Deliverable D1.6

Working Paper on Human Factors Current Practice

May 1998

Esprit Long-Term Research Concerted Action No. 24823

Spoken Language Dialogue Systems and Components: Best practice in development and evaluation

DISC

TITLE	**D1.6 Working paper on human factors current practice**
PROJECT	DISC (Esprit Long-Term Research Concerted Action No. 24823)
EDITORS	David Williams (Vocalis)
AUTHORS	David Williams, Klaus Failenschmid (Vocalis)
	Laila Dybkjær, Niels Ole Bernsen (MIP)
ISSUE DATE	18 May 1998
DOCUMENT ID	wp1d6
VERSION	1.0
STATUS	Final
NO OF PAGES	60
WP NUMBER	1
LOCATION	s:\research\projects\disc\drafts\hf5.doc
KEYWORDS	WP1, human factors, current practice

Document Evolution

Version	Date	Status	Notes
0.1	23/02/98	Draft	First draft published for review from partners. Operetta evaluation to be added by Odense.
0.2	23/03/98	Draft	Second draft with additions of help design and supporting users of differing ability. Also analysis of Operetta in the design and life-cycle grids. Ready for Odense's input.
1.0	18/05/98	Final	Implemented Odense's suggestions and extended discussions.

Working Paper on Human Factors Current Practice[1]

Vocalis Ltd., Chasten House, Mill Court, Great Shelford,

Cambridge CB2 5LD, UK

Abstract

This paper summarises key aspects of the current practice in human factors-related work in the design and implementation of commercial and research spoken dialogue systems. Human factors cover all aspects of interactive system design which are related to the end-user's abilities (perceptual, cognitive and motor), experience (system specific, domain specific and common sense), goals (both interactional[2] and transactional[3]) and organisational/cultural context. In order to provide a complete description of spoken dialogue design, the paper examines both interactive system design processes and discrete areas of human factors work in academia and industry. To ground this discussion, a number of systems are examined in terms of a best practice framework for design process and design practice; the systems are Verbmobil, Waxholm, Vocalis Operetta and the Danish Dialogue System.

---

[1] This paper forms part of Work Package One (WP1) of the ESPRIT 4th Framework LTR Concerted action project DISC - Spoken Language Dialogue Systems and Components Best Practice in Development and Evaluation'. WP1 examines the current practice in all aspects of Spoken Language Dialogue System Development.

[2] Related to maintaining the relationship between communicating parties. This includes ritualistic communication such as politeness.

[3] Related to the external goal of a communication, e.g. finding directions to the theatre.

Contents

1. Introduction

This paper summarises key aspects of the current practice in human factors-related work in the design and implementation of commercial and research spoken dialogue systems. Dialogue systems can be separated into two broad categories. Firstly, there are the more 'traditional' systems which have grown from tone-based Interactive Voice Response applications. These have a dialogue structure (including conditional branches) which is defined at compile-time, i.e. a particular dialogue flow can be specified *a priori*. At the other extreme are *dynamic* dialogues where the dialogue is generated at run-time as a function of the recognised input, or more specifically, the information that has been provided by the user. Within this range of functionality, there are systems which have a mixture of both styles by changing aspects of the dialogue at run-time dependent upon static user data, e.g. usage profiles.

Since this paper is not concerned with the natural language processing aspects of dynamic systems, we will confine our discussions to the post-NLP processing manifestation of the dialogue where human factors decisions are similar for both static and dynamic dialogue systems. Where this is not the case, the distinction between the two types of systems will be made clear.

Before defining human factors it is important to elucidate the range of types of communicative input into a system which can be regarded as spoken-dialogue systems. These are shown in the following list in order of technical complexity. Example interactions are given (S: System, U: User; the words in bold typeface are recognised by the SLDS):

a) *Tone (DTMF[4]) Detect*: Tones produced by pressing keys on a telephone keypad, Interactive Voice Response (IVR) systems use this functionality, voice output.
   S: Press 1 for sales department, 2 for marketing,...
   U: [key 2 pressed on telephone keypad]

b) *Grunt Detect*: Presence of grunt noise or silence as inputs.
   S: ...or stay silent to speak to an operator.
   U: [Caller stays silent]

c) *Yes/No*: Only recognition of the words Yes or No.
   S: Do you want to speak to an operator, yes or no?
   U: **Yes**

d) *Isolated Word*: A small vocabulary of input words (e.g. command words).
   S: ...say balance to receive your current account balance, or operator to ...
   U: **balance**

e) *Word spotting*: One input word can be recognised amongst non-valid words in an utterance.
   S: You can ask for your balance, ..., or speak to an operator.
   U: Can I speak to an **operator** please

[4] DTMF: Dual Tone Multi Frequency. This term refers to the tones produced by pressing a key on the telephone keypad.

f) *Multiple Word Spotting*: Multiple words or phrases which conform to super-lexical grammars can be recognised (e.g. connected digits).

S: Please state your account number

U: My account number is **3748 81847**

g) *Natural Language Processing*: Input utterances are parsed syntactically and semantically. More often in limited domains. Allows unrestricted input with co-references (e.g. anaphora, ellipses). Prosodic information in the utterance can be evaluated. Input can be spontaneous and continuous.

S: How can I help?

U: **Can I book a return ticket from London to Paris please**

- Additional features which can be used in cases c-g) are:
    - Confidence measures: give some measure of certainty that word X has been recognised correctly.
    - N-best: related to confidence. An ordered list of words with attached confidence measures.
    - Talkover (barge-in): Users can interrupt the system by speaking (i.e. speak while the system is speaking). Recognition can be carried out on this utterance.

Human factors cover all aspects of the interactive system design which are related to the end-user's abilities (perceptual, cognitive and motor), experience (system specific, domain specific and common sense), goals (both interactional[5] and transactional[6]) and organisational/cultural context (del Gado and Neilson, 1996). Whilst the remit of this field is broad, the theoretical and practical work tends to occupy a variety of small niches with few unifying approaches defining interactive system design on all of the dimensions noted.

The paper uses a number of exemplars systems to evaluate current design practice against a best practice framework. A brief description of each system is given in the Appendix. These are Vocalis Operetta - an automated call handling system. Examples of research systems are Waxholm, a multimodal system which provides information on boat traffic in the Stockholm archipelago (Blomberg et al. 1993; Carlson 1994; Bertenstam et al. 1995), Verbmobil a spoken language translation support system for German/English and Japanese/English and the Danish Dialogue System, a telephone-based system for reservations of Danish domestic flight tickets. For a more complete list of SLDS see (Bernsen et al. 1998; Gibbon et al. 1997).

The systems were evaluated with a view to Human Factors by researchers who were not involved in their development. The results of the evaluation of Operetta and the Danish Dialogue System were verified with the developers; verification of the Verbmobil and Waxholm results will be concluded in due course.

---

[5] Related to maintaining the relationship between communicating parties. This includes ritualistic communication such as politeness.

[6] Related to the external goal of a communication, e.g. finding directions to the theatre.

As well as specific areas of research and practical experience, the paper also addresses the unique requirements for the design life-cycle of interactive systems. This includes prototyping and descriptive methodologies.

2. Review of Human Factors in SLDS Design

The analysis of the exemplar systems must be grounded in terms of the state-of-the-art of all aspects of human factors work that is being carried out. To this end, a number of areas will be addressed in the following sections in order to provide a background to current work in academe and industry. It is intended, that armed with this knowledge the reader will appreciate the implications of the final exemplars addressing or not addressing particular areas of human factors design.

2.1 Error Detection and Correction

Human machine dialogues, like human-human dialogues, will rarely proceed without any need for some kind of communication which is not directly related to the communication goal. A key example of this is need to confirm that certain pieces of information have been correctly understood. These can be in the form of an explicit request, e.g. Did you say twenty-one?, or a more subtle repetition, e.g. twenty-one?[7]. In both cases the goal is to make sure that important information has been understood correctly as well as offer the possibility to detect communication errors early on. Detecting such errors is essential for the effective completion of the communication, as shown by the examination of human-human talk (For examples see Falzon, 1990).

However, the detection of an error is only first step, the error still needs to be corrected. Correction is needed if information has not been transferred successfully, this can happen for a number of reasons:

- One party does not hear the other
- One party misunderstands and provides an unsuitable response

Again, the error correction can take a variety of forms, ranging from the repetition of the statement in question through to a entirely different kind of dialogue structure. Examples of error correction are given by Cheepen and Monaghan (1997). Here, transcriptions that were made of call centre recordings were examined for error correction dialogues.

In human-human communication, the error detection and correction dialogues form a natural and seamless part of the talk and yet, if analysed fully, reveal a complex exchange of information. Figure 2-1 shows an example from Cheepen and Monaghan (1997).

It is the complexity of the turn-taking within such a dialogue which presents the biggest obstacle to its being modelled in human-machine dialogue, even assuming that recognition is perfect.

A variety of strategies are available to the dialogue designer depending on the recognition technology they have available. In addition, the designer must also be aware of domain-related aspects of the dialogue, e.g. how important is it that a piece of information is transferred

[7] Known as an ellipse, where the question contains an answer.

correctly between human and machine. These two areas, technology and communication domain, interact in a way which determines the optimum strategy a designer should follow. The interaction becomes evident when one considers the use of confidence level recognition technology. This allows the recogniser output to include a confidence measure for each recog-

Agent	**Caller**
..but you said something about you wanted to change the name Miss Ward.	
	Yeh, the name should now be Mrs Franton
Mrs.	
	M. Franton
Franton	
	F-R-A-N-T-O-N
T-O-N	
	yes

Figure 2-1. Example of error detection and correction in a human-human dialogue.

