6th European Conference on
Speech Communication and Technology
(EUROSPEECH'99)
Budapest, Hungary, September 5-9, 1999

CURRENT PRACTICE IN THE DEVELOPMENT AND EVALUATION OF SPOKEN LANGUAGE DIALOGUE SYSTEMS.

[1]Niels Ole Bernsen, [1]Laila Dybkjær and [2]Ulrich Heid

[1]Natural Interactive Systems Laboratory, Odense University, Science Park 10, 5230 Odense M, Denmark

[2]Institut für Maschinelle Sprachverarbeitung, Stuttgart University, Azenbergstraße 12, 70174 Stuttgart, Germany

emails: nob@nis.sdu.dk, laila@nis.sdu.dk, heid@ims.uni-stuttgart.de

ABSTRACT

The growing industrial take-up of spoken language dialogue systems (SLDSs), their constantly increasing sophistication, and the scarcity of teams which master the full system complexity as well as all the necessary steps in the SLDSs life-cycle, has created a felt need for a best practice model for development and evaluation of SLDSs. An obvious first step towards establishing a best practice model is to build a solid overview of current practice. This paper presents a model for the description of SLDS current practice with particular focus on dialogue management and human factors.

1. INTRODUCTION

Conceptually, spoken language dialogue systems (SLDSs) were always complex systems to specify, design, develop, evaluate, and maintain. Their growing industrial take-up, their constantly increasing sophistication, and the scarcity of teams which master the full system complexity as well as all the necessary steps in the SLDSs life-cycle, has created a felt need for a best practice model for the development and evaluation of SLDSs. An obvious first step towards establishing a best practice model is to build a solid overview of current practice. Once this has been done, the descriptive current practice model can be transformed into a draft proposal for a best practice model which can be iteratively refined through testing and peer critique.

The Esprit Long-Term Research Concerted Action DISC [3] is developing a detailed and integrated set of development and evaluation methods and procedures which jointly will constitute a first dialogue engineering best practice model. In addition, DISC is developing a range of support concepts and software tools. During its first year, DISC has produced a comprehensive view of current practice development and evaluation of SLDSs and their components. DISC is now establishing a best practice model incorporating the novel concepts, guidelines and software tools developed in the Action. Focus is on six key aspects of SLDSs, i.e. speech recognition, speech generation, language understanding and generation, dialogue management, human factors, and systems integration.

This paper presents the DISC approach to current practice, how an overview of current practice was created, the concepts of grid and life-cycle and the application of these. Examples are mainly drawn from dialogue management and human factors.

2. THE DISC APPROACH

The DISC current practice approach has been to (a) analyse a broad range of SLDSs and components with respect to the six key aspects mentioned above, and (b) map out their respective development and evaluation processes. In order to adequately capture current practice and overcome various problems primarily relating to the insufficient and incomparable information provided for individual systems and components, a common scheme was developed. The scheme consists of a 'grid' and a life-cycle model both of which are slot-filler structures. The DISC 'grid' enables an in-depth characterisation of the properties of any SLDS or SLDS component. The life-cycle model focuses on the development and evaluation process for SLDSs and their components. The point of departure were the grid and the life-cycle issues presented and discussed in [2]. They were further developed in DISC in an iterative refinement process based on exemplar analyses per aspect. Observations from the exemplars analysed contributed to refinements or to the definition of additional questions. After several iterations, the grid and life-cycle proved reasonably stable for carrying out the 50 exemplar analyses made. A basis had been created for compatible and comparable description of systems and components at each level and across levels.

3. DEVELOPMENT OF THE DISC CURRENT PRACTICE OVERVIEW

The DISC current practice overview was developed as shown in Figure 1 and explained below.

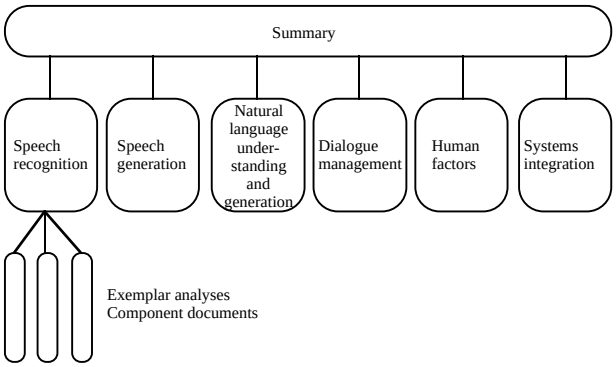

Figure 1. The DISC development of current practice grids and life-cycles.

• For each aspect, a *synthesis working paper* was produced (middle layer of Figure 1). Abstracting from individual observations, it spans the entire range of design and technology choices at hand for the key issues encountered in the exemplars analysed, and shows the range of practical approaches followed in the development and evaluation of systems and components.

• Each synthesis paper is based on several *exemplar case studies*, which serve as detailed background information (bottom layer of Figure 1). Most case studies address both grid and life-cycle issues. As a rule, a case study report contains a brief description of the system or component, answers to the grid questions for the system aspect addressed, answers to the life-cycle questions for that aspect, and some concrete

10.21437/Eurospeech.1999-267

examples, such as (annotated) traces, or sample dialogues, dictionary entries etc.

• Finally, a *reading guide and summary* document for the synthesis papers was created (top layer of Figure 1). The summary also outlines some general trends in the practical working procedures in both development and evaluation which were observed in the synthesis papers.

Although there are many scientific reports about individual SLDSs, providing an overview of current practice in the development and evaluation of SLDSs is not a trivial task.

One reason is that the diversity of systems requires a comprehensive descriptive apparatus to adequately deal with systems ranging from, e.g., call routing and information systems (e.g. of the ATIS type) through to more complex systems designed to handle several types of tasks. The grid and life-cycle were developed for this purpose.

A second cause of difficulty is the lack of documentation in the field. It often proved difficult to collect the information needed to complete the grids and life-cycles. Collaboration with the system developers generated much more information about current practice in SLDS development and evaluation than could be gathered from the literature alone.

All analysed exemplars were provided by the DISC partners. The exemplars that were analysed with respect to one or more aspects were: the French LE Arise system for telephone-accessed train time-table information [1], the CMU Phoenix parser [9], the Daimler-Benz dialogue manager [4], the Daimler-Benz parser [5], the Danish Dialogue System for flight ticket reservation [2], the Vocalis Operetta automated call routing system [6], the Vocalis Voice Activated Dialling system [8], the Verbmobil spoken language dialogue translation system [7], and the multimodal Waxholm tourist boat information system [10].

Each aspect was analysed by at least two different DISC partners. For each aspect at least three significantly different exemplars were investigated. No aspect of a system or component was analysed by a partner who had been involved in its development and evaluation. Every analysis of an aspect of an SLDS or component was verified by the developers of that particular SLDS or component.

Analysis of an aspect of a particular system or component consisted in applying the 'grid' and the life-cycle model to the description of that particular exemplar. Typically, first versions of grid and 'life-cycle were completed on the basis of available papers and reports. This first iteration always generated a - sometimes quite large - number of questions which could not be answered with sufficient certainty, or not at all, based on the initially collected information. Answers were then sought through, i.a., email or telephone interaction with colleagues who had been involved in the development and evaluation of that particular system/component aspect, access to additional data, such as transcriptions and recordings of user-system interactions, and site visits, interviews and demonstrations. In fact, site visits proved necessary to the satisfactory analysis of most DISC exemplars. The final step in the analysis of an aspect of a system or component was to invite verification from that system or component's developers in order to remove any misconceptions from the grid and life-cycle representations.

4. LESSONS LEARNT FROM INFORMATION COLLECTION

Access to detailed information about SLDSs is usually not easy to obtain. The level of granularity of system descriptions in scientific papers varies greatly. Often no full technical account of the solutions adopted is provided or only few examples shown. Moreover, life-cycle questions mainly concern the whys and hows of development practice. These are rarely made explicit in the documentation.

In addition, the design and implementation of SLDSs takes place under various types of constraint both in industry and academia. Projects generally tend to operate under severe time constraints. As a consequence, documentation tends to focus on characteristics which are of particular importance, innovative, or which for some other reason need to be communicated to the outside.

Some documentation is produced for the purpose of project-internal communication. Large projects, such as Verbmobil, typically generate more, and more detailed, documentation than smaller ones in which much project-internal communication is informal. In the latter case, the developers often perfectly remember the reasons for certain choices made, as well as details of the working procedure that was followed, but they never found the time to document those facts.

Furthermore, SLDSs development is an activity with considerable market potential. Consequently, companies and project consortia tend to be reluctant to make available to the outside details on those technical solutions which they consider of key importance to their future development work. This includes internal technical specification and requirements documents, project plans, monitoring documents etc.

An additional point needs to be considered, especially when comparing written sources and the internal information obtained through interviews. For many problems, there are elegant and fully worked out solutions described in the literature. When a system or component has to be designed and implemented under time pressure and other external constraints, however, shortcuts with respect to those ideal solutions are often seen to provide similar results as well as being more efficient and less time-consuming, at least from the point of view of a one-shot action. Changes may happen in both directions. Sometimes, the "ideal" solution is implemented and then proves too slow in processing time or otherwise unhandy. It is then replaced by a more efficient but less elegant or less principled solution, or by one which limits the complexity of, e.g., the treatment of certain linguistic phenomena. Conversely, in several cases developers informed us that, later in the life-cycle of their SLDS, an earlier, fully workable but somewhat ad hoc solution for a sub-component had been replaced by a more principled reconstruction. Such evolutions are rarely documented.

To summarise, much important information about SLDSs and their components is difficult to obtain because it is either confidential or was never documented. The collaboration of developers proved extremely valuable. Much of the value of the DISC exemplar analysis documents is due to the fact that they contain otherwise inaccessible information. Many facts about the technology and the working procedures followed in development and evaluation have never been documented before. Moreover, published papers tend to focus on new, innovative, interesting or otherwise spectacular features rather than on problems encountered, standard solutions used etc.

Another important point follows from the way current practice information was handled in DISC. The negative side to close collaboration between survey authors and developers of surveyed exemplars is that the developers always have more information than the authors. In principle, the system builder is in control of what information to release. However, as we had access to logfiles, traces and resources of the exemplars we analysed, developer control could to a considerable extent be counterbalanced. In no case, moreover, did the modifications brought to a draft case study at the request of developers

change the overall picture. Rather, those modifications mainly concerned technical details.

5. THE GRID AND THE LIFE-CYCLE

The grid questions are aspect-specific. The life-cycle questions, on the other hand, turned out to apply across aspects and, in most cases, irrespective of whether the object of analysis was an SLDS or one of its components. A full life-cycle is presented in Section 6. All its entries are relevant to dialogue management and other components whereas several entries only indirectly relate to human factors issues. This is because human factors is not a component but rather a set of cross-component perspectives. Moreover, it should be noted that evaluation is important to components as well as human factors but that individual evaluation criteria are highly aspect-dependent.

Grid entries tend to be hierarchically structured whereas life-cycle related questions do not. This has to do with the fact that SLDSs and components are typically hierarchically structured in the sense that decision to include one property may exclude other properties and at the same time make yet other properties candidates for inclusion.

The grid slots cover component architecture and function, system architecture and system integration, multimodality and general system performance, but also aspects of individual components, such as speech input and output, and language processing for user and system utterances. Dialogue management is analysed in terms of attentional state, intentional structure and segmentation structure, as well as with respect to interaction history, domain model and user model.

The life-cycle is described in terms of overall design goals and constraints on, and resources of, the development process, such as user and developer preferences, time, money and people. Attention is paid to availability of documentation at all stages, as well as to the way in which the major engineering issues, such as robustness, maintenance and portability, are handled. The human factors aspect is treated somewhat differently from the other aspects in that focus is on tasks and users rather than on technology and system.

5.1 Using the Grid and the Life-Cycle

The grid and the life-cycle have been designed as tools for the following purposes:

1. The structured description and analysis of data collected on aspects of SLDSs and components.

This is their main use so far in DISC.

2. The planning, execution and analysis of SLDS development projects.

The grid and life-cycle are checklists which provide compact information on the full range of currently available technological and procedural options.

As part of the DISC best practice work the grid and the life-cycle now serve as guidelines for monitoring ongoing SDLS development projects. They are being applied to verify that, at each stage in development and evaluation, the full range of technological and procedural choices is available. In parallel, their usability for this purpose is being assessed.

3. The definition of technological options given the actual constraints on the dialogue engineering process.

The use of the grid and life-cycle described in Point 2 above, is somehow static. In all cases, the full range of possibilities for each entry is provided. In the longer term, however, the grid and life-cycle descriptions might be used more dynamically. Choices for a given component at a certain level, or with respect to a certain technology, constrain the set of choices available at a later stage which is technically or logically dependent on those choices. In other words, the questions in the grid and life-cycle are not independent but, rather, organised in partial hierarchies of interdependencies. The grid and life-cycle should be turned into a form which can make these interdependencies clear and offer only the choices still at hand, given a set of prior decisions or external constraints. An interactive flowchart, for instance, would be such a form. The grid and life-cycle could then be used as decision support tools. It should be noted that not only the decisions of SLDS designers, but also external constraints of heterogeneous nature, such as wrt. hardware, noisy environment, development time and cost, have this constraining function.

The grid and life-cycle are thus potentially useful in different contexts and for different purposes. It is part of the DISC best practice work to achieve this potential.

5.2 Development and Evaluation Practice Trends

Overall constraints. As far as overall design goals and constraints are concerned, the need to produce robust systems with real-time (or anytime) behaviour can be seen as an underlying key topic in the design and development of SLDSs.

All analysed systems are intended for novice users, partly also under noisy conditions (e.g. information kiosks).

Documentation of the design and development process. In general, less documentation exists than was expected. Of course, not all company-internal or project-internal documentation is accessible to the outside (e.g. project plans, interface specifications, minutes etc.). However, developers need access to detailed documentation in support of the development and evaluation process. Also, comparison among systems and components requires that detailed and standardised information be available.

Maintenance and portability. Much software development in the SLDSs field is based on rapid prototyping, in most cases with emphasis on modularity and conformity with mainstream programming languages and well-understood models from language and speech processing. In most cases, it is easier to port the core of a system than any of the graphical user interfaces which may come with it.

Evaluation and test information. It seems too early to speak of common practice trends, let alone standards. Procedures for SLDSs and their components are only emerging. On the whole, more evaluation, and in particular guidance of developers with respect to in-house progress evaluation, procedures, measures, and test data, seems to be needed. The field of dialogue management is becoming aware of the need for well-defined evaluation criteria, but there is as yet little agreement with respect to actual procedures and metrics. This also applies to human factors evaluation. Some rather general user acceptance tests tend to be performed by SLDS developers, but there is a serious lack of evaluation criteria and of methods and procedures which would allow comparison of results from tests with different systems.

Overall, more work on evaluation is strongly needed, on comparative performance evaluation of finished products as well as on methods, tools and resources for evaluation during the development of SLDS and components.

The data used in evaluations and the results obtained tend to be strictly internal to the project or company in charge. To help improve evaluation methods for the benefit of all, more openness would seem desirable.

6. THE DISC LIFE-CYCLE

Overall design goal(s): What is the general purpose(s) of the design process?

Hardware constraints: Were there any a priori constraints on the hardware to be used in the design process?

Software constraints: Were there any a priori constraints on the software to be used in the design process?

Customer constraints: Which constraints does the customer (if any) impose on the system/component? Note that customer constraints may overlap with some of the other constraints. In that case, they should only be inserted once.

Other constraints: Were there any other constraints on the design process?

Design ideas: Did the designers have any particular design ideas which they would try to realise in the design process?

Designer preferences: Did the designers impose constraints on the design which were not dictated from elsewhere?

Design process type: What is the nature of the design process?

Development process type: How was the system/component developed?

Requirements and design specification documentation: Is one or both of these specifications documented?

Development process representation: Has the development process itself been explicitly represented in some way? How?

Realism criteria: Will the system/component meet real user needs, will it meet them better, in some sense to be explained (cheaper, more efficiently, faster, other), than known alternatives, is the system/component "just" meant for exploring specific possibilities (explain), other (explain)?

Functionality criteria: Which functionalities should the system/component have (expand overall design goals)?

Usability criteria: What are the aims in terms of usability?

Organisational aspects: Will the system/component have to fit into some organisation or other, how?

Customer(s): Who is the customer for the system/component (if any)?

Users: Who are the intended users of the system/component?

Developers: How many people took significant part in the development? Did that cause any significant problems, such as time delays, loss of information, other (explain)? Characterise each person in terms of novice/intermediate/expert with respect to developing the system/component and in terms of relevant background (e.g., novice phonetician, skilled human factors specialist, intermediate electrical engineer).

Development time: When was the system/component developed? What was the actual development time (estimated in person/months)? Was that more or less than planned? Why?

Requirements and design specification evaluation: Were the requirements and/or design specifications subjected to evaluation prior to system/component implementation? How?

Evaluation criteria: Which quantitative and qualitative performance measures should the system/component satisfy?

Evaluation: At which stages during design and development was the system/component subjected to testing/evaluation? How? Describe the results.

Mastery of the development and evaluation process: Of which parts of the process did the team have sufficient mastery in advance? Of which parts didn't it have such mastery?

Problems during development and evaluation: Were there any major problems? Describe these.

Development and evaluation process sketch: Please summarise in a couple of pages key points of development and evaluation of the system/component. To be done by the developers.

Component selection/design: Describe the system components and their origins.

Robustness: How robust is the system/component? How was this measured? What has been done to ensure robustness?

Maintenance: How easy is the system to maintain, cost estimates etc.

Portability: How easily can the system/component be ported?

Modifications: What is required if the system is to be modified?

Additions, customisation: Has customisation of the system been attempted/carried out (e.g. modification of vocabulary, new domain/task, etc.)? Has there been attempts to add another language? How easy is it (in time/effort) to adapt/customise the system to a new task? Is there a strategy for resource updates (e.g. a predefined sequence of update steps to be performed if a new item is added to the lexicon or if a new grammatical description is added to the grammar)? Is there a tool enforcing that the optimal sequence of update steps is followed (e.g. a menu-driven update interface, etc.)?

Property rights: Describe the property rights situation for the system/component.

Documentation of the design process: E.g. specification documents or parts thereof, architecture diagram (mandatory), user scenario(s), transcribed dialogue(s), other.

References to additional project/system/component documentation: Please refer to this information.

7. CONCLUSION

This paper has provided an overview of the DISC current practice work, focusing on the grid and life-cycle representations of dialogue engineering current practice. The work presented form the basis of ongoing work in DISC to specify and test a methodology for dialogue engineering best practice.

REFERENCES

[1] ARISE: http://www2.echo.lu/langeng/en/le3/arise/arise.html

[2] Bernsen, N. O., Dybkjær, H. and Dybkjær, L.: Designing Interactive Speech Systems. From First Ideas to User Testing. Springer Verlag, 1998.

[3] DISC: http://www.elsnet.org/disc/

[4] Heisterkamp, P. and McGlashan, S.: Units of dialogue management: an example. In Proceedings of ICSLP'96, Philadelphia, 1996, 200-203.

[5] Mecklenburg, K., Hanrieder, G. and Heisterkamp, P.: A Robust parser for continuous spoken language using PROLOG. In Proceedings of Natural Language Understanding and Logic Programming 1995, Lisbon, Portugal, 1995, 127-141.

[6] Operetta: http://www.vocalis.com/products/operetta/infoframe.html

[7] Verbmobil: http://www.dfki.de/verbmobil/

[8] Voice Activated Dialling: http://www.vocalis.com/products/speechtel/infoframe.html

[9] Ward, W. and Issar, S.: The CMU ATIS System. In the proceedings of the ARPA Workshop on Spoken Language Technology, January, 1995, 249-251.

[10] Waxholm: http://www.speech.kth.se/waxholm/waxholm.html