CORECT: Combining CSCW with Natural Language Generation for Collaborative Requirements Capture

John Levine* and Chris Mellish†

Department of Artificial Intelligence,
University of Edinburgh,
80 South Bridge,
Edinburgh EH1 1HN, Scotland, UK.

Abstract

In the CORECT project, we are building a computer-based requirements capture tool for custom-built electronic testing systems. The requirements capture process involves the participation of a wide range of different types of people – the customer, the salesperson, systems engineers, quality assurance, marketing, and so on. Our aim is to build a Computer-Supported Cooperative Working (CSCW) system which will allow these participants to define an Automatic Test System (ATS) collaboratively by adding data and making changes to an evolving design. The collected information about the design will form a large knowledge pool, all of which is pertinent to the design as a whole, but most of which is irrelevant to any particular person engaged in the design process. We will therefore be using natural language generation (NLG) technology to create documents from the central knowledge pool which are tailored to the particular information needs of the participants. These documents will give the users a snapshot of the developing design and will enable them to see how it can be improved and further developed. This paper gives an introduction to the problem we are tackling and how we are trying to solve it, and argues that combining CSCW for input with NLG for output in this way solves some of the problems which are encountered when trying to use either technology on its own.

1 Introduction

In the field of custom-built electronic systems, the requirements definition process from the initial draft specification to the final agreed design is demanding, and requires input from a wide range of skilled personnel. However, due to the lack of a centralised store of knowledge about the developing design, it can also be slow and prone to expensive oversights. This research project, being carried out in collaboration with Racal Research Ltd., Racal Instruments Ltd., Intelligent Applications Ltd., and the University of Sussex, seeks to investigate the automation of requirements capture and the creation of a database of information for system specification and documentation. The system we are developing is a Collaborative Requirements Capture Tool (CORECT) for use by all the participants in the design process, including the customer, the salesperson and the systems engineer.

At the time that this paper is written, we are at the start of what is to be a three-year project, so much of what will be said here concerns our initial ideas about the problem and how we intend to solve it. We will also be presenting our thoughts on how generated documents can be tailored to the individual needs of the various users, and on how we think that Computer-Supported Cooperative Work (CSCW) and natural language generation (NLG) can be usefully combined. Our first prototype for CORECT will be based on the tool for authoring knowledge bases which was developed as part of the IDAS (Intelligent Documentation Advisory System) project (Reiter et al., 1992, 1993). The controlled acquisition of information by this authoring tool will help to ensure that the specification is consistent and (eventually) complete. The tool will also give designers rapid feedback and make requirements information immediately available, helping customers, designers, managers and salespeople to work together by helping them to communicate better.

The role of the University of Edinburgh in this project is the development of a natural language generation component which can automatically derive various kinds of specification documents from the common underlying database. The constraints of document generation will impact on the format and contents of the database as much as the functionality expected of the specifications (e.g. verification and validation). This is an important consideration, because it is not always possible to support NLG from an application program if the needs of NLG are not taken into account as the system itself is designed (Swartout et al., 1991). In CORECT, we will be using NLG technology to create the documents for

*Email: J.Levine@ed.ac.uk

†Email: C.Mellish@ed.ac.uk

the various participants in the design process, such as the customer, the salesperson and the design engineers. Since these users have radically different information needs, as well as different areas of expertise and vocabulary, we will using user modelling techniques to tailor the generated documents to the particular type of user they are intended for.

The problem domain in which CORECT will operate is the collaborative design of an Automatic Test System (ATS). Such devices are designed and manufactured by Racal Instruments in direct response to customer requirements for automated electronic testing of complex equipment. The ATS mainly consists of modular industry-standard computer-controlled instrumentation but each system is different and often complex. In particular, a given system may require the design of a novel piece of equipment to be integrated with the standard modular components. Because a relatively small number of test systems are produced in any given configuration, it is important that the requirements capture process should be swift and effective. In addition, because of the custom-built nature of these products, the cost of the documentation for the machine is a large part of the overall cost, and hence if at least part of the documentation could be generated automatically from the completed requirements specification, this would reduce the overall cost of the ATS.

2 Combining CSCW with NLG

Computer-Supported Cooperative Working (CSCW) systems are designed to enable a group of individuals to collaborate on a piece of collective work, such as the writing of a paper with multiple authors. Many hypertext systems already support asynchronous working between different people; in the Xerox NoteCards system (Irish and Trigg, 1989), multiple authors may open and read the same node, but only one user has the ability to modify the node's content at one time. The Aquanet system (Marshall et al., 1991), under development at Xerox PARC, is a hypertext tool to support collaborative knowledge structuring. In CORECT, we will be developing this idea so that different users will have their own views of the common data, improving communication effectiveness, and building the information at a fact level rather than a document level, from which individual documents can be generated.

Techniques for ensuring that the right information gets delivered to the right people at the right time have been of interest to CSCW since the field's beginnings, with perhaps the best-known project being the MIT Information Lens (Malone et al., 1987). These ideas were further developed in subsequent projects, including Object Lens (Lai and Malone, 1988), the CMU Advisor system (Borensten and Thyberg, 1991) and the GM/EDS In-Vision system (Kass and Stadnyk, 1992). The last of these, which distributes technical documents (engineering change notices) and uses advanced user-modelling techniques as well as production rules to filter the documents, is probably closest to what we are doing in CORECT.

The above-mentioned systems all simply distributed complete messages. In CORECT, however, our intention is to go beyond this by extracting information relevant to a particular user from the common knowledge pool, and then presenting this to the user as a natural language document. Other NLG systems that extract and summarise information have been developed in other research, particularly by CoGenTex; their systems include, for example, FOG (Bourbeau et al., 1990), which produced weather reports; LFS (Iordanskaja et al., 1992), which summarised employment statistics; and Joyce (Rambow and Korelsky, 1992), which summarised software designs from a security perspective. The work on Joyce is particularly interesting because part of its justification was that natural language design summaries are useful to the designers themselves, as well as to people outside the design group. We expect that designers will find summaries even more useful in a multi-author design tool such as CORECT, since they will give them an overview of the progress of the design as a whole, and of what their colleagues have accomplished to date.

The proposed combination of CSCW for collecting and modifying the knowledge pool together with NLG for presenting users with selective views of the data is one which potentially solves problems which are encountered when trying to use either technology individually. Research in CSCW to date provides us with the means to collect data asynchronously from a diverse collection of users and hold that data in a format in which consistency checking (i.e. verification and validation) can be performed. However, for many applications of this technology, such as the collection of requirements information proposed in CORECT, the pool of knowledge soon grows in size such that it is not possible to see all of the information at once. In addition, if the data has been collected and entered by a heterogeneous user group with diverse interests and information needs, then the vast majority of the information in the database will be irrelevant to any particular user. Since the requirements capture process is iterative, in the sense that a user will use a summary of the current design in order to improve and augment it, there is a need for CSCW systems in areas such as ours to be able to present selected information from the data pool for individual users. This role can best be filled using NLG technology to generate documents which are tailored to the needs of the individual user.

The first and probably the most important requirement for natural language generation is that the initial data required for generation, i.e. the domain knowledge, should be available. It is certainly possible to say that we can use NLG technology to generate different documents and texts from the same underlying data, but if the underlying data is not there or is impoverished in some way, then no NLG can take place. In the IDAS project, our goal was the automatic generation of on-line documentation for Automatic Test Systems and other complex custom-built equipment. The knowledge base for the IDAS generator contained enough information about the equipment being documented to support different styles of documentation for the different user tasks and expertise levels. During this project, it was realised that authoring the knowledge base by hand for a complex piece of equipment such as an ATS would be a difficult task, and so a purpose-built graphical authoring tool was developed which would enable systems designers to enter this data more readily. However, by the end of the project, our conclusions were that the benefits gained from the provision of user-tailored documentation were not sufficiently large to outweigh the cost of authoring the large knowledge bases required (Reiter and Mellish, 1993; Reiter et al., 1993).