Case Study: The use of confidence measures

A small company requires an automated operator's assistant to answer in coming calls. A voice routing system is suggested which has a confidence level facility and provides the following dialogue:

***System**: Welcome to the Acme Consumables automatic switchboard. After the tone, please say the first and last name of the person you would like to speak to and I will connect you.*

***User**: David Williams*

***System**: Please hold while I transfer you* (High confidence)

A key requirement of the company is that calls are never routed to the Managing Director by mistake. For this reason, even on high confidence the system asks for confirmation of the name that was recognised.

***System**: Was that David Williams, yes or no?*

Case Study 1. Switchboard: The use of confidence measures.

nised word. In addition, there may be the facility to provide a list of vocabulary words which are ordered by their confidence. By using this facility, the dialogue flow can be diverted depending on the confidence level for a particular word (or phrase); a low confidence suggests that the system may have not recognised the spoken word correctly (error detection) and therefore a sub-dialogue is required to correct this error. However, the reliance that is placed on confidence measures must be tempered by the fact that confidence levels can mean that good words, as well as bad words, can be rejected. Given this danger, if a particular piece of information is safety critical, economically significant or would cause embarrassment if it were misunderstood it is necessary to add an explicit confirmation dialogue, even for high confidence results (for an example, see the Case Study 1 over).

2.2 Dialogue Initiative

There are a variety of ways that a spoken dialogue can be constructed.

- *System Directed*
  The system leads the user, asking them questions which determine the dialogue flow. This type of dialogue will be most common in systems which must support novice users or which have a number of pre-defined steps which must be followed in order to complete a transaction or to collect information, e.g. an account number.

- *User Directed*
  Giving the user control is more likely to support more experienced users. In this case, the system provides a range of functions which can be accessed from a main prompt such as "Which service?". A key issue here is how to make the user aware of the available functionality; this will be discussed in the next section.

System	**User**
Welcome to Acme Banking.	
Which service do you require?	
	Transfer
What type of account do you wish to transfer money from?	
	Banking
What type of account do you wish to transfer money to?	
	Savings
How many pounds?	
	Fifty
How many pence?	
	Zero

Figure 2-2. Mixed Dialogue Showing System Directed (bold) and User Directed (plain).

- *Mixed Initiative*
  There may also be the case where the system-directed and user-directed modes are mixed in a single dialogue. For example, a telebanking dialogue may begin with a user-directed service selection but later may provide a system direction dialogue to guide the user through fund transfer (See Figure 2-2).

Dialogue initiative is strongly influenced by the mode in which the interaction between the user and the system takes place. The two main modes used in SLDSs are:

- *Transactional Mode*
  The dialogue between the user and the system is focused towards achieving a common goal quickly. It is assumed that the user knows what type of information is needed and prepared to supply this information in very short utterances (e.g. isolated words). Initiative is usually assumed by the system and system output is constructed to support this. Command based systems with a menu structure (see discussions below) typically operate in a transactional mode.

- *Interactional Mode*
  In an interactional mode initiative usually lies with the user and information between the user and the system is exchanged through longer utterances which can contain numerous pieces of information. The 'dialogue route' to solving the communicative problem is decided upon during the interaction. SLDSs that use a form filling approach (see discussions below) typically operate in an interactional mode.

2.2.1 Menu Design

An important aspect of dialogue design is making the user aware of the systems functionality and how it is accessed. In a visual interface this may be done in a variety of ways, see Figure 2-3, All of the represented methods make use of the permanent and parallel nature of the visual modality. Numerous menu options can be recognised in parallel and at any time. Unfortunately, speech offers transience and serialism[8], making the designers job much harder. However, it also offers a departure from the rigid structuring of menus that do not allow the user to access commands in random order.

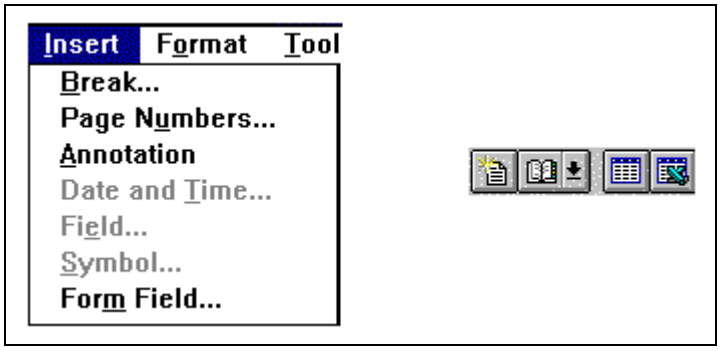

Figure 2-3. Visual Menus.

The advent of ubiquitous telephone network-based speech applications has increased the need to provide intuitive IVR (Interactive Voice Response) interfaces for the automated 'front-ends' allowing users to chose one from many services. Two related problems arise for the interface designer. How to design for users who have a wide range of usage experience and how to provide an intuitive interface for the increasing variety of network-based applications. Services such as call forwarding, call blocking, fax-mail, voice-mail, voice-dialling are now commonly available to subscribers and public users.

Traditionally, automated IVR for multi-function systems have been based on linear and hierarchical menu structures since these are well known in graphical/textual interfaces. However, other approaches have been attempted, including 'skip and scan' (Resnick, 1992).

The 'skip and scan' style of menu navigation (Resnick and Virzi, 1992) gives users more control over system output. By using a cassette-player analogy users can move backwards and

[8] It is possible to provide parallelism in spoken dialogues using non-verbal sounds (see Section 2.2.3) or stereo positioning.

forwards through menemes of a particular menu by using 'next' and 'back' Once the appropriate meneme has been found, it can be selected by saying 'select'. This process is repeated for further submenus.

The advantage of this method is that users do not need to hear all of the menemes in a menu allowing them to quickly select their required option once they hear it. In addition, since navigation is reduced to generic commands, there is no need to explicitly include these commands in the prompts. For example, a dialogue can be reduced to:

System: 'File Menu: Include'

User: 'Next'

System: 'Save as Text'

User: 'Next'

System: 'Log Message'

User: 'Select'

The method can be further enhanced by using talk-over allowing the shortened prompts to be interrupted.

Comparisons between different methods (Jack et al., 1996) have focused on search time and found little different between 'shallow and wide'(few levels, numerous commands per level), 'deep and narrow' (numerous levels, few commands per level) and 'skip and scan' methods.

The structure of a menu in a command based SLDS has a great effect on the usability and naturalness of the system and different approaches to menu design will now be discussed. An approach, namely form filling, for non-command based systems will also be introduced briefly.

A linear main menu or help prompt listing all of the functions (ordered sensibly)

This makes all of the system functionality evident and, if talkover[9] is available, can cater for both novice users (who do not know what commands are available) and expert users (who know the commands available and can interrupt the prompt). However some disadvantages are inherent:

- For the novice user, the long list of possible commands places too much of a strain on short-term memory.
- If no talkover is provided multi-function systems are very slow to use since the caller must hear all of the command names before being able to control the dialogue. If an unfamiliar command is required, placing the list in a help prompt causes a similar problem.

As with visual menus, it is important that mememes are ordered categorically, e.g. editing commands together, navigation commands together.

[9] A facility which allows callers to interrupt the system output.

A Menu-hierarchy

The linear (flat) function list is grouped into menus and sub-menus so that the number of available command words is limited. Speaking a valid command word either executes the associated function or makes a sub-menu available with an other group of command words. This approach allows the semantic grouping of functions making memorising group members easier. It may also support random access to any level via command names. However, the following disadvantages are identified:

- If the hierarchy becomes particularly deep, i.e. many sub-menu levels, then additional commands must be provided for navigation. This compounds the multi-function problem since the user must learn additional non-application commands, e.g. "back", "next", "where am I".
- The experienced user will need to navigate through the sub-menus to reach the function they require.
- Users are required to mentally 'visualise' the menu hierarchy in order to effectively navigate it. This can be problematic for a complex structure often leading to the user becoming ‘lost’ in the menu-space.

Pseudo Sub-Menus (PSMs)

The advantage of sub-menus, without the drawbacks for experienced users, can be obtained by providing a *pseudo* hierarchy using PSMs. In this approach, sub-menus do not provide a confined command space as with the normal hierarchical approach; a space which must be navigated in order to get to the lower function ‘leafs’. Instead, PSMs can be considered as extra-help prompts which group the many available functions and give the function names and sit at the same level as commands.

Thus, from the top-level prompt all of the functions can be accessed by their name. As well as these, each of the pseudo-menu names are active. Selecting these gives a prompt which simulates a sub-menu listing each of the functionally related menemes. However, all of the functions are still available from this prompt; thus giving a flat structure.

The advantages of this approach are mainly related to the experience of the user. The inexperienced user will not be presented with a long list of functions but functions grouped into intuitive sub-menus. However, once the user becomes more experienced with the available commands, they can access all of the system functions directly from the “Which Service?” prompt rather than having to negotiate the hierarchy of ‘concrete’ sub-menus. Interactions like in a linear list as well as like in a menu-hierarchy are possible.

Form filling

The PSM approach in command based SLDSs is a step towards a dialogue management approach that is based on form filling strategies. The core of this strategies is the provision of a form which contains slots for required information for a particular task to be achieved when interacting with the SLDS. For a boat time table such a form could hold information about the place of departure, the destination and the time of departure. The dialogue manager then tries to route the dialogue such that all the slots are filled in the interaction. Depending on what information is already in the form the dialogue manager queries missing information. The users are therefore not constrained in the order in which they give information to the system. The order and the number of distinct utterances in which the information is given can be chosen freely by the user.

2.2.2 System output design

It is somewhat artificial to separate system output wording from dialogue flow but it is suitable for the current discussion. The user-facing part of a spoken dialogue is the prompting, i.e. the system's voice[10] and it is here that the designer provides a representation of the system state. System output utterances (prompts) are analogous to the visual components of visual interfaces, i.e. perceptual properties (size, colour, contrast, shape).

Table 2-1 shows system output defined on four dimensions along with possible counterparts in visual interfaces.

