

## **MABLE: a Multi-lingual Authoring Tool for Business Letters**

John Tait and Jeremy Ellman<sup>1</sup>  
School of Computing, Engineering and Technology  
University of Sunderland  
St. Peter's Campus  
Sunderland SR6 ODD  
UK

Diomidis Spinelis  
SENA, S.A. Byzantiou 2  
GR-142 34 Nea Ionia  
Greece

### ***Introduction***

MABLE allows Spanish or Greek business letters writers with limited English to construct good quality, stylistically well formed British English letters. MABLE is a PC program, based on a domain dependant text grammar of fixed and variable phrases that together enforce linguistic cohesion. Interactions with the system are in the user's own language, and the constructed letter may be viewed in that language for sense checking. Our experience to date has shown that the approach it uses to machine aided translation, gives a sufficiently effective and flexible regime for the construction of genuine finished quality documents in the limited domain of business correspondence.

In this paper, we will first review the application domain in which MABLE is intended to operate. Then we will review the approach to machine assisted translation it embodies. Next, we describe the system, its architecture, and the implementation of the linguistic and programming elements. An example interaction sequence is then presented. Finally the successes and shortcomings of the work will be identified and some directions for possible future work will be identified.

### ***The Business Letter Domain***

The production and use of business letters is an important one for machine translation and machine aided translation. It is of course economically important because international trade often requires business correspondence in languages in which the correspondents are not fluent. It is also of academic and intellectual interest since it requires systems that can deal with greater linguistic and stylistic variation than is the norm in, for example, technical manuals without dealing with all the problems of legislation or literature.

---

<sup>1</sup> Contact for correspondence: [Jeremy.Ellman@sunderland.ac.uk](mailto:Jeremy.Ellman@sunderland.ac.uk)

Nowadays the great preponderance of business correspondence is prepared using sophisticated word processors on powerful personal computers. This has opened up an opportunity to easily introduce translation aids into the normal processes of multilingual offices.

MABLE is a Multi-Lingual Business Letter Authoring Tool that provides several functions to support the synergy of its user with a Word Processor. This will help them with the efficient production of high quality documents in a foreign language.

Writing high quality business letters on a PC, whether in one's native language or not, has traditionally been based around accessing a database of phrases, the choice of phrases being left to the user. This resulted in potentially incoherent and stylistically compromised output. Applications also tended to be standalone with output as "text only".

MABLE's primary aim is to help users produce high-quality (in every way) coherent, cohesive output in languages in which they have some knowledge but are not fluent.

### ***Approach to Machine Translation***

The MABLE system avoids many traditional problems of machine translation by interacting with the user in their own language. It also uses linguistic knowledge about the domain of business letters. Its range of guided interaction methods is inspired more by recent work in natural language generation and multilingual user interfaces than by conventional machine translation or translator's workbenches.

The capability to produce good quality coherent text from underlying representations has progressed significantly in recent years. Stone and Doran (1997), Bateman (1999), Bateman et al (1998), for example, have displayed a level of performance in terms of quality and readability not exhibited by earlier systems.

Of more direct relevance MABLE, a stream of work led by Scott at the Information Technology Research Institute at the University of Brighton (e.g. Power and Cavallotto, 1996, Van der Linden and Eugenio, 1996, Power and Scott, 1998, Kibble, 1999, van Deemter and Kibble, 1999). However, this work is characterised by its focus on comparatively small, specific domains, for example software instructions and drug company patient information leaflets. Our approach differs from theirs in that it depends on linguistic, rather than domain or world knowledge. We believe this should make MABLE easier to maintain and to adapt to new domains.

The information MABLE holds about business letters is in a linguistically well motivated form. The representation draws upon familiar work on text grammars by Van Dijk (1977) and lexical cohesion (Halliday and Hasan 1973, 1989). It allows us to propagate information through the partially built letter data structure in a way that maintains consistent stylistic selections, and is naturalistic in terms of its cohesion and coherence.

The user interface also draws upon recent developments in multilingual user interfaces like Lehtola and Honkela (1996) and the further developed "Open Electronic Trading"-project (OET). Here interfaces for workflow management were implemented in three languages (Finnish, Swedish, English) (Lehtola et al, 1998A, 1998B) and the user was able to change the language of the interface dynamically.

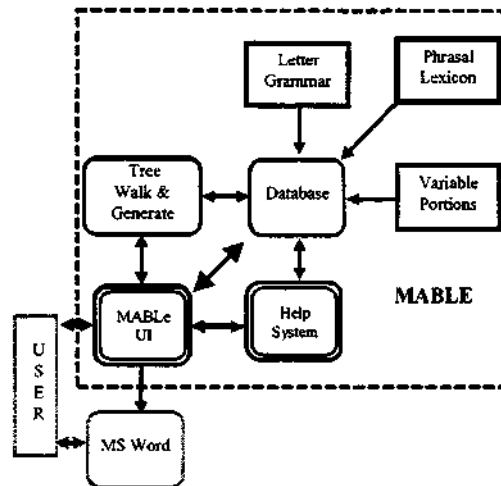
**The MABLE System Architecture**

Like a number of simpler letter-writing aids, MABLE is based around the use of a collection of phrases, but in MABLE these are organised into a "Letter Grammar". This facilitates the production of coherent letters. Furthermore, these phrases may include "slots" that the user fills from a list of choices. The contents of these slots can be chosen in the user's native language with MABLE generating the grammatically correct English equivalent. This method of phrase generation greatly reduces the size of the database while maintaining a high level of flexibility.

MABLE is designed to be closely integrated with a word processor (Microsoft Word). This allows MABLE to be perceived as an integral part of the work place environment, allowing more junior staff to perform the task of creating multilingual business letters.

MABLE is made up of a database that contains the linguistic resources accessed by the MABLE User Interface (UI). Since the MABLE letter grammar is a labelled, directed graph, writing a business letter generally involves "walking" over this graph. A subsequent pass allows "variable portions" of the phrases to be completed.

A schematic view of the architecture of MABLE is shown below.



**Figure 1: The MABLE Architecture**

## System Implementation

The MABLE architecture is distinctive as it places a database centrally. This is used to store static linguistic data and the dynamic data produced as a letter is constructed. The linguistic data is described in more detail in the next section. The database is accessed via the MABLE User Interface (UI). This consists of three submodules: Direct Search; the Composer and the Wizard. The latter two exploit a program that uses the implicit tree structure of the grammatical representation and walks over the uninstantiated data to generate a letter. Direct Search allows immediate use of the phrasal lexicon in particular. We discuss these submodules further below.

A separate output module hands over the generated letter to Microsoft Word. We will not further detail the operation of this module in this paper because although its operation is technically complex the principles are straightforward.

## Linguistic Data

The system uses a context-free phrase structure text grammar of English for business letters, a multilingual phrasal lexicon derived in large part from the Common Procurement Vocabulary, a fragmentary phrasal grammar and associated with these various morphological tables. Apart from some dominance and linear precedence information, all these data structures are held in parallel in the two source languages (Greek and Spanish) and the target language (English). (In fact, English was also implemented as a source language to simplify testing.) All the linguistic information is held in a Microsoft® Access database in eight interrelated tables. Because of this data-driven format, we believe it would be a simple task to extend the system to other source languages, and perhaps to other target languages.

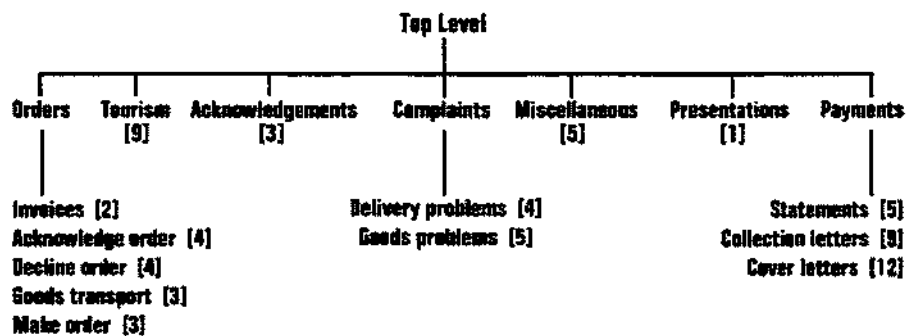


Figure 2: The Database of Resources

MABLE currently covers over 60 kinds of letters and more than 1300 phrasal terms in each of the two source and one target languages. In addition, there is much wider vocabulary in some areas. The letters cover orders, acknowledgements, complaints, payments, and various transactions in tourism and so on. They may be expressed in various levels of politeness, firmness, urgency, and the like. The hierarchy of letter types contained in the grammar is displayed in Figure 2 above.

## User Interface Issues

User Interface design, in language engineering especially, is often thought to be about the superficial ergonomic aspects of button and window layout on the user's screen. Of course, getting this right is essential to a user's basic ability to read and interpret the screen correctly. However, sophisticated products need to consider the nature of the user interaction required, and how this may be provided to the user in a controlled and controllable manner.

MABLE presents some unique opportunities in this area, since as noted previously it is understood that the program is designed for users who know that their knowledge of a foreign language is less than perfect. They do not need reminding of this constantly, rather, they need guidance and support whilst being made to feel it is they that are in control of the program, and not vice versa. At the same time, MABLE needs to fit tidily into the user's work routine, or it will not be accepted. These requirements have led us to design three user interfaces with different philosophies.

The first interface is based on the well-known *Wizard* interaction style found on PCs. Here a user is guided through an interaction, and the program selects the best options based on user input. For MABLE, this entails following a rigid tree walk of the letter grammar to produce grammatical output. Although the user may track both his input and the target language output, there is a rigid feel to the program, with the user possibly feeling out of control. Thus, although this interface is integrated with Microsoft® Word it has not found instant popularity with the users in the second phase of our user requirements work.

Our second interface is known as the *Composer* and is based on representing the grammar as a clickable tree in a standalone application. This has the advantage of the full screen area being exploited with users choosing phrases as they like.

The third interface allows users to identify and select phrases as required, and bypasses the grammatical constraints on coherence. Let us look at these in turn.

### *Wizard*

The MABLE wizard integrates MABLE directly with Microsoft Word 6.0 (Figure 3). Users are guided through the MABLE letter grammar stored in the database using the familiar Microsoft Wizard approach.

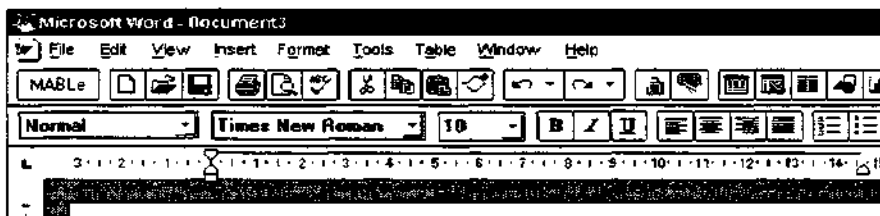


Figure 3: A view of MS Word illustrating the MABLE button on the "Button Bar"

On clicking the MABLE button, the MABLE Wizard will execute.

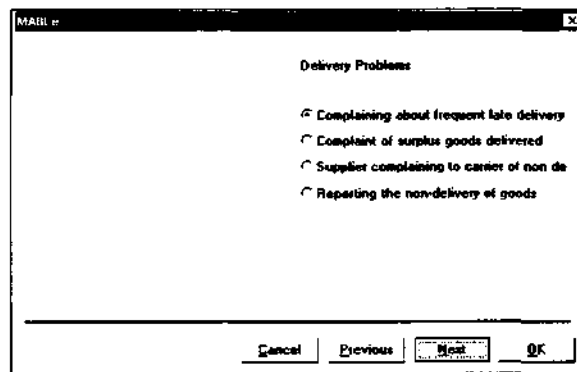


Figure 4: MABLE Wizard screen. (Source language is English)

A Wizard dialogue is separated into three regions by convention. Below the solid black line are the user navigation controls. Here there are buttons for the operations CANCEL, PREVIOUS, NEXT and OK. Above the solid black line, there are conceptually two halves: the left-hand side is used for graphics and help text and the right-hand side for user interaction.

On choosing NEXT users are guided through various windows asking them for their preferred interaction language and introductory screens. Users are then enter the Business Letter Grammar. They can customise the letter as defined by the grammar. Two interaction types are supported: Check Boxes and Radio Buttons. MABLE follows the conventions used throughout MS Windows where Radio Buttons signify an either/or option to the user, while Check Boxes are none/many options. Using this interaction users are guided over the grammar and on completion the text of the letter is exported as a Word document.

### ***Composer***

The aim of the Composer Module is to assist its users in composing new business letters in different target languages. The Letter Grammar is represented in the Composer as a hierarchical tree. This approach is familiar to users as it is used in the Microsoft Windows Explorer. Users interact with the Composer in their own language, to produce a letter in one or more target languages. At any stage of the letter composition users can preview the composed letter in the source language and the target language. In addition, users can search for phrases or constituents specifying various search criteria.

The Composer Module can also be used to maintain the MABLE Database. This is achieved by entering the Advanced User Mode. Working in this mode, an expert user can edit the contents of the Database, change the letter structure, enrich the Database etc.

Letter composition takes place as follows. Users navigate through the tree by selecting different nodes. Terminal nodes contain letter phrases. When these are selected, they are inserted into the letter. An example screen from the Composer module is shown in Figure 5.

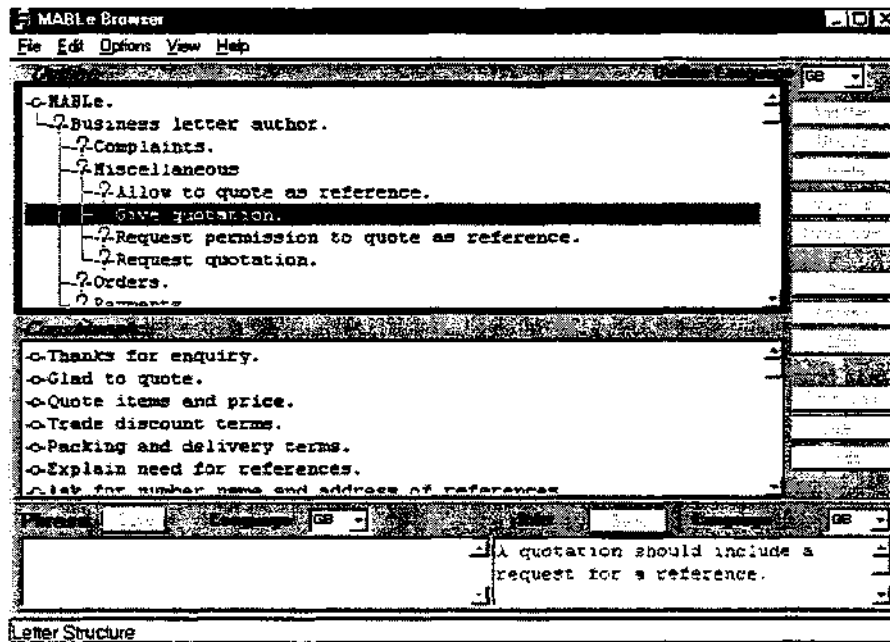


Figure 5: The Composer Module

As the user navigates the grammar, any additional information concerning the highlighted node will be displayed in the Phrase box (see Figure 5). An information box might also be shown that contains advice relating to the particular business phrase.

If MABLE notices any inconsistencies from the selection of a phrase (based on attributes associated with the phrases) then the Composer issues a warning. Users can use the Undo button of the Composer dialogue if they want to revise their choice.

The Composer module was designed to allow the maintenance of the business letter grammar in addition to simple letter composition. This *super-user* mode lets the user both edit the text currently stored in the database and to add new nodes. Thus, new letter types may be added without requiring database skills.

### **Direct Search**

The Direct Search interface allows users to identify and select *individual* phrases from the database. The target language equivalent of that phrase may then be inserted into the letter where it is wanted. Direct search provides a cross lingual information retrieval capability to users who wish to find the target language equivalent of a source language idiom, phrase, or piece of information text. It is a powerful facility, but obviously requires a greater degree of expertise in the target language than the composer or especially the wizard.

Direct search allows users to violate the cohesive integrity of the text grammar as they may include phrases at will. The composer issues a warning if this occurs, but users may go on if they wish. As such, the Direct Search is targeted at users with sufficient knowledge to

manually correct the sense of the letter produced. The interface to the Direct Search is shown in Figure 6 below.

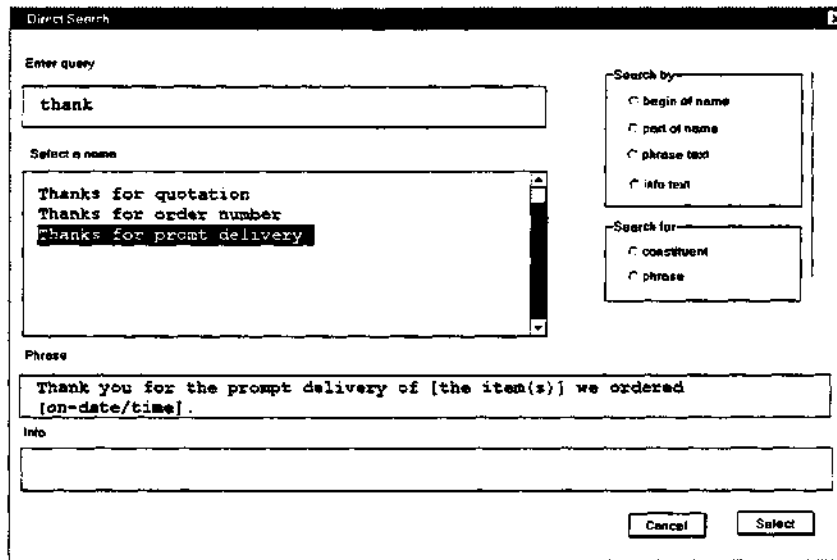


Figure 6: DIRECT SEARCH dialogue

### **Example MABLE Letter**

A typical example of a Greek business letter generated by MABLE (using approximately 20 mouse clicks) is as follows:

Ευχαριστούμε για την παραγγελία σας της "ημερομηνία" για "το είδος/τα είδη" . Εξαιτίας μερικών κατασκευαστικών δυσκολιών με λύπη μας σας πληροφορούμε ότι πρέπει να αναβάλλουμε τη συμφωνημένη ημερομηνία παράδοσης μέχρι "ημερομηνία" . "το/τα είδος/είδη)" έχουν ήδη πουληθεί και ο προμηθευτής μας μας γνωστοποίησε ότι δε θα μπορέσει να μας ξαναπρομηθεύσει στο άμεσο μέλλον. Παρόλα αυτά οι κατασκευαστές υποσχέθηκαν μια επιπλέον προμήθεια "ημερομηνία" , και εάν μπορείτε να περιμένετε μέχρι τότε θα μπορούμε να ικανοποιήσουμε τις απαιτήσεις σας. Ελπίζουμε ότι αυτή η μικρή καθυστέρηση δε θα δημιουργήσει καμία αναστάτωση.

The corresponding English is:

Thank you for your order of "date" for "item(s)" . Because of some manufacturing difficulties we regret to inform you that we shall have to defer the agreed date of delivery until "date" . "item(s)" have already been sold and our supplier has let it be known that he will not be able to supply us further in the foreseeable future. However, the manufacturers have promised a further supply by "date" , and if you could wait until then we would be able to meet your requirements. We hope that this small delay will not cause any inconvenience.



A Spanish version may simply be generated by changing the output language:

Gracias por su pedido del "fecha" para el suministro de "artículo(s)". Lamentamos informarle que tendremos que retrasar la fecha acordada hasta "fecha" debido a problemas del fabricante, "artículo(s)" están vendidos y nuestro proveedor no tiene planes de nuevos suministros en un futuro. Sin embargo, los fabricantes han prometido una nueva entrega sobre "fecha", por lo tanto, si pudiese esperar hasta esta fecha, podríamos satisfacer su solicitud. Esperamos que este pequeño retraso no le cause mayor inconveniente.

As can be seen there is still some work to be done. This may be done manually, by a non-fluent writer who knows the correspondence domain, or by the independent *Variable Phrase Filler* module.

The *Variable Phrase Filler* module replaces the general expressions, such as "date", or "item(s)", with translated forms. It is a template-based system, where closed-class words may be selected from the EU common procurement vocabulary (CPV). This provides baseline translation equivalents of several thousand common European trade goods (e.g. olive oil, oranges, video recorders etc.). Lemmatisation is done with software based on the Plain system from the University of Heidelberg (Hellwig, 1986, 1993).

### **Evaluation**

MABLE was evaluated by representatives of 16 companies (out of 63) who attended a presentation on the project in the Athens Hilton in 1998 as part of a business conference. The companies selected represented various business sectors active in international commerce. Almost all the selected companies are of small or medium size. There were no participants from heavy industry as their needs for business letters are small, since their customer base is limited, and their contracts are complicated and of long duration.

The experiment involved a following a set task using MABLE, and then responding to a general human factors questionnaire. This was composed of general questions such as: *Tasks can be performed in a straight forward manner using this system*", and *"It was difficult to know what to do to achieve what I wanted."* to which subjects could agree or disagree. Such general questions have been shown to a better indicator of software utility than program specific ones.

The results of the users trials were encouraging: All the subjects stated that the system could be used, not only by those who already knew about it and 75% believed that the system was good. On the other hand, 91.7% believed that it was easy for them to do what they wanted once they knew how to do it. Most of the users felt in control of the system (66.7%) but half of them would not like to use the system every day. All believed that the system did not require too many steps to operate, and they understood (83.3%) how to work with it.

### ***Conclusions and Future Work***

Our plans for future work include technical and linguistic strands. From the technical point of view, MABLE requires maintenance to keep pace with changes in software technology. Also, we wish to take this model of machine-aided translation forward to other domains and forms of text. Linguistically we want to improve the cohesion of the letters produced. This requires a careful analysis of the letter contents, and a global analysis of reference and identity chains. It may be the existing VaP module requires a careful reworking from linguistic first principles to provide effective cover for the full range of linguistic phenomena we are likely to encounter.

It became apparent during the project that the close integration with Microsoft products had both advantages and disadvantages. Although maximising the market of available users, future system development and maintenance is critically dependent on Microsoft continuing to support the interfaces used which they are not committed to do. Our view is now that a more open software system and document format should be adopted.

Our work to date has demonstrated the feasibility of constructing a business letter authoring tool based on office technology. Our prototype is on a realistic scale, and although it has a few problems, these are of a technical, rather than fundamental kind. User trials have not suggested any fundamental difficulty, although we plan to undertake a more scientific and detailed evaluation. The approach shows every sign of being a sound basis for further research, development, and application.

### ***Acknowledgements***

The authors would like to thank our numerous colleagues who worked on the MABLE Project, especially Prof. Dr. Peter Hellwig and Heinz-Detlev Koch of the University of Heidelberg, Dr. Chris Smith of MARI, Periklis Tsaheas of SENA, and our other partners STI, PEA and CES.

The MABLE project was supported in part by the EU Framework IV Language Engineering programme under contract LEI203.

Microsoft® Word and Access are registered trademarks of Microsoft Corporation.

## **Bibliography**

- Bateman, J.A., T. Kamps, J. Klein, and K. Reichenberger (1998) "Communicative Goal-driven NL generation and data driven graphics generation: an architectural synthesis for multi-media page generation". Proceedings of the 1998 International Workshop on Natural Language Generation, pp8-17, Niagara-on-the-Lake, Canada.
- Bateman, J.A. (1999) " Using aggregation for selecting content when generating referring expressions" Proceedings of the 37<sup>th</sup> Annual Conference of the Association For Computational Linguistics, pp 127-134, University of Maryland, College Park, Ma.
- Halliday, M.A.K. & Hasan, R.: 1976, "Cohesion in English", Longman, London.
- Halliday, M.A.K. & Hasan, R.: 1989, "Language, context, and text". Oxford University Press, Oxford, UK.
- Hellwig Peter 1986. Dependency Unification Grammar. In Proceedings of the 11th International Conference on Computational Linguistics COLING 86, pages 195-198.
- Hellwig Peter 1993. Extended Dependency Unification Grammar. In Eva Hajicova, editor, Functional Description of Language, pages 67-84. Faculty of Mathematics and Physics, Charles University, Prague.
- Kibble, R. (1999) "Cb or not Cb? Centering theory applied to NLG" Proceedings of the workshop on "The Relation (Eds. D. Cristea, N. Ide, D. Marcu), pp 72-81 at 37<sup>th</sup> Annual Conference of the Association For Computational Linguistics, pp 127-134, University of Maryland, College Park, Ma.
- Lehtola Aarno and Honkela Timo, "Multilinguality in Electronic Commerce - Research Issues", Proceeding of the Workshop on "Multi-linguality in the Software Industry: the AI Contribution", W12 in ECAI-96, ed. D. Spyropoulos, Budapest, pp 21-26, August 1996.
- Lehtola Aarno, Tor Lillqvist, Kari Kekki, and Seija Hakkarainen. Business-to-business Electronic Trading as Co-operative Workflows. In: Jean-Yves Roger, Brian Stanford-Smith and Paul T. Kidd Eds.). Technologies for the Information Society: Developments and Opportunities. IOS-Press, Amsterdam, 1998A, pp. 373-380.
- Lehtola Aarno, Bounsaythip Catherine, and Jarno Tenni. Controlled Language Technology in Multilingual User Interfaces. Proceedings of the 4th ERCIM Workshop on User Interfaces for All (UI4ALL'98), Stockholm, 1998B, pp. 73-78.
- Power R. and N. Cavallotto, "GIST, Multilingual Generation of Administrative Forms", Demonstrations and Posters of the Eighth International Natural Language Generation Workshop, (INLG '96), Herstmonceaux Castle, Sussex, U.K. pp 17-19, June 1996.
- Power R. and D. Scott (1998) "Multilingual authoring using feedback texts" In Proc. COLING/ACL 1998
- Stone M. and Doran C., "Sentence Planning as Description Using Tree Adjoining Grammar", Proceedings of the 35th Annual Meeting of the Association of Computational Linguistics and 8th Conference of the European Chapter of the Association for Computational Linguistics, Madrid, Spain, pp 198-205, July 1997.
- Van Deemter, K., and R. Kibble (1999) "What is coreference and what should coreference annotation be?" Proceedings of the workshop "Coreference and its application"

(Eds. A. Bagga, B. Baldwin, S. Shelton), pp 91-96, at 37<sup>th</sup> Annual Conference of the Association For Computational Linguistics, pp 127-134, University of Maryland, College Park, Ma.

Van der Linden K. and Di Eugenio B., "Learning Micro-Planning Rules for Preventative Expressions" Proceedings of the Eighth International Natural Language Generation Workshop, (INLG '96), Herstmonceaux Castle, Sussex, U.K. pp 11-20, June 1996.

Van Dijk Teun A., 1977 "Text and context. Explorations in the semantics and pragmatics of discourse" London: Longman.