Given this need for the knowledge required for natural language generation to be collected more cheaply, it makes sense to see whether the data used in other processes, such as the data used during the design of the equipment, could be used for NLG. In CORECT, we are taking this one stage further, by making NLG an integral part of a tool whose primary function is to capture requirements data. Therefore, in this particular application, as far as NLG is concerned, the data comes with no additional cost attached. In addition, the knowledge base constructed during the design process makes a very good starting point for the construction of a knowledge base for a user-oriented system such as IDAS. Although it would be necessary to add information which is not necessary for the design but which is vital for use, maintenance and repair of the machine, the data collected during the requirements capture process would provide a very useful skeleton for the creation of knowledge base for on-line user documentation. Therefore, the use of CSCW for the effective collection of data in CORECT has the potential for solving the authoring problem in natural language generation, at least for applications such as this one.

3 An Overview of CORECT

The basic architecture for the CORECT system is shown in Figure 1. Each of the different types of user interacts with a graphical user interface, which allows the users to add components from a component store to the developing design. Each individual item in the component store is a terminal node of an *is-a* hierarchy, which allows for the use of inheritance when defining the properties of individual components. The structure of the ATS being designed consists of a collection of components which are connected together, where an individual component may be a collection of subcomponents, all of which have to be authored in order to make up a large sub-system of the ATS itself. In essence, the user can pick up components from the parts store and either add them to to a developing parts hierarchy or block diagram showing connections.

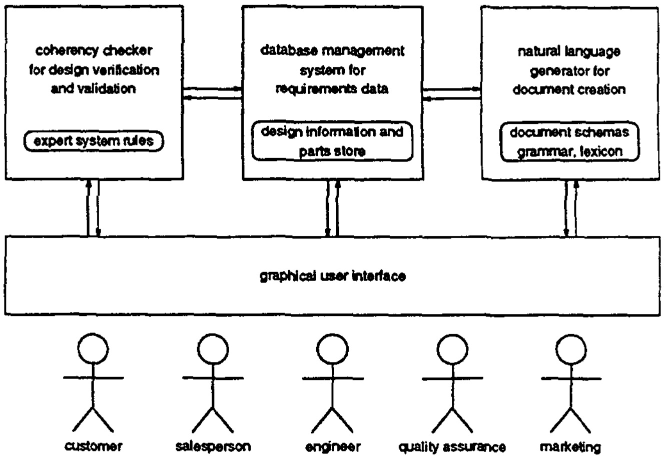
coherency checker for design verification and validation
expert system rules
database management system for requirements data
design information and parts store
natural language generator for document creation
document schemas grammar, lexicon
graphical user interface
customer
salesperson
engineer
quality assurance
marketing

Figure 1: The Architecture of the System

The actual data corresponding to the component store, parts hierarchy and connections is held within the system's database. This is held in a form which is sufficiently detailed for consistency and coherency checking to be performed using expert system rules. The use of a central database of information which is examined and added to by the other three modules of the system is important, since the data pool can be regarded as the core of this system. Using this data-central architecture allows us to develop the system in a strictly modular way with the minimum number of interface specifications. This means that the database manager can be regarded as the minimal system, with the other three modules being extensions to this system which increase its functionality. This also means that if further modules are proposed, these can be added in much the same manner.

The third component of the system is the natural language generator. This will be invoked by the user interface when the user requests that a particular document, such as a costing summary or a proposal, should be generated. The generator will select information from the database which is appropriate to this document, decide on how it should refer to the database concepts for this particular user, and then generate a final surface form for the document together with formatting directives (which could be in SGML or Latex, for example). The finished document will be returned to the user interface which will present it to the user on the screen or send it to be printed. The three phases of generation (content determination, sentence planning and linguistic realisation) will be broadly similar to those used in IDAS (Reiter et al., 1992) and in Joyce (Rambow and Korelsky, 1992).

The primary function of the NLG component in CORECT is to distribute information between the people who are engaged in the design process, allowing them to see different views of the data which are tailored to their particular needs. For example, a customer will be very interested in the overall cost of the machine, and in seeing that the functionality expected of the various components of the machine is met, and so a document prepared for this type of user should contain this sort of information with other more technical material being left out. One of our main aims in designing the CORECT NLG module is to investigate the issues involved in tailoring the content of what is said, and in finding a mechanism which is sufficiently powerful to allow a range of documents to be generated while also stressing that the methods used should be practical and implementable.

The final module of CORECT is the coherency checker, which will perform verification and validation checks on the design. Initially, this will be invoked manually by the user via the user interface, and it will then use expert system rules to see whether there are any gaps in the current design (i.e. components which still need to be added), and whether there are any inconsistencies in the current design, such as the wrong type of connecting cables being used. Considered as a whole, there are three aspects of the CORECT system which solve problems in collaborative requirements capture as it is currently practiced: (a) all the design data is kept in one place; (b) the system can provide different users with different views of this data using NLG; and (c) the system can provide verification and validation of the design, helping to minimise costly oversights.

Acknowledgements

The CORECT project is partially funded by UK SERC grant GR/J/53454 and UK DTI grant IED4/1/7025, and we are grateful to SERC and the DTI for their support of this work. We would also like to thank the CORECT collaborators – Racal Instruments Ltd., Racal Research Ltd., Intelligent Applications Ltd. and the University of Sussex – for the help that they have given us in performing this research. We would also like to express our thanks to Dr. Ehud Reiter for providing many of the original ideas from which this research has developed.

References

Borensten, N. and Thyberg, C. (1991). Power, Ease of Use, and Cooperative Work in a Practical Multimedia Message System. *International Journal of Man-Machine Studies*, 34, 229–260.

Bourbeau, L., Carcagno, D., Goldberg, E., Kittridge, R. and Polguere, A. (1990). Bilingual Generation of Weather Forecasts in an Operations Environment. *Proceedings of the 13th International Conference on Computational Linguistics (COLING-90)*, Volume 1, 90–92.

Iordanskaja, L., Kim, M., Kittridge, R., Lavoie, B. and Polguere, A. (1992). Generation of Extended Bilingual Statistical Reports. *Proceedings of the 14th International Conference on Computational Linguistics (COLING-92)*, Volume 3, 1019–1023.

Irish, P. and Trigg, R. (1989). Supporting Collaboration in Hypermedia: Issues and Experiences. *Journal of the American Society for Information Science*, 40(3), 192–199.

Kass, R. and Stadnyk, I. (1992). Using User Models to Improve Organisational Communication. *Proceedings of the Third International Workshop on User Modelling*, 135–137.

Lai, K. and Malone, T. (1988). Object Lens: A 'Spreadsheet' for Cooperative Work. *Proceedings of the Conference on Computer-Supported Cooperative Work (CSCW '88)*, Portland, Oregon, 115–124.

Malone, T., Grant, K., Turbak, F., Broust, S. and Cohen, M. (1987). Intelligent Information-Sharing Systems. *Communications of the ACM*, 30, 390–402.

Marshall, C., Halasz, F., Rogers, R., Janssen, W. (1991). Aquanet: a Hypertext Tool to Hold Your Knowledge in Place. *Proceedings of the 3rd ACM Conference on Hypertext*, San Antonio, Texas, 261–275.

Rambow, O. and Korelsky, T. (1992). Applied Text Generation. *Proceedings of the Third Conference on Applied Natural Language Processing*, Trento, Italy, 40–47.

Reiter, E. and Dale, R. (1992). A Fast Algorithm for the Generation of Referring Expressions. *Proceedings of the Fourteenth International Conference on Computational Linguistics (COLING-92)*, Volume 1, 232–238.

Reiter, E., Mellish, C. and Levine, J. (1992). Automatic Generation of On-Line Documentation in the IDAS Project. *Proceedings of the Third Conference on Applied Natural Language Processing*, Association for Computational Linguistics, 64–71.

Reiter, E. and Mellish, C. (1993). Optimizing the Costs and Benefits of Natural Language Generation. *Proceedings of the 13th International Joint Conference on Artificial Intelligence (IJCAI-93)*, Chambery, France.

Reiter, E., Mellish, C. and Levine, J. (1993). Automatic Generation of Technical Documentation. Submitted to *Applied Artificial Intelligence.*

Swartout, W., Paris, C. and Moore, J. (1991). Design for Explainable Expert Systems. *IEEE Expert*, June 1991, 58–64.