Attribute	**Visual Counterpart**
Wording (e.g. politeness)	Organising Metaphor, icons resemblance, text
Register (e.g. speed, pitch)	Type setting, colour, size, shape
Tone (prosody)	No equivalent
Real/Synthetic	Digitised images/geometric or drawn images

Table 2-1. Prompt Dimensions.

All of these aspects must be considered in the particular context of use within a dialogue flow. It is interesting to note that spoken words provide a rich variety of dimensions that can encode information, some of which have no parallel in visual interfaces. The main difficulty in the effective design of prompts is their evanescence along with the lack of specificity of spoken

[10] The system voice can be augmented by non verbal sounds or a variety of speakers (see James, 1997).

language and the requirement for highly contextualised interpretation. Given these constraints it is essential that the following areas are addressed:

- *Language*
  Before embarking on the design of a prompt list care must be taken that the vocabulary and grammatical structure matches the abilities of the end-users. For example, any domain specific jargon should be avoided where possible. It is important to point out that the wording of the system output affects the recogniser vocabulary. Key words used in system outputs have to be present in the recogniser vocabulary since users are likely to repeat them.

- *Prompt Context*
  It is essential that the designer is aware of what has occurred earlier in the dialogue. This will determine what words and register should be used, i.e. the context of the prompt. For example, as a user moves 'deeper' into the dialogue it may be justifiable to shorten prompts. There may also be particular pieces of information that need repeating.

- *Communication Context*
  It may be necessary to consider the global context of the interaction. For example, a banking dialogue will have its own 'etiquette' or speech style. Work by Williams et al.(1998) suggests that such highly transactional domains require a particularly curt style of speech which does not contain politeness words, personal pronouns or anthropomorphism.

In addition to that the type of system output needs to be considered since this has a strong effect on the kind of response a user is likely to give. There are three main types of system output:

- *Complete Speech Recordings*
  Whole messages are recorded by a professional speaker and played to the user as a whole. If the prompts are carefully recorded, this is the highest quality speech output that can be used.

- *Concatenated Speech Recordings*
  With this method only recordings of parts of phrases are made by a professional speaker. When a complete phrase is to be played out to the user, relevant parts are concatenated together and then played to the user as one utterance. Recording such prompts is more complex than recording individual phrases, since intonation of items changes according to the position within a phrase. For instance, intonation for individual digits varies according to the position in a string of digits. Concatenated speech is regularly used to speak back telephone numbers to callers, however, only one recording per item (e.g. digit) is normally used which results in 'robot-like' utterances.

- *Synthesised Speech*
  The most versatile means of output generation is the use of a Text-to-Speech (TTS) system. Such a system is capable of producing synthetic speech output from a text string. Although this allows the production of virtually any prompt, it has limitations and disadvantages. The speech quality is usually much worse than pre-recorded speech, and it can be very difficult to use a Text-to-Speech system to correctly pronounce certain classes of words such as

names and place names. Current Text-to-Speech systems either produce prosodically rich speech output and low comprehension, or high comprehension and low prosody.

In contrast to concatenated speech which can sound 'robot-like' and synthesised speech which does not (yet) imitate real speech very well, complete speech recordings can be easily perceived as 'life-speech' from a real person. Users are therefore more likely to respond in a conversational manner (e.g. long utterances) to a system that makes use of complete speech recordings. Speech output that is more obviously created by a machine (concatenated prompts or synthesised speech) attracts more constrained responses. Hence, not only the degree of naturalness in the system responses, but also the capabilities of the speech recognition technology that recognises and interprets the user's utterance need to be considered when deciding on the method of speech output.

2.2.3 Cues for Turn-taking

Not only in turn-taking based systems output also must encourage users to speak at the appropriate time and be passive when no speech can be processed. Thus at the lexical level, a typical human-computer dialogue in a spoken language system consists of two stages, system output and user input. Critical to these stages are the cues that allow effective turn-taking to be carried out, i.e. cues for the user to speak (and the system to 'listen') and vice-versa. Even in systems that do not follow a rigid structure of turn-taking it is important to clearly signal when input from users can not be accepted.

Cues may be explicit, e.g.. "Please speak after the tone <beep>" or implicit, e.g. the user stops talking. As with human-human conversation, a good proportion of turn taking clues are given by lapses in talk, i.e. silences. These potentially ambiguous representations are resolved by a variety of means such as surrounding semantic, syntactic and prosodic information or physical gestures, e.g. eye-brow raising. To utilise some of the verbal cues such as surrounding semantics and syntactic and prosodic information SLDSs can make use of the following components:

- *Syntactic Parser*
  Based on a syntactic description of utterances and phrases that are expected by the system a syntactic parser identifies the type of phrase that was uttered by the user. It can distinguish whether a string of words is the transcription of a question, a statement or other kind of speech act that is necessary to distinguish in an application. This not only helps to constrain the semantic interpretation of the utterance but also provides a clue as to whether the user is likely to have finished speaking an utterance and a response is needed. A syntactic parser tends to be tightly integrated with a semantic parser.
- *Semantic Parser*
  In conjunction with syntactic parsing it is normally useful to analyse the utterance spoken by the user semantically. Semantic analysis makes use of syntactic and domain specific information of the language likely to be used by the user to find out the meaning of an utterance. Semantic parsers often make use of dialogue history to make it possible to deal with language phenomena such as anaphora or ellipses. Understanding the meaning of an

utterance enables the system to find out whether an utterance is 'complete' and therefore a response from the system is expected.

- *Prosody Interpretation Component*
  In addition to syntactic and semantic parsing, prosodic interpretation can be used to ascertain where a sentence uttered by the user ends and what type of sentence it might have been. Simple prosody such as raising and lowering intonation are normally used at the end of sentences and to mark questions or statements. Coupled with syntactic and semantic information prosodic information can give a very strong clue as to whether a response from the system is expected by the user and what type (e.g. answer to a question, clarifying question, etc.)

However, systems that only use speech as input medium cannot respond to non-verbal cues. Furthermore, in automated spoken dialogues, silences on the system's part my not be so easily resolved if they are of the implicit kind. In addition there is the requirement for basic signalling of an open channel, i.e. the system has not crashed. What is lacking in current (speech-only) system is any explicit representation of these system states (the Waxholm system has an animated face on a monitor which informs the user of the state the system is in).

Whilst the meaning of the initial silence may be implied by the preceding prompt, e.g. questioning intonation, the move from speech processing to application processing is rarely explicitly represented in the dialogue. The consequences of this are a result of the user's interpretation and action, and the incongruence between the system response expected by the user to their input and the actual response. For example, if the user misinterprets the silence after finishing speaking to mean that the system has not recognised their speech, they may speak again. This can have two consequences depending upon whether the system is processing the speech or in the application state (e.g. accessing a database). In the former case, the user's speech will be processed as part of the original utterance and could well cause a misrecognition. In the second case, the users utterance will be ignored, causing an unexpected system response if the utterance was not simply a repetition. This sort of misunderstanding is often only detected in a later state of the dialogue.

The problem lies with the paucity of representation for a number of system states. Since users rely on gaining application knowledge from the interface representation of the system state, without this knowledge the user is liable to act erroneously or react negatively to unexpected system output. What is required is a solution based on the adequate and intuitive representations.

Sound	**Referent**
Stylised wind chimes	Top Level prompt, e.g. Which service?
Final resolving piano chords	Valediction
Several taps on hollow bottle	Name management mode

Two discordant chords	Error

Table 2-2. Non-Verbal Auditory Mappings (from Dutton et al. 1997).

This has been addressed by a number of authors (Brewster et al., 1994; Dutton et al., 1997) who suggest that non-verbal sounds in a dialogue would provide additional information on system state. For example, error conditions (misrecognised or invalid inputs), changes in dialogue mode (vocabulary, grammar) or greeting and valedictory messages. Possible mappings are shown in Table 2-2.

2.2.4 Supporting User Abilities

If a truly user-centred design process is to be followed the designer must initially identify the target end user population. This analysis will allow the level of help required in the dialogue to be decided. All automated dialogues must be designed to support the eventual end-users of the system - the callers - and consideration must be given to two major aspects of the caller/system communication. First, the caller must be able to understand what the system is capable of, and second, the caller must understand how to proceed with the dialogue in order to achieve the goal(s).

2.2.4.1 Help Design

Help information is intended both as an addition to and a respite from the mainstream dialogue and can be portrayed in a variety of ways:

1) As an implicit part of the dialogue, e.g. command names made explicit or example utterances provided ("Say connect to be connected or operator to speak to an operator").

2) As an alternative part of the dialogue which must be explicitly requested by a 'help' command. This includes:

- demonstration utterances
- a more in depth description of commands
- a sample dialogue

3) Automatically enabled if the user is having repeated misrecognitions (detected by the system through confidence measures or explicit disconfirmations). This includes:

- progressive or incremental help (each time a user encounters recognition problems, new more explicit help is provided; from list of command words to example utterances)[11]
- suggestive prompting (those alternative commands which have not been used are suggested)

[11] This facility is used in the Vocalis VAD (SPEECHtel system)

Help information may be specific to the current context or provide generic advice on the available services or how to speak more clearly. The information should do the following:

- Tell the user what has gone wrong (in the user's not the system's language)
- Tell the user how to get out of the help message
- Encourage the user not to get into the same situation again

Help messages should not be as short as possible. Often a certain degree of verbosity is required to explain the problem (and ways to resolve it) to the user. If an error loop is encountered the system output for each error loop should be adjusted following the guidelines above.

Providing useful help mechanism remains a difficult task not only because the kind of help needed is highly user specific, but also because the task domain impacts upon it.

2.2.4.2 Dealing with Naive, Novice and Expert Users

Different application domains (e.g. banking, ticket reservation) may share some dialogic features, such as menu navigation commands, which will allow callers to use experience gained interacting with one system in the interaction with an other. This, however, is not necessarily the case for all sections of all systems. Additionally, the designer must bear in mind the need to cater for the absolute novice - i.e. the caller who has never before used an automated system. In other words, callers do not bring the same initial kinds of knowledge, experience and abilities to different application domains. Key areas where differences in caller characteristics may arise are:

- familiarity with the general domain
  - e.g. in a telebanking domain, does the caller know about statements, balances etc.?
- familiarity with automated spoken dialogue systems
  - does the caller know roughly what to expect when talking to a machine?
- caller status
  - is the caller either a member of the general public who may use the system only rarely, or a (potential) expert who may (in time) use the system with great frequency?
- caller motivation
  - what is the advantage to the caller of using an automatic system?
  - is the caller paying for the call?

Case Study: **Serving different levels of experience**

A voicemail system is required which is driven by voice and will allow users to communicate with other account holders. To provide a more personalised service the following user profile is defined:

Personalised Greetings

Login Count Usage Count per Function This log is used to allow prompts to be shortened for an expert caller (called in many times over a short period) What do you want to do? -> What now? You have received a new message -> New message

Case Study 2. Voicemail: Serving different levels of experience.

Given these different knowledge types users can be broadly categorised into:

- *naive users*
  no experience of domain or spoken dialogue systems
- *novice*
  experience of domain and/or spoken dialogues. Some experience with specific application
- *expert*
  frequent users of the application and expert in the domain

A given dialogue does not need to support all of these users, rather the designer should be aware of what the typical user will be. Of course, there will always be naive callers to a system, the key is how often they use the system and therefore how quickly theywill become experts. Ideally, a dialogue should provide for this transition if a system is to be used frequently. If this is not the case, help information should be 'near the surface' of the dialogue since there will be no expert users to be hindered.

Examples of dialogue styles which allow the transition from novice to expert are:

- Dialogues using *talkover* or *barge-in*. The long prompts required for new users can be interrupted by experts.
- A progressive help mechanism that begins with a prompt for an expert, e.g. "Which service?" , but quickly lengthens if a caller is having problems, e.g. "The services available are..."
- A change in prompt style which is more suited for a particular caller (see Case Study 2 above)
- Suggestive help mechanisms which reveal increasingly more complex aspects of the system's functionality to experienced callers.
- Provide a backup human operator for calls where the user experiences persistent difficulties.

The ideal situation can occur in systems that have some form of user identification. This allows a specific user's usage pattern to be matched to the relevant help strategy (see User Modelling and Case Study 2)

2.3 Shaping User Expectations

As a user spends more time with a system research has shown that they adapt their verbal behaviour both to the system's apparent capabilities and to the style of language the system uses. An equivalent process is demonstrated in human-human dialogue where the participants settle on an 'operational' language that they are both able to understand. These may contain a restricted vocabulary, grammar and semantics.

Franzke et al. (1993) showed that subjects interacting with a speech system over a human operator used fewer words and less complex grammars. Zoltan-Ford (1991) suggested that user input is shaped by system output. This means that care must be taken not to encourage callers to use language that the system does not understand. In general terms this type of mismatch occurs when the user's model of the system (its perceived capabilities) differs from the system (actual capabilities). It is the role of the interface to make sure that this mismatch does not take place. Of course, the interface cannot be entirely blamed for users trying to operate outside the system envelope. The user may bring with them knowledge from other systems which they think is relevant to the new system. For example, they may be used to a system where word spotting is employed so that key words can be spotted amongst noise words, e.g. "I'd like to place an order please", moves to a system that only recognises isolated words, e.g. "order". A similar phenomenon can occur with respect to permitted grammar constructs in the input utterance. This incorrect analogical reasoning can cause spurious recognitions by the system and degrade overall performance.

2.4 User Modelling and User-Dependent Dialogues

2.4.1 Anonymity vs. Identity

Two types of speech systems can be defined, vis a vis their knowledge about the user and their activities. The first type is typical of utility applications for the general public, e.g. automated call centres, electricity or water metering. Here the user provides no identification, they can be anybody. In the second type of systems the users are given a personal account number which identifies them to the system. Such systems have a database record for each user which can contain a variety of dialogue or more usually domain-related information.

In addition to information gathered across whole transactions, there may also be the facility for information garnered between individual dialogue steps to be stored and used.

Case Study: User modelling

A voicemail system is required which is driven by voice and will allow users to communicate with other account holders. To provide a more personalised service the following user profile is defined:

Personalised Greetings

Login Count

Usage Count per Function

This log is used to:

provide recommendations of which services have not been used very often;

e.g. *I'd recommend you say play, back, delete*.

shorten prompts when usage exceeds a threshold for any given command

e.g. *What do you want to do?* becomes *What Now?*

Case Study 3. Voicemail: User modelling.

2.4.2 Uses of User-specific Information

The use of user-specific information will often depend on the needs of the application. However, a number of more generic uses can be identified:

- *Prompt Adaptation*
  As users become more experienced with a system there may be less need to use verbose prompts. Knowledge about the user's transaction history allows new shorter prompts to be used once the transaction count exceeds a certain number.

- *Dialogue Flow Adaptation*
  As well as prompt adaptation there is also scope for the dialogue itself to change. For example, more experienced users may wish to input multiple pieces of information at once rather than use a system directed dialogue.

- *More complex recognition*
  The use of anaphora in human speech is common. Such references can only be resolved if previous information is stored as the dialogue progresses. Given this is the case, the possibility of correctly interpreting "Move the money to that account" becomes a reality. In truth, such complexity is not in the realm of commercial applications but will be soon as user populations become more comfortable with speaking to machines.

- *Suggestive Dialogues*
  Common user behaviour can be extrapolated from user data allowing systems to suggest particular dialogue strategies.

2.5 Culture and Dialogue Design

Spoken dialogue systems are now sold around the world, therefore it is essential that the linguistic and meta-linguistic content and logical description of tasks is matched to the target country (del Gado and Neilson, 1996). For instance, designers of a word-spotting based system to be deployed in the USA have to consider that Americans are known to use the phrase "no problem" to mean "yes". If a system is designed such that it only spots the words "yes" or "no" in an utterance, the system response to a "no problem" answer given by a user will be completely wrong.

3. Previous Work in Spoken Dialogue Design Life-Cycles

Thus far, this paper has described individual aspects of spoken dialogue design. However, there is more to effective dialogue design than addressing these differentiated facets, a design and implementation life-cycle must be considered.

3.1 Why are Interactive Systems Different?

The structure of the design life-cycle of spoken dialogues is determined by the interactive nature of the artefacts under design; interactivity necessitates a particular style of design for the following reasons.

- Used by people and organisations;
    - have prejudices
    - learn
    - erratic behaviour
- Can only be fully validated by use
- Systems are complex; interfaces are more complex and environments are even more complex. The designer must be aware how these different components interact.
- It is difficult to validate systems with a human component since human behaviour is highly erratic and contextualised.

3.2 An Introduction to Interactive System Design

Successful interactive system design focuses on representative users (where possible, any user in the pragmatic case) being included as early as possible in the design process. With this in mind, Poltrock and Grudin (1995) suggest four key aspects of interactive system design processes.

Early and continual focus on users: Designers should have direct contact with intended or actual users via interviews, observations, surveys and participatory design. The aim is to understand users' cognitive, behavioural, attitudinal and ergonomic characteristics and the characteristics of the job/task they are doing.

Early and continual user testing: The only presently feasible approach to effective user interface design is an empirical one. This involves observations and measurements of user behaviour with mock-ups or pilot systems.

Iterative design: A system must be developed in response to user-tests of functions and the whole interface. Early iterations should be cheap and flexible, e.g. text versions of voice dialogues. As the product develops later iterations will involve users less and be on the target

computing platform. The (software) tools to make interface design iteration possible must be available.

Integrated Design: All parts of system development (user interface, documentation, training) must evolve in parallel and should be under one management.

In addition to these four key aspects of the design process continuous and comprehensive analysis and evaluation of the information gathered as well as the design process has to be performed. It is not enough to measure user characteristics etc. Care has to be taken that the right (appropriate) characteristics are measured using the correct methodology.

Case Study: The perils of Ignoring Users

A large bank decided that it would provide a speech and tone driven telephone banking service. One of the services offered was the ability to pay a bill. To do this the customer was required to nominate a bill reference number and an amount. The transaction was then confirmed and the caller was asked if they required another of the banking services on offer. This dialogue was decided without any consultation with end-users (other than the designer-employees themselves) and was implemented and deployed.

Once out in the field it appeared that the bill payment dialogue was causing the average call length to increase. The reason was not because of poor speech recognition but was related to the typical behaviours of banking callers when paying bills.

It transpired that users often wanted to pay a clutch of bills within one call. The initial dialogue design allowed this, but only after renegotiating the whole dialogue again. This proved both time consuming and frustrating for callers. In response to this user study, the dialogue was changed and users were given the option 'Pay another Bill?'.

Case Study 4. The perils of Ignoring Users.

Life-cycle studies can be divided between those related to interactive design in general which are relevant to spoken dialogues and those which are specific to spoken language systems. Essentially, speech systems are no different from other more traditional interfaces however distinctions are made in the literature, so for the sake of consistency this paper will make a similar dissection.

Currently, the majority of spoken-dialogue systems that are developed in industry do not follow the same processes as other interactive applications since they are viewed more as technical systems. There is little formal user analysis (tasks, environments) and evaluation

occurs late in the product's life. The results of the late evaluations often uncover common-sense errors which could have been corrected during the requirements capture stage of the product (See Case Study 4).

3.3 Commercial vs. Research

The design and development of speech systems does not occur in isolation. Rather it is part of an organisational context which places constraints on the type of life-cycle that is possible. It is often the case that the ideal life-cycle is not adhered to due to contextual reasons. A gross distinction can be made between industrial and research environments. Important aspects, i.e. those which directly affect system design processes, of these two environments will now be discussed in detail.

3.4 Time Constraints

In an ideal interactive system design process a great deal of time is devoted to user studies and iterating of designs on what may be a number of different design tracks. This takes a great deal of time. Within an academic environment this time is more readily available. However, this time is not necessarily spent on Human Factors improvements, but rather on technical improvements. A selective focus on one issue of the complete system in an academic project might mean that other aspects do not attract enough attention in the design process.

Industrial projects operate within much more rigorous constraints which may well be outside the system developers control and occur unexpectedly. In particular, a key constraint with speech technologies is the need for rapid time to market since a number of companies are developing similar technologies.

3.5 Specifying Requirements

The specification of requirements in an industrial context can be of two types:

- Client-led-third party provision: the contract will be based on the satisfying of these requirements. It is however not uncommon for the client to change requirements at a late stage.
- In-house led: a new product range: the product requirements will be set by the marketing department in response to the market trends. Alternatively, there may be a need to rapidly demonstrate a recent advance in technology. In this case, the actual application will be of little relevance, e.g. confidence levels for multiple word spotting, echo-cancellation.

Research applications will not have such rigorously defined criteria except in the case of close industrial collaboration and/or funding. It is also possible that the researchers simulate the industrial setting in order to put appropriate emphasis on all life-cycle aspects, though this is not common.

3.6 Working within Existing Design Processes

The 'waterfall' model is predominant in industry but is not ideal for systems where rapid and frequent changes are required. Its main drawbacks lie in the need for rigorous and comprehensive reporting and documentation at the culmination of each design stage. This often means that a design which may be drastically changed by user trials fully documents what may be a fundamentally flawed design. In reality this rarely arises due to the very overhead involved in changing the documentation.

Within a research environment which does not seek to simulate an industrial setting problems occur in the opposite direction, there is little formal design process. This leads to incremental changes with little formal justification making software re-usability difficult.

3.7 Access to Users

The use of representative users is an essential part of interactive system development. However, industrial partners often do not wish their products to be seen by the general public in the early stages of development, consequently user testing is left until the beta-trial stage, when there is a much greater inertia to change. If any early testing does take place, a less representative sample is normally obtained from company employees for fear of adverse publicity.

In a research environment, users will more often than not be non-representative student users. However, it is becoming common practice for market research agencies to recruit users who match criteria of a given profile covering age, occupation, gender, etc.

3.8 Validation and Verification

One of the most challenging aspects of systematic user interface design is the generally held view that human factors aspects of designs cannot be systematically and repeatedly validated. For this reason they are not defined in functional specifications and not included in test specifications. In this way, they suffer pauperdom in contrast to the quantitative measures of active vocabulary size, recognition performance and downtime per year. For this reason, Roast and Siddiqi (1997) attempt to place user interface requirements "on the same footing as functional requirements, and expressing and analysing them with well developed formal notations and techniques" (pp. 155).

4. Current Practice in Spoken Dialogue Design Life-Cycles

The complex nature of spoken dialogue systems requires that their design and implementation impacts on a number of different skill-bases over some considerable period of time. Not least in this process is the consideration of the type of product life-cycle which is to be adopted. This section of the paper investigates a number of approaches which have been adopted in industry and academe. These are cited in order to ground the discussion of the previous section in tangible and complete design process. The section concludes with a description of what is still lacking in these processes, *vis-a-vis* the specific requirements and challenges of interactive system design.

4.1 Bernsen et al. (1998) - Designing Interactive Speech Systems

This work is on spoken dialogue systems design with strong focus on the dialogue component. It advocates a design process which is based on an underlying interactive speech theory; summarised in Figure 4-1. Central to this theory is that a designer will have some clear idea of user populations and tasks in order to provide a basis for launching the design. With this established, an *interaction model* is defined which captures, at a technology independent level, the dialogue flow. This model is iterated using two sources of information.

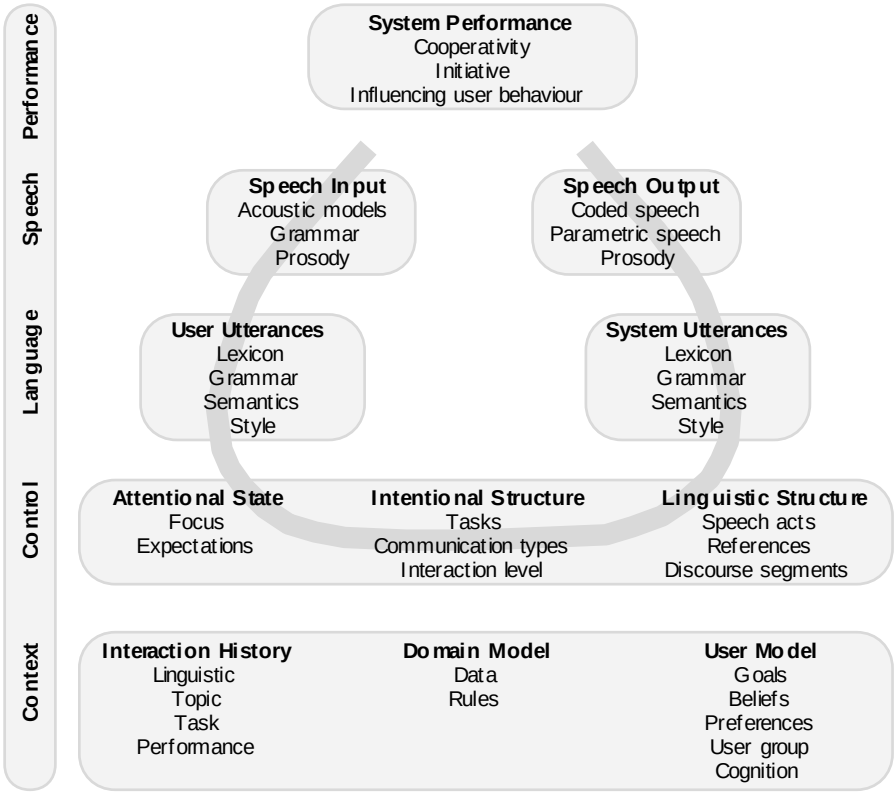

Figure 4-1. Elements of interactive speech theory. Element types are shown in bold. The grey band and grey boxes reflect the logical architecture of spoken dialogue systems.

The first is a set of 13 generic co-operativity guidelines which include Grice's maxims for effective human-human communication, plus a set of 11 specific guidelines. The fundamental assumption is that a system should always be co-operative. For examples of these guidelines see Bernsen et al., 1998.

The second source of information is the results of Wizard of Oz experiments (or alternatively, results from implement-test-revise iterations).

There is little distinguishing between commercial and research development in the design process. This is unsurprising as the recurrent exemplar in the life-cycle description was the Danish Dialogue System, a long term research project.

In summary, the following life-cycle is proposed:

Survey: Provide an overview of project feasibility in terms of strategic goals, e.g. service improvement and cost savings, system goals and constraints, e.g. feasibility estimations for the domain and task(s) and estimation of the ability to satisfy end-user requirements, and resource constraints, such as time and manpower available.

Requirements Specification: The clear identification of goals and constraints which the envisaged system should meet. At this stage, precisely how the requirements are satisfied is not decided, this comes in the functional specification. The requirements document should be iterated with the client. It is developed in the survey phase.

Evaluation criteria: Established for use in evaluating the final system and based on the requirements specifications. The evaluation criteria state the parameters that should be measured and the measurement results that should be achieved for the final system to be acceptable. They are developed in the survey phase.

Design specification: Operationalises how to build a system which will satisfy the requirements specification and meet the evaluation criteria. The design specification is primarily developed in the **analysis and design phase** following the survey phase.

Defining completeness and consistency of specifications: The elements of Figure 4-1 may act as a high-level checklist of functional aspects to be considered for inclusion.

Representing design space: Design Space Development and Design Rationale (DSD/DR) is proposed as a tool to keep track of consensus building during development and evaluation and to represent particularly important pieces of design reasoning.

Interaction model development: The factors to consider in developing the first interaction model are all or most of the elements of Figure 4-1. The set of co-operativity guidelines mentioned above are proposed as a tool which may help ensure an appropriate dialogue model design. Development may then proceed along two different routes. Given a first version of the interaction model, one may either proceed straight to implementation, following the **implement-test-and-revise** strategy, or one may choose to **simulate** the interaction model before implementation.

Evaluation: Should alternate with development. At any stage of development the following three types of evaluation may be applied:

- *performance evaluation*, i.e. measurements of the performance of the (simulated or real) system and its components in terms of a set of quantitative parameters;
- *diagnostic evaluation*, i.e. detection and diagnosis of design and implementation errors;
- *adequacy evaluation*, i.e., how well do the (simulated or real) system and its components fit their purpose and meet actual user needs and expectations.

Moreover, objective evaluation addressing objectively measurable parameters of system or component performance, and subjective evaluation addressing the opinions which users have formed of the system, are mentioned. Finally, different types of tests, such as controlled user test, field test, and acceptance test, are distinguished and described.

4.2 Gilbert, Williams and Cheepen (1998) - Guidelines for Advanced Spoken Dialogues

The work of the Guidelines project focuses on a traditional framework for dialogue design. 'Hanging off' this framework are a variety of guidelines. The guidelines are based on:

- Linear and iterative approaches
- Empirical investigation of *de facto* standards in commercial dialogues.

Central to the life-cycle, unlike Bernsen et al. (1997) is the applicability and uptake of the guidelines within an industrial context. To this end, the project began with an investigation of the current design practice in commercial environments (See Cheepen, 1997). With this as a starting point, the work has focused on identifying the different audiences within the company which are involved in dialogue design, i.e. developers, marketing/sales and researchers. Each group has a stake in the system but very different requirements in terms of the information they require in order to follow a suitable design process. Only by addressing each group in their own terms, can a process description hope to succeed.

In addition to the iteration of the process within a commercial setting, the project also challenges *de facto* standards within the industry. This includes:

- The use of human-like tokens in (English) system output prompts, e.g. 'please', 'thank you', 'I', 'you' (See Williams and Cheepen, 1998).
- The use of only verbal aural output in dialogue.

4.3 A Typical Industrial Life-Cycle

But what is really happening within the commercial dialogue community? There are only a small number of interactive spoken dialogue system projects to examine. The following example represents typical aspects of these. The life-cycle assumes a third party vendor is providing a customised speech system for a client.

Initial Client Meeting: The meeting is an opportunity for the client to outline its requirements.

Tender: The client offers the contract. This will contain a high level 'service' description which will specify functionality and a rudimentary dialogue. No user information will be included.

Response: The third part provides a response specifying which aspects of the tender can be complied with. Since no dialogue requirements have been specified there is no assessment of how user requirements will be met.

Requirements Documentation: In conjunction with the client, requirements will be specified. This will specify the functionality to be supported and will **not** assess whether this is actually the functionality that is required by the end-users. This document will be updated on a regular basis. The client and contractor will sign this agreement.

Functional Requirements: A contractor document which will specify system and dialogue design, including suggested system outputs. The dialogue and prompt design will not be validated in any way, save with the management level (not end-users) of the client company. No prototyping of dialogue flow and/or prompting will be carried out.

Testing: Functional testing. No usability testing as no usability targets are specified in the functional specification.

Implementation and Trials: The system will be integrated. System problems (excluding usability) will be ironed out first. It is often only much later that it is realised problems are a result of poor dialogue design. This includes the functionality offered, command vocabularies and tasks decompositions.

Revision: At this stage users will finally be asked what are the problems. Invariably, customer confidence has been damaged.

4.4 Gaps in Current Industrial Design Life-Cycles

The life-cycles discussed attempt to incorporate interactive system design rationale into current non-interactive system design models. What they lack are:

- A set of measurable standard usability objectives which can be integrated into functional and testing specifications.
- Education of clients in the need for extensive user studies to guide requirements. In particular, clients need to make their end-users available.
- Clients need to be willing to accept longer design and development times brought about by an iterative design process.
- Systems need to be thought of also in terms of satisfying some need rather than as a piece of technology. Ideally, from a user perspective, technology has to be constrained by the interaction, and not the interaction by the available technology.

5. Human Factors Design Checklist

To evaluate the exemplar systems (see the Appendix for brief descriptions), a taxonomy of human factors related issues is proposed which is based on the previous discussion. Each exemplar system is examined on these criteria for specific aspects of their functionality and design process. At the end of each criterion a discussion draws any important comparisons between exemplars. ***In all of the following tables, a `X` shows a positive response to the best practice criterion, a `?` shows an unknown quantity; a `-` shows a negative response. Aspects which are not relevant to a particular system are defined as NA.***

Key: **WAX**-Waxholm, **DD**-Danish Dialogue System, **VM**-Verbmobil, **OP**-Operetta

5.1 Input From User (System Understanding)

Attribute	WAX	DD	VM	OP
Speech Style (SS)				
SS1. Does the system understand connected words such as connected digits?	X	X	X	NA
SS2. Does the system restrict input to isolated words ?	-	-	-	X
SS3. Does the system allow continuously spoken utterances?	X	X	X	X
SS4. Is grunt detect / silence detect available?	?	X	?	X
SS5. Is there a limit on the length of an utterance?	-	X	-	X
SS6 Is it possible to adjust max. utterance length?	?	?	?	X
SS5. Is prosodic information interpreted?	-	-	X	-
SS6. Are anaphora understood by the system?	X	-	X	NA
SS7. Are ellipses understood by the system?	X	X	X	NA
SS8. Are confidence levels supported	X	X	X	X
SS9. Are command words provided for dialogue navigation?	-	X	X	NA
Natural Language Processing (NLP)				
NLP1. Is syntactic parsing carried out?	X	X	X	NA
NLP2. Is semantic parsing carried out?	X	X	X	NA
NLP3. Is the dialogue specified *a priori*?	-	-	-	X
NLP4. Is the dialogue dependent on the user providing information, e.g. a destination as well as a starting point?	X	X		NA
NLP5. Are discourse segments/topics used?	X	?	X	NA
Multimedia Input/Output (MMIO)				
MMIO1. Are the following channels of user input supported:				
Speech?	X	X	X	X
Gesture?	-	-	-	-
DTMF?	X	-	-	X
MMIO2. Does the system support text and / or graphic input (e.g. via the WWW)?	X	-	-	-
MMIO3. Is a mix of the above possible?	X	NA	NA	NA
MMIO4. Is the interaction telephone-based?	-	X	-	X

Discussion

The exemplar systems clearly show the distinction between what is possible, i.e. natural language processing that allows user directed dialogues, and what is marketable, i.e. finite state dialogue systems which are turn-taking based. Where the NLP systems differ is on their provision of additional media to supplement speech input. Waxholm (W) uses a textual/graphical display to represent the results of speech queries. Furthermore, an animated face simulates a communication partner which can give non-verbal queues for turn-taking. The selection of non-speech output was based on the linear nature of speech in displaying tables, making it difficult for the user to randomly search the query results. Also, there was too much information to be played (verbally) to the user. This represents an important stage in SLDS design, identifying whether the verbal/aural modality is the right choice for the tasks and task-related information which is inherent in the target domain (Bernsen 1997).

The visual display in Waxholm is also used to signal to the user what state the system is in. If, for example, a query of the timetable database does not yield a result within a specified timeframe, an animated face on the monitor signals that through a 'hmmm' sound.

The finite state dialogue system (Operetta) differs on the complexity of its speech recognition capability rather than dialogue management or modelling of linguistic phenomenon. The dialogue task for this system is relatively simple in that it only requires to capture (and understand) one piece of information, the name of the person the caller wants to get connected to. In contrast to the other three systems user utterances are extremely constrained to isolated phrases only containing the first and second name of the person the user wants to speak to. An alternative to speech input is provided through pressing keys on the telephone keypad (DTMF). However, this functionality is never announced to the user. Waxholm offers the possibility to use a special button to interrupt system output.

Verbmobil is the only system that uses prosody interpretation (SS5). This is used along with syntactic and semantic parsing to determine if an utterance was a question (shown by rising intonation) or a statement (level intonation). The variant of the system evaluated is not accessible through the telephone, but rather uses a good quality microphone for speech input.

Verbmobil and the Danish Dialogue System both provide command-like words which can be used at any stage in the dialogue and have a context independent function, e.g. 'repeat' will cause the last system utterance to be repeated. Waxholm attempts to interpret such words in terms of the current dialogue context.

Key: **WAX**-Waxholm, **DD**-Danish Dialogue System, **VM**-Verbmobil, **OP**-Operetta

5.2 Output to User

Attribute	**WAX**	**DD**	**VM**	**OP**
Channel (C)				

C1. Are non-verbal sounds (e.g. music, background noise, auditory icons) used by the system?	X	-	-	X
C2. Does the system make use of engineered sounds (e.g. earcons)?	X	-	-	-
C3. Is text and graphical output possible?	X	-	-	NA
C4. Does the system support a mix of outputs (e.g. speech and text concurrently)?	X	NA	NA	NA
C5. Is synthesised speech used?	X	X	X	-
C6. Was any evaluation of speech output quality carried out?	?	?	?	-
Signalling (SG)				
SG1. Can the system be interrupted?	X	-	X	-
SG 2. Does the system make use of a beep to prompt users?	-	-	-	X
SG3. Does the system use an explicit prompt structure (i.e. menu structure)?	-	-	-	X
SG4. Is a command structure supported (e.g. flat hierarchy)?	X	X	X	-
SG5. Is there use of programmable intonation patterns in system output?	X	-	-	-
SG6. Is a curt/denatured style used?	X	X	X	-
SG7. Does the system use a transactional mode?	X	X	-	-
SG8. Or an interactional mode?	-	-	X	X
SG9 Is a form filling/frame-based dialogue structure used?	X	X	-	-
Help (H)				
H1. Is a help facility available?	?	?	?	-
H2. What kind of help facilities are used?	?	?	?	NA
Single shot?				
Incremental?				
Context Dependent?				
H3. Is it possible to change the input modality?	?	NA	-	NA
H4. Are prompts worded such that it is possible to interrupt them before everything is said (i.e. semantically valid cut-off points)?	?	?	?	NA
H5. Is an human operator backup provided?	-	X	?	X
H6. Is additional system information required, end-user guide?	X	?	X	-
H7. Are escape routes out of the dialogue available?	X	X	X	X
H8. Are explicit confirmations used?	?	X	?	X
User Model (UM)				
UM1. Are user preferences reflected in the dialogue?	- X	- X	- X	- -
UM2. Is a interactional history maintained?	?	?	?	NA

UM3. Is a change of dialogue history possible?				
Dialogue Style (DS)				
DS1: Is the dialogue driven by the system?	X	X	X	X
DS2: By the user?	X	X	X	-
DS3: Mix	X	X	X	-
Novice and Expert (NE)				
NE1: Does the dialogue explicitly cater for novices and experts?	?	X	NA	X

Discussion

As mentioned before, Waxholm is the only system that makes use of meta-verbal utterances (C1 - "umm", "oh!") to signify lengthy processing or change of topic.

Waxholm uses a frame-based (SG9) user model in the form of a history of what the system has understood from previous user utterances. This includes the recognised topic (discourse context), e.g. time table enquiry, existence enquiry, valediction. This information is used in determining the next stage in the dialogue.

All the research systems use dialogues which are open-ended in that the user can provide all or some of the pieces of information required to generate a valid database query. The system takes the initiative in order to obtain necessary information items, e.g. a destination for a travel enquiry. It must be stressed that this system initiative may be in response to a misrecognised user utterance, i.e. the user said a destination but this was not interpreted as such by the system.

Operetta and the Danish Dialogue System make use of a human backup which the user can go to for help at any time (in Operetta, for instance, this is by pressing 0 or saying 'operator'). Operetta does not allow the interruption of system output through speaking and it is therefore necessary to a have long and precise introduction to how the system should be used ("Good morning. You are speaking to the Vocalis automatic switchboard. After the tone please clearly say the first and last name of the person you want and I will connect you. If you prefer to speak to the operator please stay on the line."). However, and this information is never given to the user, it can be interrupted by pressing the '*' key on the telephone keypad. The Danish Dialogue System does not offer redirection to a human operator but provides a phone number which the user may call in order to speak to a human.

5.3 User Description

The following checklist lists all user criteria which should be explicitly addressed during a system design. A cross means the issue has been addressed positively, blank negatively, a question mark shows no consideration was given to this issue.

Key: **WAX**-Waxholm, **DD**-Danish Dialogue System, **VM**-Verbmobil, **OP**-Operetta

Attribute	WAX	DD	VM	OP
Group (UG)				
UG1. For which group(s) does the system provide an interface:				
Agents?	X	X	-	X
Caller/User?	X	X	X	X
System Administrators?	X	X	-	X
Experience (UE)				
UE1. Is the system designed for naive / novice users?	X	X	?	
UE2. Is the system designed for expert users?	-	X	?	
Frequency (UF)				
UF1. What frequencies of use are mainly supported by the system:				
Often?	-	X	X	X
Occasional?	X	X	-	X
UF2. Does this apply to the whole application?	?	X	?	X
UF3. Only to certain Functions?	?	?	?	-
Domain Description (DD)				
DD1 Was this studied before system design?	X	X	X	X
DD2 Are the tasks transactional?	X	X	X	X
DD3 Is information provided from a database?	X	X	X	X
System Knowledge (SK)				
SK1. Is knowledge of SLDSs required?	X	-	X	-
SK2. Is knowledge of previous service (e.g. agent based system) required?	X	?	X	X
SK3. Do first time callers receive guidance from the system?	X	X	?	X
Domain Knowledge (DK)				
DK1. Is knowledge of application domain required?	X	X	X	X
Use Location (UL)				
UL1. Where is the system to be called from?/used				
Home	X	X	-	X
Work	X	X	X	X
Mobile, e.g. car, train	-	X	-	X
UL2. Are the users subscribers?	-	X	X	-
UL3. Are the users paying premium rates for the service?	NA	NA	NA	NA
Language/Culture (LC)				
LC1. Does the system cater for different	?	-	X	-

languages,? LC2. Are different dialects supported? LC3. Is the dialogue dependent on the user's gender? LC4. Were cultural dependencies identified?	? - -	X - -	? - -	X - X

Discussion

There was little consideration given to how frequency of use may vary between different system functionalities (UF2/3). Also, additional support for the first time user was only provided in Waxholm with the animated face describing the basic functionality of the system and in the Danish Dialogue System. Operetta provides only a very limited service in which one item of information has to get collected from the user. There is no information collected about the caller and it is therefore not possible to determine whether the user has used the system before.

Only Verbmobil addressed different languages due to its basic functionality of providing anglo-germanic, anglo-japanese translations.

Operetta addressed cultural issues (LC4) when it was sold in the USA. The introductory prompt was shortened and the Americanisation of the prompts was carried out.

Only Operetta can be configured to use non-verbal sounds to signal dialogue states. While Operetta tries to connect a caller, music is played to the user.

6. Project Life-Cycle Checklist

Each exemplar will now be discussed in terms of their design life-cycle. The checklist provides a best practice list of life-cycle steps; some may not be relevant to a particular exemplar (shown by NA).

Key: **WAX**-Waxholm, **DD**-Danish Dialogue System, **VM**-Verbmobil, **OP**-Operetta

6.1 Context Analysis

Attribute	WAX	DD	VM	OP
User Study (US)				
US1: Representative user defined?	-	X	-	X
US2: Personal details?	-	-	-	-
US3: Experience considered?	-	X	-	X
US4: Language/dialectical issues considered?	-	X	X	X
US5: Knowledge defined; domain, system, general?	X	X	X	X
US6: Organisational position?	NA	-	X	X
US7: Job/role of users considered?	-	X	X	X
US8: Studies formally documented ?	-	X	-	X
Domain/Environment Study (DE)				
DE1: Working environment description, e.g. ambient noise etc. ?	X	-	-	X
DE2: Organisational effects of system considered	NA	NA	-	X

Discussion

The analysis of the intended users was carried out at a high level in all of the systems. Waxholm was preceded by two informal studies, firstly the domain was studied by interviewing agents in the Waxholm booking office, secondly, members of the Waxholm team were quizzed for initial utterances they would use to find out ferry information. Neither study was formally documented.

Neither Waxholm nor Verbmobil used any study of the target organisational environment they would sit in (DE), e.g. Waxholm in a booking centre, Verbmobil in an office environment. Whilst these were both research systems, the target environment can have an important effect on the chosen design. The Danish Dialogue System would have little organisational impact since it was envisaged it would be used over the telephone.

The development of Operetta involved an analysis of typical organisations (DE2). A methodology was designed which involved interviewing key users within a given

organisational context. Also possible effects of background noise on the recognition performance were evaluated.

6.2 Requirements Capture

Attribute	WAX	DD	VM	OP
Task Analysis (TA)				
TA1: Vocabulary definition done?	X	X	X	X
TA2: Core task description done?	X	X	X	X
TA3: Task description formalism used?	X	X	-	-
Usability Measures (UM)				
UM1: Quantitative measures identified?	-	X	?	-
UM2: Qualitative measures ?	X	X	-	X
Prototyping (P)				
P1: Pen&paper WOZ	-	-	X	X
P2: WOZ test?	X	X	-	X
P3. Were subjects aware there was a wizard?	X	-	-	-
P4. Use of WOZ recordings as speech corpus	X	-	-	X
P5. Use of WOZ recordings to refine dialogue design?	X	X	-	X
P6 Were walk-through techniques used?	X	-	-	-
P7. Subjective evaluations carried out?	X	X	-	X
P8. Evaluation of prototypes documented?	-	X	-	-
P9. Prototypes thrown away?	-	X	-	-
P10. Documents prototyped?	X	-	X	X
P11. Did evaluations involve subjects using scenarios?	X	X	-	X
P12. Were the experimenters aware that scenario wording may influence user utterance wording?	X	X	-	X

Discussion

The WOZ methodology was fundamental to the DD system design and evaluation (Pn). Waxholm did not use a simple prototype, rather a near complete system minus the recogniser was built and then evaluated. Verbmobil used a pen and paper study of negotiation tasks (P1). Fairly comprehensive and regular evaluations were carried out during the development and installation of Operetta. Evaluations on 'life' systems are now carried out when a problem occurs.

6.3 Evaluation and Field Trials

Attribute	WAX	DD	VM	OP
Evaluation Description (ED)				
ED1. When				
During requirements capture	-	X	-	X
In iterations after implementation	-	X	-	X
Once on implementation	X	-	X	X
ED2. Who with?				
In-house	X	X	X	X
Friendly client	-	X	-	X
Cold client	-	-	-	-
ED3. Size				
Alpha	X	-	X	X
Beta	-	-	-	-
EM1. Evaluation Methods				
WOZ	X	X	-	-
Cognitive walk-through	-	X	-	-
Heuristic	-	X	-	X
Statistical	-	-	X	X
ET1. Evaluation Type				
Performance	X	X	X	X
Diagnostic	-	X	-	X
Subjective (adequacy)	-	X	-	X

Discussion

The Danish Dialogue system used extensive heuristic evaluations at all stages (EM1) based on co-operativity guidelines (see discussion on product life-cycles). Waxholm relied on informal scenario-based evaluation once it was substantially completed (minus the recogniser)

Operetta predominantly used beta-site evaluations which focused on quantitative measures such as recognition accuracy. Some subjective evaluations were carried out.

Usability evaluation of SLDSs is in need of additional methodological guidelines and tools. Transaction success evaluation is a necessary but far from sufficient instrument in evaluating the quality, as seen from the user's point of view, of SLDSs. Use of the co-operativity guidelines which are being further developed in DISC, could become another important tool for early evaluation and error correction of flaws in usability due to bad system dialogue design. Still, user questionnaires and interviews remain important to SLDSs user evaluation. The problem with these methods, and especially with questionnaires, is that there are no guidelines for designing and interpreting them with special reference to SLDSs. It is easy to get the user score a series of questions about system friendliness, robustness, etc., but it is not

known which questions are the "right" ones to ask and how the results of user answers to a series of questions should be scored overall. One promising way forward is to attach usability research to field deployment of advanced SLDSs, such as the work done in the Netherlands in connection with the launch, in January 1998, of the Dutch train time-table information system which is basically similar to the German and Swiss systems. An interesting possibility is to "map back" the results of the user evaluations done in the field, onto the usability evaluations done prior to field deployment. This might enable a better understanding of which early usability test methods attain the highest conformance with field trials.

7. Conclusions

This paper has provided an overview of the Human Factors-related aspects of commercial and research spoken-dialogue systems. Those areas of particular interest have been elucidated and placed in the context of an engineering life-cycle for interactive system development. In order to provide an indication of the state-of-the-art, both individual aspects and life-cycle best practice were used to evaluate a number of commercial and research systems. From this analysis the following deficiencies were identified in current practice.

Documentation

- There was little formal documentation at any stage of the design and implementation process except for Danish Dialogue System.

Ethnology

- Little consideration given to organisational effects of systems and how to design for these.
- Few real users were used in evaluations.
- Context analysis was limited**.** The reason given for this in Waxholm and Danish Dialogue system was that the goal of the project was to research some aspect of the system, e.g. speech recognition, rather than meet user needs. The Danish Dialogue System was a little bit different in that it focused on Human Factors. However, since the systems carried out evaluations with real subjects, it seems possible that results for the component under study will be confounded by the negative impact of a system ill-fitted to its context.

Help

- In NLP systems there was little explicit help capability. This could be in the form of example dialogue flows or typical utterances.

The next step is to characterise how these deficiencies can be overcome, and provide a comprehensive description of a realistic best practice for future system developments. These will be the next stages of the DISC project.

8. Acknowledgements

Many thanks go to the DISC partners who have demonstrated their systems in order to provide information for the exemplar analysis grids.

9. References

Bertenstam, M. (1995). "The Waxholm system - a progress report." In *Proceedings of the ESCA Tutorial and Research Workshop on Spoken Dialogue Systems*. Vigsø,

Proceedings of Eurospeech '93, Berlin, 1993, pp. 1867-1870.

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

Bernsen, N. O., L. Dybkjær and H. Dybkjær (1998) "Designing Interactive Speech Systems - From First Ideas to User Testing". Springer Verlag 1998.

Blomberg, M.(1993). "An Experimental Dialogue System: Waxholm". In *Proceedings of Eurospeech '93*, Berlin, 1993, pp. 1867-1870.

Brewster, S. A., P. C. Wright, A. D. N. Edwards (1994) "The Design and Evaluation of an Auditory Enhanced Scrollbar". *In Proc. of SIGCHI'94. ACM Press. pp 173-179.*

Carlson R. (1994). "Recent developments in the experimental "Waxholm" dialog system". In *Proceedings of the ARPA Human Language Technology Workshop*. 1994.

Cheepen, C. (1996) "Designing advanced voice dialogues - what do designers do and what does this mean for the future?", *http://www.soc.surrey.ac.uk/research/reports*

Cheepen, C. and Monaghan, J. (1997) "Dialogue Design: Confirmation in Transactional Telephone Dialogues and the Problem of Yes and No". In Proc. of 1st International Workshop on Human Communication, Bellagio, Italy. July.

del Galdo, E. M.; J, Neilson, 1996. "International User Interfaces", (New York: Wiley).

Dutton, D, C. Kamm, S. Boyce (1997) "Recall Memory for Earcons" *Proc. of EuroSpeech'97.*

Falzon, P. (1990) "Human-computer interaction: Lessons from human-human communication" In *Cognitive Ergonomics, Understanding, Learning and Designing: Human Computer Interaction*, Academic Press Ltd.

Falzon, P. (1985) "The Analysis and Understanding of an Operative Language", in B. Schackel (ed), *Proc. of InterACT'85,* pp 437-441.

Franzke, M. (1997) "Is Speech Recognition Usable? An Exploration of a Speech based Voice Mail System" In SIGCHI Bulletin, Vol. 25(3), ACM Press, pp 49-51.

Gibbon Dafydd, Roger Moore and Richard Winski (1997). "*Handbook of standards and resources for spoken language systems*." Mouton de Gruyter. Berlin, New York.

Gilbert, N., D. M. L. Williams and C. Cheepen (1997) "Guidelines for Advanced Spoken Dialogue Design" *To Appear.*

Hone, K., C. Baber (1995) "Using Simulation to Predict the Transaction Time Effects of Applying Alternative Levels of Constraint to User Utterances with Speech Interactive Dialogues" *In Proc. of the ESCA Workshop on Dialogue Systems, Vigo, .pp 209-212.*

Jack, M (1996) "An Investigation of Menu Strategies" Report of Dialogues 2000 Projects, CCIR, Edinburgh University.

James, F. (1997)"AHA: Audio HTML Access" In Proc. of 6th WWW Conference, http://www6.nttlabs.com/HyperNews/get/PAPER296.html

Poltrock, S. E. and J. Grudin. (1995). "Organisational Obstacles to Interface Design and Development: Two Participant User Studies". *In Human Computer Interface Design, (M. Rudisill, C. Lewis, P. B. Polson, T. D. McKay, Eds.z), (SF, Calif.: Morgan Kaufman*), pp. 303-337.

Resnick, P (1992) "Skip and Scan: Cleaning up Telephone Interfaces" *In Proc. ofSIGCHI'92, ACM Press, pp 419-426.*

Roast, C. and P. Siddiqi (1997) Using the Template Model to Analyse Directory Visualisation" (1997) *Interacting with Computers, Vol. 9(2), Elsevier: Holland, pp-172.*

Williams, D. M. L. and C. Cheepen (1998) "Just Speak Naturally: Design for Naturalness in Spoken Dialogue Prompts. *To Appear in CHI'98 Conference Companion.*

Yankelovich, N (1997) "Using natural language dialogues as the basis for speech interface design", in Luperfoy, S. (ed), *Automated Spoken Dialogues*, MIT Press

Yankelovich, N., (1995) "Designing speech acts: Issues in speech user interfaces", in *Proceedings of SIGCHI '95*, ACM Press.

Zoltan-Ford, E., (1991) "How to get people to say and type what computers can understand", *Int. Journal of Man-Machine Studies (34),* Academic Press Ltd, 527-547.

10. Appendix

10.1 Exemplar System Descriptions

10.1.1 The Danish Dialogue System

The Danish Dialogue System is a research prototype for domestic flight ticket reservation. The system runs on a PC and is accessed over the telephone. It is a speaker-independent continuous speech understanding system which speaks and understands Danish with a vocabulary of about 500 words. The prototype runs close to real-time.

10.1.2 WAXHOLM

Waxholm is a multimodal prototype boat traffic and accommodation information demonstrator which gives information on boat traffic in the Stockholm archipelago. It references timetables for a fleet of some twenty boats from the Waxholm company which connects about two hundred ports. Besides dialogue management, speech recognition and synthesis, the system contains modules that handle graphic information such as pictures, charts, etc. The speech recognition component handles speaker independent continuous speech with a vocabulary of approximately 1000 words.

10.1.3 Verbmobil

Verbmobil is a large R&D project sponsored by the German Federal Ministry of Science, Technology and Research. The aim is to develop a speaker independent, spontaneous spoken language translation support system for German/English and Japanese/English human-human negotiation dialogues. The first phase is used in meeting negotiation. The system translates into English and includes a discourse model for anaphora resolution. A vocabulary of about 2000 words is in place. The second phase will involve a more complex tasks.

10.1.4 Operetta

A commercially available product from Vocalis price-routing application for small to medium sized business. Operetta uses speaker-independent recognition with confidence levels to allow callers to say the name of a person they want to speak to. Operetta has a vocabulary of up to 100 names.

11. Operetta Grid Questions

DISC partner: MIP

Authors: Laila Dybkjær and Niels Ole Bernsen

11.1 Introduction

This paper presents an analysis of human factors in the Operetta system. The analysis is presented in the form of (a) a 'grid' which describes the system's properties with particular emphasis on human factors and evaluation results, (b) a life-cycle model which provides a structured description of the system's development and evaluation process, and (c) supporting material, such as system architecture, example screen shots, dialogues and scenarios.

The presented information will be cross-checked with the developers of Operetta as well as with the complementary descriptions of other aspects of Operetta provided by the DISC partners. These other descriptions address speech recognition, done by KTH, language understanding and generation, done by IMS, and system integration, done by LIMSI.

Demonstrator: Available in Cambridge

Developer: Vocalis

Contact: David Williams

Human factors of the Operetta system

The Operetta system is a commercial call answering and routing system (an "Operator's assistant"). Another part of Operetta is a touch-tone driven voice mail system (ignored in what follows). In what follows, we shall also ignore the system's interface to the call recipients. The system was developed by Vocalis, UK.

URL: http://www.vocalis.com/pages/products/operetta.htm

System performance	
Cooperativity	System prompts were originally based an a typical receptionist. However, the greeting message, i.e. Welcome to the company X automatic switchboard, has been changed on a beta-site by beta-site basis. DW has been involved in proof reading ideas from clients but there has been no formal evaluation. The latest incarnation of a short prompt can be hear on the Vocalis Operetta now. In general prompts are suggested by DW and then changed on an ad hoc basis as a function of the client's feedback.
Initiative	Domain communication: system-directed. The user is asked for a name. No further automated help is offered if the caller stays silent, they are simply connected to an available operator. Meta-communication: No meta-communication. Operetta only supports routing operations, i.e. names or DTMF, to a person or the operator.
Influencing users	Explicit and implicit user instructions; walk-up-and-use system. The system's introduction presents the system and instructs the user to clearly say the first and last name of a person, offering operator back-up as an alternative. There are a variety of messages: operator busy: please hold the line or leave a voice mail; operator no answer: leave a voice mail; out of hours (business hours of company defined in the Operetta set up): 'Sorry the company is closed etc.' - say a name (someone you know may still be in the building) or leave a voice mail The system's introduction is optional and is de-selected by keying in the recipient's extension number. Users are clearly instructed to say the first and last name of the person they want to talk to. The opening prompt would be too long if all commands were elucidated - this is a major consideration. It was supposed people would learn of the 'star' to speak option through word of mouth, in training etc. Around 70% of people say the right thing. This was evaluated formally when Operetta was first put on the Vocalis switch.
Real-time	The system responds in real time.
Transaction success	Transaction success=callers getting the person they wanted with the Operetta/Operator hybrid (Operetta is marketed as the Operator assistant, not replacement) is 100%!
General evaluation	Confirmative feedback is used (ISO 9241-10)
Speech input	
Nature	Continuous; speaker-independent; accent, age and gender independent; English.
Device(s)	1. