications, Case Studies* - Part 2, IOS Press, pp. 947 - 954.

Levy, A., Rousset, M.C. (1998): “Combining Horn Rules and Description Logics in CARIN”. *Artificial Intelligence*, vol. 104, pp. 165-209.

Loimaranta, Outi (2000): “EUROMAP HLT Case Study: Webtran – a controlled language machine translation system for building multilingual services on Internet”. CSC Scientific Computing Ltd, Espoo, 7 p. (http://www.hltcentral.org/usr_docs/ case_studies/euromap/FIN_webtrans.doc)

Maedche, A., Staab, S (2000a): "Discovering Conceptual Relations from Text". In: *Proceedings of ECAI2000*, pp 321-325.

Maedche, A., Staab, S. (2000b): "Mining Ontologies from Text". In: Dieng, R., Corby, O. (eds), *EKAW-2000 - 12th International Conference on Knowledge Engineering and Knowledge Management*. October 2-6, 2000, Juan-les-Pins, France, LNAI, Springer, 14 p.

Mahesh, K., Nirenburg, S. (1995): “Semantic Classification for Practical Natural Language Processing”. In: *Proceedings of the 6th ASIS SIG/CR, Classification Research workshop: An Interdisciplinary Meeting*, Chicago, 8 October 1995, 14 p.

McLean, I., J. (1994), "Example-Based Machine Translation using Connectionist Matching". In: *Proceedings of TMI Conference 1994 (International Conference on Theoretical and Methodological Issues in Machine Translation)*.

Means, L., Godden, K. (1996): “The Controlled Automotive Service Language (CASL) Project”. In: *Proceedings of the 1st International Workshop on Controlled Language Applications (CLAW'96)*, Katholieke Universiteit Leuven, Leuven, p. 106-114.

Melby, A. K., Smith, M. R., Peterson, J. (1980): *“*ITS: Interactive Translation System*”*. In: *Proceedings of the 8th International Conference on Computational Linguistics (COLING-80)*, pp 424-428.

Nagao, M. (1984): "A Framework of Mechanical Translation between Japanese and English by Analogy". In: Elithorn, A. (ed), *Artificial and Human Intelligence*, Elsevier Publishers, pp 173-180.

Nelimarkka E., Jäppinen H. & Lehtola A. (1984): “Two-way Finite Automata and Dependency Theory: A Parsing Method for Inflectional Free Word Order Languages”. In: *Proceedings of the 10th International Conference on Computational Linguistics (COLING-84)*, Stanford, pp. 389-392.

Nirenburg, S. (1987): *“Machine Translation – Theoretical and Methodological Issues”*. Cambridge University Press, Cambridge, 1987.

Odgen, C. K. (1932*): “Basic English, A General Introduction with Rules and Grammar”*. Paul Treber & Co., Ltd. London, UK.

Pereira, C. N., Warren, D. H. D. (1980): “Definite Clause Grammars for Language Analysis – A Survey of the Formalism and a Comparison with Augmented Transition Networks”. *Artificial Intelligence*, Vol. 13, pp. 231-278.

Sato, S. (1995), "MBT2: a method for combining fragments of examples in example-based translation". *Artificial Intelligence*, no 75, Elsevier Science B.V.

Schachtl, S. (1996): “Requirements for Controlled German in Industrial Applications”. In: *Proceedings of the 1st International Workshop on Controlled Language Applications (CLAW'96)*, Katholieke Universiteit Leuven, Leuven, p. 143-149.

Schwitter, R. and Fuchs, N. E. (1996): "Attempto Controlled English – A Seemingly Informal Bridgehead in Formal Territory", In*: Proceedings of poster session of JICSLP'96*, Bonn, Germany, September 1996.

Shieber, S. M. (1986) “*An Introduction to Unification-based Approaches to Grammar*”. CSLI Lecture Notes 4, Stanford University, Stanford, 105 p.

Shieber, S. M. (1992): “*Constraint-Based Grammar Formalisms: Parsing and Type Inference for Natural and Computer Languages*”. The MIT Press, Cambridge, 183 p.

Simard, M., Foster, G. F., Isabelle, P. (1992): "Using Cognates to Align Sentences in Bilingual Corpora", In: *Proceedings of the 4th International Conference on Theoretical and Methodological Issues in Machine Translation*, Montreal/Canada, pp.67-81.

Sharp, R., Streiter, O. (1992): “Simplifying the complexity of machine translation”. In: *Meta Journal*, vol. 37, no. 4, pp. 681-692.

Sharp, R. (1994): *“CAT2 Reference Manual – Version 3.6”.* IAI, Saarbrücken, 117 p.

Slocum, J. (1983): “A Status Report of the LRC Machine Translation System”, In: *Proceedings of the 1st Conference on Applied Natural Language Processing*, ACL, pp. 166-173.

Slocum, J., Bennett, W. S. (1985): “An Evaluation of Metal: The LRC Machine Translation System”, In: *Proceedings of the 2nd Conference of the European Chapter of the Association for Computational Linguistics*, ACL, pp. 62-69.

Steimann, F., Brzoska, C. (1995): “Dependency Unification Grammar for Prolog”. *Computational Linguistics*, vol 21, no 1, pp. 95-102.

Steimann, F. (1998): “Dependency Parsing for Medical Language and Concept Representation”. *Artificial Intelligence in Medicine*, vol 12, no 1, pp. 77-86.

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, 8 p.

Taveter, K., Lehtola, A., Jaaranen, K., Sorva, J., Bounsaythip, C. (1999): “Ontology-based Query Translation for Legislative Information Retrieval”. In: *Proceedings of the 5th ERCIM Workshop on User Interfaces for All (UI4ALL’99)*. GMD Report 74, GMD, Sankt-Augustin, Germany, pp. 47 - 58.

Tenni, J. (1999). *"Methods and a Tool for controlled language definition"*. VTT Information Technology, Research Report TTE1-3-99, Master’s thesis, University of Helsinki, 81 p.

Tenni J., Lehtola A., Bounsaythip C. and Jaaranen K. (1999). "Machine Learning of Language Translation Rules". In: *Proceedings of IEEE SMC'99*, Tokyo, pp. 171 - 177.

Tiedemann, J. (1997): *“Automatical Lexicon Extraction from Aligned Bilingual Corpus”*. Diploma thesis, Otto von Guericke Universität, Magdeburg, Germany, 103 p.

Tjong, Kim Sang, E. (1996): *“Aligning the Scania Corpus”*, Internal report, Department of Linguistics, Uppsala University, Sweden.

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

Varile, G.B., & Lau, P. (1988): “Eurotra: practical experience with a multilingual machine translation system under development”. In: *Proceedings of the Second Conference on Applied Natural Language Processing (ANLP'88)*, pp. 160-167.

Weaver, W. (1955): “Translation”. In: Locke, W.N. and Booth, A.D. (eds.), *Machine Translation of Languages: Fourteen Essays*, MIT Press, Cambridge, pp. 15-23.

Whitelock, P., Kilby, K. (1995): *“Linguistic and Computational Technique in Machine Translation System Design”*, 2nd edition, UCL Press Limited, London, 208 p.

White, J. S. (1987): “The research environment in the METAL project”. In: Nirenburg, S. (Editor), *Machine Translation – Theoretical and methodological issues,* Cambridge University Press, Cambridge, pp 225-246.

Winograd, Terry (1983): *“Language as a Cognitive Process - Volume 1: Syntax”*. Addison-Wesley, 640 p.

Woods, W. A. (1970): “Transition Network Grammars for Natural Language Analysis”. *Communications of the ACM*, Vol. 13, pp. 591-606.

Ørsnes, B., Music, B., Maegaard, B. (1996): “PaTrans - A Patent Translation System”. In: *Proceedings from the 16th International Conference on Computational Linguistics (COLING-96)*, pp. 1115-1118.

Next is the list of referred WWW-sites. The list has been checked in March 2004. Due to the impermanent nature of WWW pages, the later accessibility to the following pages cannot be guaranteed:

WWW-AltaVista 2004, AltaVista translation service, http://babelfish.altavista.com/cgi-bin/translate?

WWW-ACL 2004, ACL Anthology, A Digital Archive of Research Papers in Computational Linguistics, http://acl.ldc.upenn.edu/

WWW-DublinCore 2004, Homepage of the Dublin Core Metadata Initiative http://www.dublincore.org/

WWW-EAMT 2004, European Association for Machine Translation, http://www.eamt.org/

WWW-Euromap 2004, Language engineering success stories page of EUROMAP, http://www.hltcentral.org/page-333.shtml

WWW-GlobalReach 2004, Global Reach Inc., Internet usage statistics, http://www.glreach.com/

WWW-HotBall 2004, Hot Ball Ltd., SYNTAX MT Software homepage, http://www.emgs.net/

WWW-Humanitas 2004, Interface page of Humanitas International organisation to several freely available WWW page translation services, http://www.humanitas-international.org/newstran/more-trans.htm

WWW-Kielikone 2004, Kielikone Ltd. home page, http://www.kielikone.fi/en/

WWW-LingDataCons 2004, Linguistic Data Consortium, University of Pennsylvania, http://www.ldc.upenn.edu/

WWW-Mkbeem 2004, Mkbeem project homepage, http://www.mkbeem.com/

WWW-Systran 2004, SYSTRAN Software Inc., http://www.systransoft.com/

WWW-Scania 2004, Homepage of Scania Swedish project, http://stp.ling.uu.se/~corpora/scania/

Appendix A: Terminology

ALE: See Augmented Lexical Entries.

Anaphora: Back references to other sentences and their components. Typically uses pronouns.

Augmented Lexical Entries: (1) Formalism for describing controlled language definitions. (2) Entries created using this formalism.

Cataphora: Forward references to other sentences and their components.

Compositionality: Translation is compositional when the translation of a complex expression is some (reasonably straightforward) function of the translation of the basic expressions it contains, plus the translation of their mode of combination.

Constituent: A word or a phrase or a clause forming part of a larger grammatical construction. As for dependency grammars a constituent is represented by a tree structure with nodes labelled by words and arcs denoting the linguistic dependency relations between the words.

Content/Service Provider (CP/SP): Company, who offers products or services to customers on Internet, e.g. Ellos, NetAnttila etc.

Context: From a wide point of view language communication: Collection of facts, knowledge and beliefs shared by two interlocutors on several layers. The history of the communication process is a part of the context. From a very narrow point of view of text processing: Context of a word consists of words and sentences in its neighbourhood.

Controlled language (CL): Disambiguated human languages characterised by specific use domain, selected vocabulary and simplified syntax.

Controlled language definition: In Webtran consists of an ALE entrybase and a lexicon.

Dependency grammar: Human language grammar based on specifying word-to-word relationships in a sentence. These relationships may be strictly syntactic or they may also reflect semantic roles of the words.

Dependency tree: Results from analysing a sentence using a dependency grammar are usually presented as a dependency tree, which has as its root the regent word and subordinated to it the found dependents. The tree is recursive in the sense that each subtree presents a dependency tree of its own carrying the information found from the corresponding phrasal constituent. In some grammars the tree may be replaced with a DAG (directed acyclic graph), if the uniqueness principle is abandoned to let a word have more than one regent or if the grammar solves anaphora and ellipsis as structure sharing. This thesis does not consider those generalisations.

Dependent: In a dependency tree a subordinate word of a headword.

Entry: One controlled language definition rule described a lexical entry in ALE-formalism.

Entrybase: All the entries of a controlled language definition.

Example-based machine translation (EBMT): An example-based machine translation system translates by analogy, using past translations to translate other, similar source-language sentences into the target language. The basic premise is that, if a previously translated sentence occurs again, the same translation is likely to be correct.

Feature constraint: In grammars constituents may have features denoting their lexical, syntactic or semantic properties. Usually a feature has a feature type to which it belongs. Feature constraints may be expressed, e.g., by stating conditions of the form *feature_type = feature_value*, like in many unification based formalisms. Examples of such constrains are, e.g., number=singular and case=genetive. In ALE formalism it suffices to name only the feature values, as they are not allowed to be ambiguous in their type.

Formal grammar: A formal grammar is a way to describe a formal language. It consists of of a finite list of symbols and a finite set of rules for forming the expressions in the language.

Formal language: A formal language is a language for which a strict definition is given. This is done with a formal grammar.

Formalisation: Formalisation is the act of creating a formalism, in an attempt to capture the essential features of a real-world or conceptual system in formal language.

Formalism: In this thesis the term formalism refers to a knowledge representation language, which is a formal language developed to be used for some particular knowledge modelling purpose. A formalism is defined using a formal grammar. In minimum a formalism is just a notation. In this thesis we are interested in formalisms that are implementable for computing purposes.

Fully automatic machine translation (FAMT): Machine translation that does not need any human intervention. The machine translation system Taum-Meteo is a good example of FAMT.

Globalisation: Internationalisation + localisation in multiple locales.

Head: In a dependency tree the regent word of a consituent.

Homonymy: One surface level word derived from multiple different words or presenting different inflectional forms of a word.

Human assisted machine learning: A machine learning method that requires human as a guide or evaluator of the results. All the learning methods developed for acquiring ALE grammars require human only as a checker.

Human assisted machine translation (HAMT): Machine translation process during, which is required human involvement to select choices and evaluate intermediate proposals. The system controls the job.

Human language (HL): Language used by human as opposed to formal languages like computer programming languages.

Human language processing (HLP): All language processing tasks that process human language.

Internationalisation (I18N): During internationalisation a system is (re)constructed so that it can be localised with relatively small effort.

Language analysis: Producing a syntactico-semantic representation of a human language text input by using syntactic and semantic knowledge (the opposite direction is called language generation).

Language generation: Generation is the term for producing a string in human language out of a syntactico-semantic representation (the opposite direction is called language analysis).

Lexicon: Includes morphologic, semantic and translation information for words that are present in controlled language definition.

Locale: Area of same language + culture + local practices + business rules. In eCommerce context corresponds tọ market area.

Localisation (L10N): Localisation refers to the activities performed to modify a system to meet the needs specific to different locale(s) than it was originally developed for.

Locution: Self-contained language expression of a proposition. A word, phrase, or expression, especially as used by a particular person, group, etc.

Machine aided human translation (MAHT): Machine translation tool that assists the human translator in her/his job by providing lexicons and other language processing services. The user controls the job.

Machine Translation (MT): Translation of text in a human language into another human language by a set of rules, a lexicon etc by a machine. For introduction, see [Hutchins 1992].

Meaning extraction: Deriving a set of meanings from a sentence or short text.

Natural language (NL): Language used by humans as opposed to the formal languages like computer programming languages etc. Term “human language” (HL) is used instead of natural language throughout this thesis.

Natural language processing (NLP): All language processing tasks that process human language (natural language). In this thesis “human language processing” used instead.

Ontological Formula: A language-independent formula containing the semantic information of a corresponding phrase of human language enriched by information from the domain specific ontology.

Ontology: An ontology is a formal, explicit specification of a shared conceptualisation [Gruber 1993]. The words in the definition can be understood in the following ways:

- conceptualisation: abstract model of some phenomenon in the world which identifies the relevant concepts of that phenomenon
- explicit: concepts used and the constraints on their use are explicitly defined
- formal: machine readable + processable
- shared: consensual knowledge, accepted by a group

Ontology types: Ontology types include:

- Domain ontologies: e.g. electronic, medical, clothing etc. domain
- Metadata ontologies: e.g. Dublin Core initiative for describing content of on-line info sources [WWW-DublinCore 2004]
- Generic or common sense ontologies: CYC, meteorology, colour etc.
- Representational ontologies: e.g. frame ontology
- Other, incl. method and task ontologies: e.g. workflow mgmt definitions

Phrase structure grammar: A phrase structure grammar breaks up a sentence into phrases/constituents, which are then broken into smaller constituents etc. Such phrases can be, e.g., noun phrase, verb phrase, prepositional phrase etc. Specification can be done using production rules. The origins of phrase structure grammars can be tracked to the ancient Stoics. They also make basis for the formal language theory.

Polysemy: Polysemy can be defined as the association of the same expression with more than one (related) concept.

Predicate: In this thesis this term is always corresponds to a semantic predicate. A predicate is language independent.

Projetivity: In a dependency tree no crossing branches are allowed. If word A depends on word B, then all words between A and B are also subordinate to B.

Regent: In a dependency tree the headword of a constituent. Also called shortly a head.

Rule: see entry.

Rulebase: see entrybase.

Semantic class: Attribute-value-pair associated to a word to denote its semantical classification. Words belong to maximum of one class. Semantic classes are not hierarchical in Webtran.

Translation: Process of exchanging an utterance U1 expressed in some language into another utterance U2 expressed in another language, where U1 and U2 are supposed to convey the same meaning. Translation may also refer to conversion of unit values (yards->meters), meaning extraction (converting textual input into ontological formula) etc.

Semiautomatic machine learning: See Human assisted machine learning.

Understanding: As for computers, this includes both meaning extraction and interpretation.

Word: A word is a human language string (e.g. a regular expression: [A-ZÄÖÜÉÈÀÙÓÅ]?[a-zäöüéèàùóåß-]+\.?), which can be mapped to any inflected or base form of the language-dependent lexicon.

Appendix B: Annotated bibliography of the Webtran project

The public documents of the Webtran project have been listed in chronological order:

1. Lehtola, A., Tenni J., and Bounsaythip, C. (1998): “Definition of a Controlled Language Based on Augmented Lexical Entries”. In: *Proceedings of CLAW´98*, CMU Pittsburgh, pp. 16-29.
    - First paper about ALE formalism, which was still under design at the time of writing. Includes comparison to general-purpose machine translation. Introduces the idea of relating the language definitions with product models.
2. Lehtola, A., Bounsaythip, C., and Tenni, J. (1998): “Controlled Language Technology in Multilingual User Interfaces”. In: *Proceedings of the 4th ERCIM Workshop on User Interfaces for All (UI4ALL´98)*, Stockholm, pp. 73-78.
    - The paper was based on ongoing work and outlines the Webtran project.
3. Bounsaythip C., Lehtola, A., and Tenni, J. (1998): “Automatic Translation in Cross-lingual Access to Legislative Databases”. In: *Proceedings of the 8th DELOS Workshop on User Interfaces in Digital Libraries.* Stockholm, pp. 33-37.
    - Based on ongoing work. Discussion of the possible use of controlled languages in cross-language information retrieval.
4. Lehtola, A., Tenni, J., Bounsaythip, C., Jaaranen, K. (1999): “Controlled Languages as the Basis for Multilingual Catalogues on the WWW” In: Jean-Yves Roger, Brian Stanford-Smith and Paul T. Kidd (Eds.). *Business and Work in the Information Society: New Technologies and Applications*. IOS-Press, Amsterdam, pp. 207-213.
    - Paper for business audience about using controlled language processing and the Webtran machine translation system in multilingual catalogue production.
5. Lehtola, A., Tenni, J., Bounsaythip, C. (1999): Webtran Helps Construction of Multilingual Network Services. *Research News*, July 1999, VTT Information Technology, pp. 6-7.
    - An article about the Webtran project in VTT IT’s customer magazine.
6. Taveter, K., Lehtola, A., Jaaranen, K., Sorva, J., Bounsaythip, C. (1999): “Ontology-based Query Translation for Legislative Information Retrieval”. In: *Proceedings of the 5th ERCIM Workshop on User Interfaces for All (UI4ALL’99)*. GMD Report 74, GMD, Sankt-Augustin, Germany, pp. 47 - 58.
    - The paper discusses of the use of domain ontologies in cross-lingual information retrieval. Concept matching is outlined. Moreover, the paper describes the solution developed for the Eulegis consortium in the Webtran project.

7. Tenni, J. (1999): *"Methods and a Tool for controlled language definition"*. VTT Information Technology, Research Report TTE1-3-99, VTT Information Technology, Master's thesis, University of Helsinki, 81 p.
    - The report handles in detail the machine learning methods for discovering ALE based grammars from sample translations.
8. Kankaanpää T. (1999): *"Design and Implementation of a Conceptual Network and Ontology Editor"*. VTT Information Technology, Research Report TTE1-4-99, Master's thesis, Helsinki University of Technology, 74 p.
    - The report handles the ontology editor of the CONE software, which became the cross-lingual IR interface solution of the Webtran project.
9. Tenni, J., Lehtola, A., Bounsaythip, C., Jaaranen, K. (1999): "Machine Learning of Language Translation Rules". In proceedings of: *1999 IEEE Systems, Man and Cybernetics Conference (SMC`99)*, October 12-15, 1999, Tokyo, pp. 171 - 177.
    - The paper describes the machine learning methods used for the automated discovery of initial grammars from corpuses of sample translations. Test results are outlined.
10. Lehtola, A., Tenni, J., Bounsaythip, C., Jaaranen, K. (1999): "WEBTRAN: A Controlled Language Machine Translation System for Building Multilingual Services on Internet". In proceedings of: *Machine Translation Summit VII `99 (MT Summit 99)*, Singapore, pp. 487 - 495.
    - This is an updated description of the Augmented Lexical Entries as considered in the end of the Webtran project. The paper also describes the Webtran machine translations system, which partly implements the formalism.
11. Jaaranen, K (2000): "Kontrollerat språk vid maskinöversättning – en fallstudie av översättningsstrategier och subspråket i productbeskrivningarna hos postorderföretaget Ellos Postimyynti Oy" (In Swedish). Master's thesis, University of Helsinki, 71 p.
    - The report handles the modelling of a grammar for translating product description articles of Ellos' mail-order catalogues from Swedish to Finnish using the ALE formalism.
12. Jaaranen, K., Lehtola, A., Tenni, J., Bounsaythip, C. (2000): Webtran tools for in-company language support. In proceedings of: *Language Technologies for Dynamic Business in the Age of the Media*, Vereinigung für Sprache und Wirtschaft, 23 - 25 Nov. 2000, Köln, pp. 145 – 155.
    - The paper elaborates in-company languages as company specific controlled languages. Experiences from Ellos' product description language and its automatic processing are summarised. The paper was written after the Webtran system had been in production use for almost a year.
13. Lehtola, A. (2004): "Grammar Formalism for Controlled Language Machine Translation: Augmented Lexical Entries". Licentiate's thesis, Helsinki University of Technology, 81 p.

Appendix C: Examples of translations

Clothing product descriptions translated by Webtran from Finnish into French and English

fi	*Ulkoiluasu* *Ulkoilupuku polyamidia, vuori 65% polyesteriä ja 35% puuvillajerseytä. Takissa kaksi vetoketjullista sivutaskua sekä yksi sisätasku. Vetoketju edessä, joustava nyöri ja stopperit helmassa. Housuissa vyötäröllä kuminauha ja nyöri. Vetoketjutaskut. Vetoketju myös lahkeensuissa. Takin pituus 77 cm, lahkeen sisäpituus 81 cm koossa 5. Pesu 40°.* *290-9673 Mariini – valkoinen*
fr	*Survêtement* *Survêtement en polyamide, doublure 65% en polyester et 35% en jersey pur coton. Blouson avec deux poches zippées sur le coté et une poche intérieure. Fermeture à glissière, cordon élastiqué et autobloqueurs à la base. Pantalon avec élastique et cordon à la taille. Poches zippées. Fermeture à glissière aussi aux bas de jambes. Longueur du blouson 77 cm, entrejambe 81 cm en taille 5. Lavage 40°.* *290-9673 Bleu marine – blanc*
en	*Tracksuit* Tracksuit in polyamide, *lining 65% in polyester and 35% in cotton jersey. Jacket with two zipped side pockets and one inner pocket. Zip, elasticated drawcord and toggles in the hem. Trousers with elastic band and drawcord on the waist. Zip pockets. Zip also in the leg openings. Jacket length 77 cm, inside leg 81 cm in size 5. Wash 40°.* *290-9673 Navy blue – white*

fi	*Hiihtohousut* *Lumilukko ja tarranauha lahkeensuissa. Ekstraleveät polvet. Polyamidia, polyamidivuori ja polyesteritoppaus. Pesu 40°.* *253-3161 Tumma kanerva*
fr	*Pantalon de ski* *Guëtre interieur et Velcro aux bas de jambes. Genoux larges. En polyamide, doublure en polyamide et matelassage en polyester. Lavage 40°.* *253-3161 Bruyère foncée*
en	*Ski pants* *Snow cuff and velcro in the leg openings. Broad knee. In polyamide, polyamide lining and polyester wadding. Wash 40°.* *253-3161 Dark heath*

fi	*Takki* *Neljä taskua, kaksisuuntainen vetoketju. Kauluksessa reunus akryylitekoturkista. Pestyä puuvilladenimiä tai vaatepestyä kanvaasia, 65% polyesteriä ja 35% puuvillaa. Polyesterivanutoppaus, polyamidivuori. Pituus 95 cm koossa 38/40. Pesu 30°.* *Koot 34/36, 38/40, 42/44, 46/48.* *139-2892 Beige*
fr	*Blouson* *Quatre poches, fermeture à glissière dans les deux sens. Bord au col. En denim pur coton lavé ou en canvas lavé, 65% en polyester et 35% en coton. Matelassage en ouate polyester, doublure en polyamide. Longueur 95 cm en taille 38/40. Lavage 30°.* *Tailles 34/36, 38/40, 42/44, 46/48. 139-2892 Beige*
en	*Jacket* *Four pockets, two-way zip. Edging in the collar. In washed cotton denim or in garment washed canvas, 65% in polyester and 35% in cotton. Polyester wadding, polyamide lining. Length 95 cm in size 38/40. Wash 30°.* *Sizes 34/36, 38/40, 42/44, 46/48.* *139-2892 Beige*

fi	*Fleeceaamutakki* *Harjattu sisäpinta. Edessä lyhyt vetoketju sävy sävyyn. Kauluksessa nyöri, sivuilla halkio ja taskut. Kirjailu kauluksessa ja rinnassa. Polyesterimikrofleeceä. Pituus 137 cm joka koossa. Pesu 40°.* *119-3463 Liila*
fr	*Robe de chambre en maille polaire* *Surface intérieure douce. Fermeture à glissière courte ton sur ton.* *Cordon au col, fente et poches aux côtés. Broderie au col et à la poitrine. En maille polaire micro en polyester. Longueur 137 cm en chaque taille. Lavage 40°.* *119-3463 Lilas*
en	*Fleece dressing gown* *Brushed inner surface. Short zip tone-on-tone. Drawcord in the collar, slit and pockets on the sides. Embroidery in the collar and in the breast. In polyester micro fleece. Length 137 cm in every size. Wash 40°.* *119-3463 Lilac*

fi	Toppatakki Muhkea malli, olkapäissä vahvikkeet. Painonapeilla kiinnitetty huppu, jossa joustava nyöri. Vetoketjun alla suojalista. Kaksi kannellista taskua. Joustimet hihansuissa. Heijastin. Päällyskangas ja vuori polyamidia. Polyesterivanutoppaus. Pesu 40°. 254-4041 Tumma syklaami
fr	Doudoune Modèle moelleux, épaules à renfort. Capuche attaché par pressions, avec cordon élastiqué. Patte sous la fermeture à glissière. Deux poches à rabat. Élastiques aux bas de manches. Réflecteur. Dessus et doublure en polyamide. Matelassage en ouate polyester. Lavage 40°. 254-4041 Cyclamen foncé
en	Quilted jacket Puffy model, reinforcements on the shoulders. Hood attached with snap buttons, with elasticated drawcord. Storm flap under the zip. Two flap pockets. Elastics in the cuffs. Reflector. Overlay and lining in polyamide. Polyester wadding. Wash 40°. 254-4041 Dark cyclamen

fi	Hame ja toppi Hame ja toppi samaa, joustavaa trikoota. 78% polyesteriä, 17% polyamidia ja 5% Spandexia. Topissa syväänuurrettu selkäosa. Hameessa puolivuori. Kuminauhavyötärö ja halkio takana. Topin pituus noin 58 cm koossa 38/40. Hameen pituus noin 90 cm joka koossa. Konepesu 40°. Koot 34/36 - 50/52. 435-1207 Turkoosi Hinta 199.
fr	La jupe et le top La jupe et le top en tricot similaire et élastiqué 78% polyester, 17% polyamide et 5% Spandex. Top avec dos décolleté profond. Jupe avec demi-doublure. Ceinture élastiquée et fente au dos. Longueur de top env. 58 cm en taille 38/40. Longueur de jupe env. 90 cm en chaque taille. Lavage machine 40°. Tailles 34/36 - 50/52. 435-1207 Turquoise Tarif 199.
en	Skirt and top Skirt and top in same, elasticated tricot 78% polyester, 17% polyamide and 5% Spandex. Top with plunging back. Skirt with half-lining. Elastic waist and split in the back. Top length appr. 58 cm in size 38/40. Skirt length appr. 90 cm in every size. Machine wash 40°. Sizes 34/36 - 50/52. 435-1207 Turquoise Price 199.

Vacation cottage descriptions translated by Webtran from Finnish into French and English

fi	*010-5004 Kittilä ***** *6 Henkeä 85 m2* *Rovaniemeltä 180 km pohjoiseen, Ylläsjärvi 1 km (pituus 2 km, leveys 1 km), matala, hiekoitettu pohja. Hirsirakennus 1990, keittiö, olohuone, makuuhuone 2 vuodetta, pesuhuone, sauna, avokuisti, yläkerrassa 2 makuuhuonetta 2 vuodetta kummassakin ja parveke.* *Varustus: Sähkövalo, sähköliesi, sähköjääkaappi, sähkölämmitys, takka, kylmä ja lämmin vesijohto, suihku, 2 x WC, mikroaaltouuni, TV, radio, astianpesukone, vaatteiden kuivauskaappi, ulkogrilli.* *Etäisyydet (km): Helsinki 1020, rautatie 37, linja-auto 2, ravintola 4, kauppa 2, naapuri 30 m.* *HINTAKAUDET: Viikkohinnat koskien varauksia ajalle 28.4.2002 - 26.4.2003:* *A 660 EUR B 550 EUR C 550 EUR D 523 EUR*
fr	*010-5004 Kittilä ***** *6 personnes 85 m2* *Á 180 km au nord de Rovaniemi, lac Ylläsjärvi 1 km (longueur 2 km, largeur 1 km), fond bas et sable. Chalet en rondins 1990, une cuisine, un séjour, une chambre à coucher avec 2 lits, une salle d'eau, un sauna, une terrasse, 2 chambres à coucher dans l'étage supérieur chacun des deux avec 2 lits et un balcon.* *Équipement: lumière électrique, four, réfrigérateur, chauffage électrique, cheminée, conduite d'eau froide et chaude, douche, 2 x WC, micro-ondes, TV, radio, lave-vaissalle, séchoir pour les vêtements, barbecue.* *Distance (km): Helsinki 1020, train 37, bus 2, restaurant 4, commerce 2, voisin 30 m.* *PÉRIODES* *Tarifs/semaine 28.4.2002 - 26.4.2003:* *A 660 EUR B 550 EUR C 550 EUR D 523 EUR*
en	*010-5004 Kittilä ***** *6 occupants 85 m2* *180 km north of Rovaniemi, lake Ylläsjärvi 1 km (length 2 km, width 1 km), shallow, sand bottom. Log cottage 1990, kitchen, living room, bedroom with 2 beds, washroom, sauna, terrace, 2 bedrooms upstairs both with 2 beds and balcony.* *Equipment : electric light, electric stove, refrigerator, electric heating, fireplace, cold and warm water-supply line, shower, 2 x WC, microwave, TV, radio, dish washing machine, drying cabinet for clothing, barbecue.* *Distances (km): Helsinki 1020, railroad 37, bus 2, restaurant 4, shop 2, neighbour 30 m.* *PERIODS* *Week prices 28.4.2002 - 26.4.2003:* *A 660 EUR B 550 EUR C 550 EUR D 523 EUR*

fi	*030-6252 LOIMAA ****** *9 Henkeä 240 m2* *Helsingistä 140 km luoteeseen, Loimijoki 300 m (leveys 30 m), syvä, pehmeä savipohja, oma vene + kanootti. Kaksikerroksinen mökki 1922, peruskorjattu 1995, alakerta: makuuhuoneessa 2 vuodetta, keittiö, olohuone, pesuhuone, pukuhuone, sauna (sähkökiuas), yläkerta: 2 makuuhuonetta 2 vuodetta kummassakin, makuuhuoneessa yksi vuode, makuuhuoneessa 2 vuodetta, aula.* *Varustus: Sähkövalo, sähköliesi, sähköjääkaappi - pakastin, sähkölämmitys, takka, kylmä ja lämmin vesi, suihku, WC, mikroaaltouuni, TV, radio, videot, astianpesukone, pyykinpesukone, ulkogrilli.* *Etäisyydet (km): Helsinki 140, rautatie 7, linja-auto 7, ravintola 7, kauppa 7, naapuri 70 m.* *HINTAKAUDET* *A 777 EUR B 733 EUR C 611 EUR D 611 EUR*
fr	*030-6252 Loimaa ****** *9 personnes 240 m2* *À 140 km au nord-ouest d'Helsinki, rivière Loimijoki 300 m (largeur 30 m), profond, douce fond d'argile, propre barque + canot. Chalet duplex 1922, rénové 1995, un étage inférieur : une chambre à coucher avec 2 lits, une cuisine, un séjour, une salle d'eau, un dressing, un sauna (électrique), un étage supérieur : 2 chambres à coucher chacun des deux avec 2 lits, une chambre à coucher avec des lit, une chambre à coucher avec 2 lits, une entrée.* *Équipement: lumière électrique, four, réfrigérateur- congélateur, chauffage électrique, cheminée, froid et eau chaude, douche, WC, micro-ondes, TV, radio, magnétoscope, lave-vaisselle, lave-linge, barbecue.* *Distance (km) : Helsinki 140, gare 7, bus 7, restaurant 7, commerce 7, voisin 70 m.* *Périodes* *A 777 EUR B 733 EUR C 611 EUR D 611 EUR*
en	*030-6252 Loimaa ****** *9 occupants 240 m2* *140 km north-west of Helsinki, Loimijoki river 300 m (width 30 m), deep, soft clay bottom, own boat + canoe. Duplex cottage 1922, renovated 1995, downstairs : bedroom with 2 beds, kitchen, living room, washroom, dressing room, sauna (electric), upstairs : 2 bedrooms both with 2 beds, bedroom with one bed, bedroom with 2 beds, hall.* *Equipment: electric light, electric stove, refrigerator- freezer, electric heating, fireplace, cold and warm water, shower, WC, microwave, TV, radio, VCR, dish washing machine, washing machine, barbecue.* *Distances (km) : Helsinki 140, railroad 7, bus 7, restaurant 7, shop 7, neighbour 70 m.* *Periods* *A 777 EUR B 733 EUR C 611 EUR D 611 EUR*

fi	030-7090 ASKAINEN ** 5 Henkeä 70 m2 Turusta 35 km luoteeseen, Saaristomeri - Mynälahti 12 m, matala, pehmeä savipohja, oma vene. Lautarakenteinen mökki 1998, tupakeittiö, makuuhuoneessa 2 vuodetta, makuuhuoneessa 2+1 vuodetta, pesuhuone, sauna, avokuisti. Varustus: Sähkövalo, sähköliesi, sähköjääkaappi - pakastin, sähkölämmitys, takka, mikroaaltouuni, TV, radio, ulkogrilli, juomavesi omistajalta , ulkokäymälä. Etäisyydet (km): Helsinki 190, rautatie 35, linja-auto 3, ravintola 1, kauppa 3, naapuri 30 m. HINTAKAUDET A 425 EUR B 390 EUR C 325 EUR D 260 EUR
fr	030-7090 Askainen ** 5 personnes 70 m2 À 35 km au nord-ouest de Turku, archipel- Mynälahti 12 m, bas, douce fond d'argile, propre barque. Chalet en bois 1998, une cuisine intégrée, une chambre à coucher avec 2 lits, une chambre à coucher 2 + 1 lits, une salle d'eau, un sauna, une terrasse. Équipement: lumière électrique, four, réfrigérateur- congélateur, chauffage électrique, cheminée, micro-ondes, TV, radio, barbecue, eau potable chez propriétaire, latrines. Distance (km): Helsinki 190, gare 35, bus 3, restaurant 1, commerce 3, voisin 30 m. Périodes A 425 EUR B 390 EUR C 325 EUR D 260 EUR
en	030-7090 Askainen ** 5 occupants 70 m2 35 km north-west of Turku, archipelago- Mynälahti 12 m, shallow, soft clay bottom, own boat. Wooden cottage 1998, living room/kitchen, bedroom with 2 beds, bedroom 2 + 1 beds, washroom, sauna, terrace. Equipment: electric light, electric stove, refrigerator- freezer, electric heating, fireplace, microwave, TV, radio, barbecue, drinking water from the owner, outdoor privy. Distances (km): Helsinki 190, railroad 35, bus 3, restaurant 1, shop 3, neighbour 30 m. Periods A 425 EUR B 390 EUR C 325 EUR D 260 EUR