Interactive advice on the use of speech in multimodal system design with SMALTO

Saturnino Luz and Niels Ole Bernsen
Natural Interactive Systems Laboratory, University of Southern Denmark

July 20, 2000

Abstract.

With the recent spread of speech technologies and the increasing availability of application program interfaces for speech synthesis and recognition, system designers are starting to consider whether to add speech functionality to their applications. The questions that ensue are by no means trivial. SMALTO, the tool described below provides advice on the use of speech input and/or output modalities in combination with other modalities in the design of multimodal systems. SMALTO (**S***peech* **M***odality* **A***ux*i**L***iary* **TO***ol*), implements a theory of modalities and incorporates structured data extracted from a corpus of claims about speech functionality found in recent literature on multimodality. The current version of the system aims mainly at supporting decisions at early design stages, as a hypertext system. However, further uses of SMALTO as part of a complete domain-oriented design environment are also envisaged.

Keywords: Multimedia systems, modality combinations, speech functionality, speech recognition, decision support tool.

1. Introduction

Speech is, arguably, the most natural and robust form of human communication. Recent technological advances in speech recognition and synthesis have made it possible for user interface designers to incorporate a significant degree of speech functionality into existing and novel applications. Yet, deciding on whether or not to include speech in an application, and assessing how speech (input and/or output) and other modalities fit together in multimodal systems is not a trivial task. In (Bernsen, 1997b), this task is defined and described as the *speech functionality problem*. The speech functionality problem is the question of what speech is good or bad for, or under which conditions to use, or not to use, speech for information representation and exchange. This paper describes a design support tool called SMALTO which aims at providing system (architecture) designers with advice on questions relating to speech functionality.

Section 2 defines semi-formally the speech functionality problem and introduces the theory of input and output modalities on which SMALTO is based. Section 3 describes a rendering of the SMALTO

knowledge base as a World Wide Web tool, and presents an example of a typical speech functionality *data point*. Section 4 describes the general architecture of the system in terms of its data constraints, along with the main XML *document type definition* used for marking up the speech functionality claims. Section 5 discusses processing and implementation issues. Sections 6 and 7 describe the ways in which we envisage that the system may be incorporated into existing design environments, reports on initial user experiences with the system, and discusses different contexts of use for SMALTO.

2. Background: speech functionality in multimedia systems

The speech functionality problem is described semi-formally in Table I. Expressions in boldface identify *domain variables*, or parameters which help classify and situate the problem. As the speech functionality problem becomes one of increasing practical importance, the research literature is becoming replete with studies of speech functionality. These include aspects of speech in multimodal systems, such as:

- speech and multimedia,
- speech and graphics,
- speech and gesture,
- speech in auditory interfaces,
- speech, pen and graphics,
- email versus voice mail, etc.

However, it seems extremely unlikely that empirical studies alone will suffice in telling system developers what they need to know in a timely fashion in order to avoid user dissatisfaction or poor system performance due to erroneous choices of modality combinations. SMALTO addresses this issue by providing a database of case studies organised within a theoretical framework which tries to identify the basic properties of different modalities.

The theory of speech modalities implemented by SMALTO (Bernsen and Dybkjær, 1998) is based on the assumption that it would be useful for developers to be able to rely largely on comprehensible theoretical guidance instead of lengthy experimentation. The theory derives from *modality theory* (Bernsen, 1997a), whose purpose is to describe the objective properties of all unimodal modalities in acoustics, graphics

Table I. The complexity of the speech functionality problem.

[combined speech input/output, speech output, or speech input modalities M1, M2 and/or M3 etc.] or [speech modality M1, M2 and/or M3 etc. in combination with non-speech modalities NSM1, NSM2 and/or NSM3 etc.] are [useful or not useful] for [**generic task** GT and/or **speech act type** SA and/or **user group** UG and/or **interaction mode** IM and/or **work environment** WE and/or **generic system** GS and/or **performance parameter** PP and/or **learning parameter** LP and/or **cognitive property** CP] and/or [preferable or non-preferable] to [alternative modalities AM1, AM2 and/or AM3 etc.] and/or [useful on conditions] C1, C2 and/or C3 etc.

and haptics. The observation that all speech functionality claims refer to modality properties gave rise to the idea of testing the explanatory power of modality theory on a small but well-defined fragment within the scope of the theory, i.e. a set of claims about speech functionality.

A set of over 120 claims about speech functionality has been systematically gathered from papers dedicated to the issue (Baber and Noyes, J. (Eds.), 1993), and it has been shown that 18 modality properties suffice to justify, support or correct 97% of the 109 claims that were not flawed in one way or another (Bernsen, 1997b). A more recent, larger, control study has confirmed this result (Bernsen and Dybkjær, 1999a). The properties were classified according to the modalities they characterise: *linguistic input/output*, *arbitrary input/output*, *acoustic input/output*, *acoustic output*, *static graphics*, *dynamic output*, *dynamic acoustic output*, *speech input/output*, *speech output*, *synthetic speech output*, *non-spontaneous speech input*, *discourse output*, *discourse input/output*, *spontaneous spoken labels/keywords and discourse input/output*, and *notational input/output*.

These 18 modality properties include all the properties that modality theory could contribute to the data analysis. Examples of modality properties are statements such as the following:

MP4 "Acoustic input/output modalities are omnidirectional".

MP5 "Acoustic input/output modalities do not require limb (including haptic) or visual activity".

It has been possible to categorise all claims as belonging to one in 13 claim types such as: claims *recommending* combined speech input or output, claims *positively comparing* combined speech input and output to other modalities, etc. Eleven of these 13 types were represented in

the data. The following is an example of a speech functionality claim (in its original form, prior to semi-formalisation and evaluation):

> "One limitation of auditory interfaces is the difficulty in presenting an overview of the interface contents." (Mynatt, 1997).

Claims, their semi-formal representation, their classification according to the parameters shown in Table I, and their evaluation against modality properties comprise the core of SMALTO's data. The system described in this paper essentially indexes the data analysed in the study reported in (Bernsen and Dybkjær, 1999a) as a knowledge base, providing user-friendly means for retrieving this knowledge[1]. The data that forms the basis for (Bernsen, 1997b) will also undergo XML annotation (along the lines described below) and should be added to the SMALTO knowledge base in the near future.

3. Interactive advice on speech functionality

At one level, one can think of SMALTO as a tutorial introduction to aspects of speech functionality in human-computer interaction, from the point of view of modality theory. It can also be seen as a reference hypertext on interface design issues involving speech modalities. However, a broader context of use was taken into account in the design of the system in order to enhance our understanding of its intended user profiles and provide for further development and integration with existing frameworks. We will start by describing the current incarnation of SMALTO as a stand-alone system on the World Wide Web.

The system is implemented as a dynamic hypertext whose structure is depicted in Figure 1. The user-tracking phase will always take place transparently, though it has been conceptualized as the first node visited by any user. In the likely case of a user trying to "bookmark" a page (static or CGI-generated) and returning to it later, the system will intercept the request and attempt automatic identification before sending the user to the requested page or to the last page visited according to the user's interaction log, if different from the former.

The main entry points to the system lead to the following paths: a *tutorial* introduction to SMALTO and the speech functionality problem, advice on the use of speech in specific *applications*, for specific *tasks*, and more detailed advice on interfaces involving the use of *speech input*, *speech output* and *speech input and output* when combined with

[1] According to the Webster dictionary a "smalto" is a "colored glass or enamel or a piece of either used in mosaic work". We like to think that the data (claims) in the theory bear some analogy with smalti in a mosaic, the former being like small pieces, neatly organised in the bigger and complex picture of speech functionality.

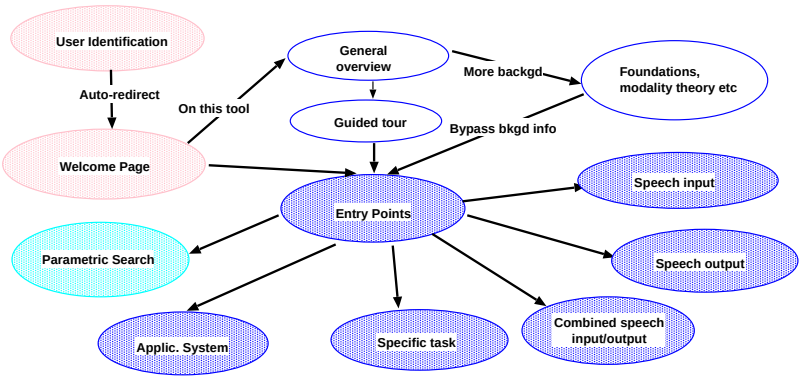

Figure 1. SMALTO's navigation structure.

other modalities. Applications are structured into generic groups according to the parameters that best describe their characteristics as *data points* (see Figure 2). Examples of such groups include: generic systems, specific tasks, functionalities, interaction modes, work environment, learning parameters, user groups, etc. The shallow hierarchy into which the corpus of claims has been structured is loosely defined. One will find at the same level of *generic system*, for instance, entries as diverse as *personal intelligent agents* and *complex relational databases*. This might seem surprising at first, but it has been part of our design philosophy from the outset. Although it would have been possible to define a tighter hierarchy, we have decided not to try and impose an artificial semblance of regularity to a domain of requirements specification where great variations appear to be the rule rather than the exception. Since SMALTO mainly targets decision making that occurs at early stages of systems life-cycles, its users are cautioned to consider carefully all aspects in which their own cases might differ from the cases evaluated by SMALTO.

Search in the database of structured *claims*, parameterized via the domain variables shown in Table I, is also possible, though it is normally hidden from first-time users in order to encourage them to get acquainted with the theory and the navigation structure of the site.

Figure 2 shows a speech functionality claim as rendered by SMALTO on a web browser. The number on the top left corner represents the unique identifier of this claim in the system. The text in the upper box, extracted from (Roth et al., 1997), and linked to the appropriate reference by the icon at the bottom, is a literal transcription of the claim. The data point shows a formalization of the claim according to modality theory parameters, along with its evaluation and links to the relevant modality properties. The modality properties used in the

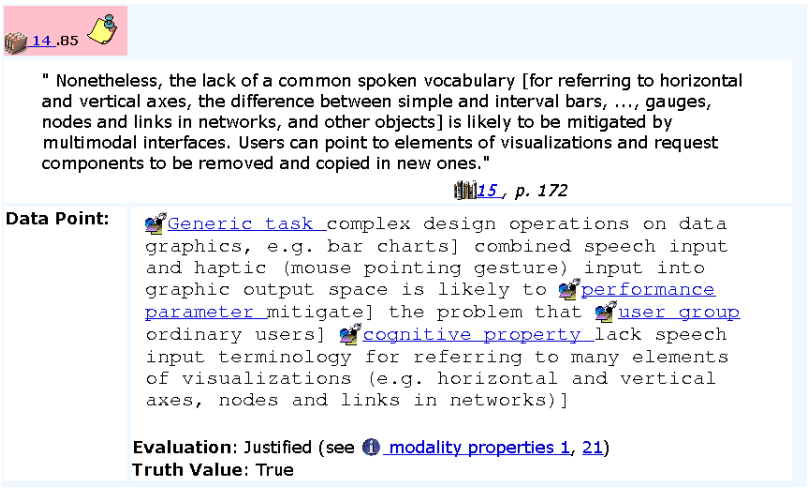

Figure 2. A speech modality claim rendered on a web browser.

evaluation of the claim shown in this example are given below, for the sake of illustration.

MP1 Linguistic input/output modalities have interpretational scope, which makes them eminently suited for conveying abstract information. They are therefore unsuited for conveying high-specificity information including detailed information on spatial manipulation and location.

MP21 Haptic deictic input gesture is eminently suited for spatial manipulation and indication of spatial location. It is not suited for conveying abstract information.

4. A data view of SMALTO's architecture

Claims on speech functionality, as standardized and evaluated by the theory, are the main data component of SMALTO. Speech functionality claims are entered into the system database after rigorous evaluation against the set of *modality properties* (see above) and classification according *parameters* or *domain variables*. Modality properties and parameters constitute the core of the theory and therefore will drive most of the navigation in SMALTO. Claims will often be the outcome of a search process where queries (either explicit or dynamically built during hypertext navigation) are initiated through some instantiation of domain variables backed by certain modality properties.

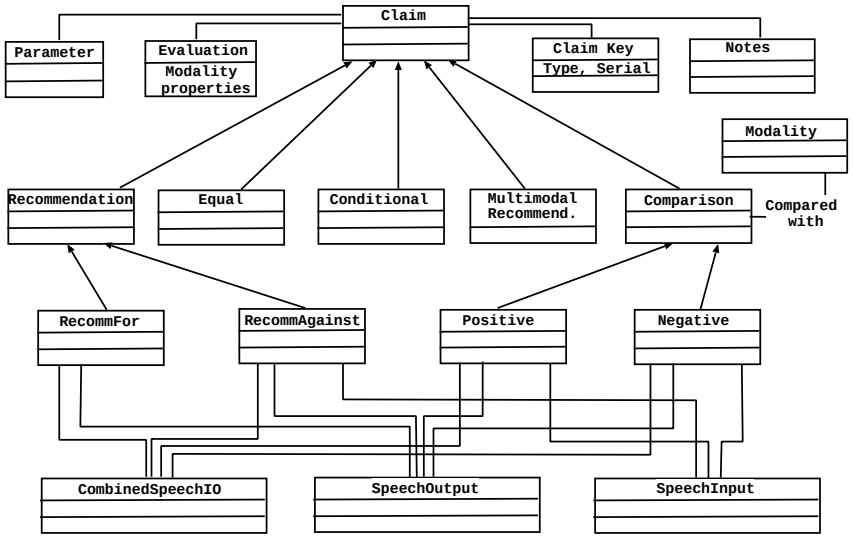

Figure 3. Speech functionality claims.

A hierarchy of claims as presented in (Bernsen and Dybkjær, 1999b) and some of their connections with other data elements are represented[2] by the diagram of Figure 3.

Parameters help *situate* claims with respect to applications in terms of *tasks*, *user groups*, *speech-act types*, *interaction modes*, *work environments*, *generic systems*, *performance parameters*, *learning parameters* and *cognitive properties*. Each claim is assigned a unique identifier and typically points to one bibliographical reference (the *data source*). Assumptions underlying the claim and comments are sometimes added. Thus, a designer (user) may ask: "Are there any evaluated claims concerning the combination of spoken and pen-based input for complex graphics manipulations?". The system will return the relevant claims along with their evaluations and modality properties on which the evaluations are based.

Another key element of our data view of the tool is the description of the *modality properties* which account for most of the explanatory power of this speech functionality branch of modality theory. In order to provide the user with statistics on the generality and explanatory power of the different modality properties, an inverted index of claims by modality properties is used. The modality property viewer enables the user to browse through claims whose evaluations used a specific property.

[2] Boxes represent *classes*, arrows represent *generalisation* and plain connectives represent *aggregation*. The representation is not meant to be exhaustive, since our aim at this point is to emphasize data constraints rather than implementation issues.

4.1. Navigation devices and user notes

In addition to the elements necessary to encode the elements of modality theory incorporated by SMALTO and their data, SMALTO uses other data elements in support of navigation, search and user interaction in general. These data elements are: user generated *annotations* and the *interaction log*. Annotations are first-class objects containing user feedback and whatever else the user might want to add to (his own view of) SMALTO. Access to (and input of) these data elements are signalled on the interface by "Post-It"-looking icons, which is meant to reflect the nature of annotations: personal notes to be (re)collected and used at a later time[3].

Annotations will, perhaps, appear mainly on modality properties and claims. However, comments on the tool itself are also allowed (and encouraged). Searches and paths of followed links will be logged throughout the user's interaction with the system, and may be kept across sessions. User identification interaction is, however, kept to a minimum. Too many forms to fill out often puts off even the keenest user.

The most likely consumer processes for these annotation data elements are: (user edited) messages to the developers and (system edited) navigation reports containing claims and combinations of parameters and/or modalities for which the user has searched. However, future work on the analysis of user interactions with SMALTO is planned as a means to refine and improve the interface of the tool.

4.2. Data collection and XML markup

The data used in SMALTO were collected from a corpus of papers on speech and multimodal interfaces, analysed, evaluated and formatted in terms of the parameters shown in Table I. This pre-processed corpus was then marked up in XML. XML has been used as an intermediate data format between the "raw" speech functionality claims and the compiled and indexed database.

After the markup phase, the XML data undergo parsing, data point extraction and indexing, being then uploaded to the SMALTO database. Parsing is performed using `XML::Parser`, Perl's top-level interface to the `expat` library. SMALTO classes were built which encapsulate the

[3] The mechanism used to support persistent, distributed personal annotations in SMALTO involves a combination of Java applet technology and common gateway interface (CGI). This module, which we call *Y-notes*, has been generalised and is currently distributed as *free software* by the NIS Laboratory. Further details can be found at http://disc.nis.sdu.dk/y-notes/.

original event-based parser in order to perform data extraction and indexing.

The main elements and attributes of a claim *document type definition* (DTD) are the following:

S-CM: the top level claim tag. This element's attributes are:

ID: a unique claim identifier (an integer),
TYPE: the claim type (an integer to be mapped to the table of claim types, as shown in Figure 3),

S-QUOTE: the original wording of the claim. This element has two attributes:

REF: an integer identifying a unique bibliographical reference,
PAGE: the number of the page where the claim is made,

S-DP: the top level data-point tag,

S-PARAM: the domain parameter tag. This element has one attribute:

TYPE: the parameter type,

S-EVAL: the result of the claim evaluation process. The relevant modality properties used in the evaluation of the claim are enclosed in S-EVAL tags. This element has one attribute:

TYPE: the type of evaluation outcome (e.g. the claim may be "justified", "supported", etc,

S-NOTE: the evaluator's notes about the claim and the evaluation process.

S-TV: The truth-value of the claim (e.g. "true", "false", "moot" etc).

The fragment shown in Table II is an example of a valid XML speech functionality claim and data-point.

5. Processing and implementation issues

From a processing perspective, as the user advances beyond the tutorial and the *entry point* node the presentation is built on the fly. This is due to the fact that, given the complexity of the speech modality problem, pre-building all possible pages for all possible variable instantiations in static HTML would be highly impractical. This processing constraint has practical import with respect to the data constraints stated above: since a hierarchical structure can be easily imposed to claim data elements it makes sense to implement each claim as an object with which the modules in charge of dynamically composing the different pages will interact. Implementing each claim as an object maximises

Table II. Sample XML encoded claim.

```
<S-CM ID="3" TYPE="11">
  <S-QUOTE REF="1" PAGE="10">
    Speech output is [slower and more difficult compared] to
    other means in conveying complex information. A variety
    of information can be displayed at once by images and text.
  </S-QUOTE>
  <S-DP>
    <S-PARAM TYPE="Generic task">
      conveying complex information
    </S-PARAM>
    speech output is
    <S-PARAM TYPE="performance parameters">
      slower and more difficult
    </S-PARAM>
    than graphics (combined images and text) output.
    <S-EVAL TYPE="Justified">
      1, 19
    </S-EVAL>
    <S-NOTE>
      "Complex information" is a woolly term which may mean,
      e.g. highly abstract information as well as
      high-specificity information, such as that found in a
      photograph (a static graphic image). What the present
      claim really says, then, is that a combination of
      analogue graphics and any natural language modality,
      such as speech, must be superior in expressiveness to
      speech-only.
    </S-NOTE>
    <S-TV>
      True
    </S-TV>
  </S-DP>
</S-CM>
```

scalability of the *corpus* of claims and modifiability, as the rendering of claim-aggregated data such as links to the relevant modality properties, references etc, can be changed at the top of the class hierarchy.

According to the general data constraints of the architecture, instances of SMALTO objects can be accessed and rendered in different formats. The canonical rendering of any SMALTO object is valid XML. Object persistence is implemented along the lines of the *document object model* (DOM) specification (Apparao et al., 1998). In order to increase accessibility of the prototype, the current version of the system presents data in HTML format and serves them through an HTTP back end.

6. SMALTO in design environments

We envisage that the utility of SMALTO might stretch beyond its use as a tutorial introduction to speech functionality from the point of view of modality theory (Bernsen, 1997a), or as a reference hypertext on interface design issues involving speech modalities. A broader context of use was taken into account in the design of SMALTO so as to enhance our understanding of its intended user's profiles and provide for further development and integration with existing frameworks.

In traditional design life cycles — say, those which use an *architecture design tool* in early design and an *analysis tool* to determine formal properties of the product — SMALTO may serve as an added resource to the architecture design tool, providing *advice*, at the earliest stage of decision making, as well as *evaluation* at later stages (including *justification*, *support* or *rejection* of early-design ideas). SMALTO's advice and evaluation can be used in the generation of *first ideas* and for documentation purposes, for instance. Notice that SMALTO is not meant to be used directly in conjunction with tools that produce detailed specifications (or code) — as in *program synthesis* or *visual programming*, for instance.

A limitation for the use of SMALTO in traditional design environments is that those environments do not favour analysis of partial design representations. Once the design decisions have been formalised — e.g. by means of diagrams, process logics, automata, etc — the next designer iteration will occur only after the design representation has been put through an analysis tool and its formal or semi-formal properties fed back to the design team.

Although SMALTO can be a useful tool for decision support in a traditional (software architecture) design process, we believe that its potential will be better exploited in *domain-oriented design environments*, DODE (Fischer et al., 1992). In such environments the design activity is viewed as a composition of several processes: a decision process, architecture representation, analysis on partial representations, and "critics" (software agents) feedback.

In a domain-oriented design scenario, SMALTO would be used not only at the early stages of the process — via designers' explicit requests or previous experience — but throughout the design cycle. The architecture assumed in this scenario is shown in Figure 4. DODEs, which only recently have started making their way into more traditional CASE tools, typically target system architecture design situations where the number of variables and variable instantiations (or "parameters", as we call them in SMALTO) is too great to be covered by a simple set of guidelines and rules-of-thumb — which we assume to

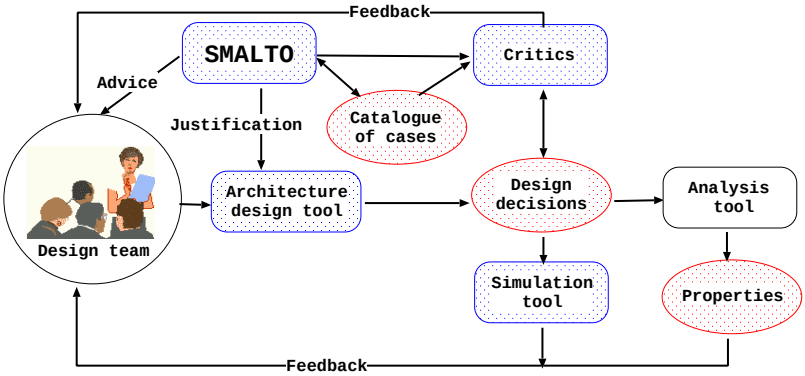

Figure 4. SMALTO in a DODE design cycle.

be the case of multimodal systems involving speech. Domain-oriented tools seek to handle the complexity of the problems they address by providing the user (a system designer) with additional memory capabilities. This is often done by keeping track of decisions made or postponed, and by using "critics", that is, software agents which run on the background and provide situated advice based on a hyperlinked case catalogue (Robbins et al., 1998).

SMALTO's own architecture is flexible enough to permit its integration with this, and other methodology-specific environments by providing support to critics and designers. This kind of support can be achieved by using SMALTO as the basis for argumentative hypermedia, as well as a framework for structuring a catalogue of cases and design situations. Preliminary studies on extending SMALTO in that direction have been conducted within the scope of the DISC project[4].

7. Reported uses of SMALTO and initial user feedback

SMALTO has been used in teaching post-graduate students in interactive media, and has been made available for use and evaluation by DISC advisory panel members. Valuable feedback has been gathered this way.

It has been suggested that the tool can be used in real-world situations where the software designer needs arguments to convince a non-expert (e.g. management, a customer) of the advantages of using a speech-enabled interface. In that case, the software expert would be

[4] Spoken Language Dialogue Systems and Components: Best practice in development and evaluation. The project's website is located at http://www.disc2.dk/.

using the corpus of claims as a means of getting an idea across to a non-specialist, rather than in a practical design situation.

It is hard to tell how their different backgrounds will affect what different users (designers, students, managers) are able to get out of the tool. If nothing else, one can regard SMALTO as an exceptionally well-structured list of *frequently asked questions* (FAQ), whose underlying theoretical principles are made explicit from the outset. FAQs are seldom tuned to a particular user profile. The reason why they are so useful is not so much that one is often able to find a straight answer to one's questions in them. Rather, their usefulness comes from the fact that, through them, one learns what questions to ask.

SMALTO has been inspired by the FAQ model, which we have tried to make more interactive by providing dynamic views of the database, and allowing the user to annotate parts of the text as well as collect those annotations in the form of a report.

8. Conclusion

This paper described a tool for advising designers of multimodal systems on the properties and aspects of speech functionality. We focused this presentation on our main design goals and on the uses we envisage for the knowledge-base and software infrastructure that have been built. Although some level of implementation details has been described, work on SMALTO is still under development and therefore those details might change in the future.

We would like to gather as much user feedback as possible before proceeding with the specification of further functionality. A beta version of the system is available on the World Wide Web at:

http://disc.nis.sdu.dk/smalto/.

Please send us your comments!

9. Acknowledgements

The research described in this paper formed part of the DISC project (Spoken Language Dialogue Systems and Components: Best practice in development and evaluation), an Esprit Long-Term Research Concerted Action of the European Commission. We are grateful for the support. The authors also wish to thank Marc Blasband for valuable discussions and feedback on the prototype.

References

Apparao, V., S. Byrne, M. Champion, S. Isaacs, A. L. Hors, G. Nicol, J. Robie, P. Sharpe, B. Smith, J. Sorensen, R. Sutor, R. Whitmer, and C. Wilson: 1998, 'Document Object Model (DOM) Level 1 Specification'. Technical report, The World Wide Web Consortium (W3C). Version 1.0. Available at http://www.w3.org/TR/REC-DOM-Level-1/.

Baber, C. and Noyes, J. (Eds.): 1993, *Interactive Speech Technology*. London: Taylor and Francis.

Bernsen, N. O.: 1997a, 'Defining a taxonomy of output modalities from an HCI perspective'. *Computer Standards and Interfaces* **18**(6–7), 537–556.

Bernsen, N. O.: 1997b, 'Towards a tool for predicting speech functionality'. *Speech Communication* **23**, 181–210.

Bernsen, N. O. and L. Dybkjær: 1998, 'Is speech the right thing for your application?'. In: *Proceedings of ICSLP '98*. Sydney, Australia, pp. 3209–3212.

Bernsen, N. O. and L. Dybkjær: 1999a, 'A theory of speech in multimodal systems'. In: P. Dalsgaard, C.-H. Lee, P. Heisterkamp, and R. Cole (eds.): *Proceedings of the ESCA Workshop on Interactive Dialogue in Multi-Modal Systems*. Irsee, Germany, pp. 105–108.

Bernsen, N. O. and L. Dybkjær: 1999b, 'Working Paper on Speech Functionality'. Technical Report D2.7, DISC Spoken Language Dialogue Systems and Components: Best practice in development and evaluation.

Fischer, G., A. Girgensohn, K. Nakakoji, and D. Redmiles: 1992, 'Supporting Software Designers with Integrated Domain-Oriented Design Environments'. *IEEE Transactions on Software Engineering* **18**(6), 511–522.

Mynatt, E. D.: 1997, 'Transforming Graphical Interfaces Into Auditory Interfaces for Blind Users'. *Human-Computer Interaction* **12**(1/2), 7–45.

Robbins, J. E., D. M. Hilbert, and D. F. Redmiles: 1998, 'Software Architecture Critics in Argo'. In: *Proceedings of the 1998 International Conference on Intelligent User Interfaces*. pp. 141–144.

Roth, S. F., M. C. Chuah, S. Kerpedjiev, J. A. Kolojejchick, and P. Lucas: 1997, 'Toward an Information Visualization Workspace: Combining Multiple Means of Expression'. *Human-Computer Interaction* **12**(1/2), 131–185.

Address for Offprints:
Natural Interactive Systems Laboratory
Forskerparken 10
DK-5230 Odense M
Denmark
E-mail: luzs@acm.org or nob@nis.sdu.dk
Tel.: (+45) 65 50 35 51
Fax: (+45) 63 15 72 24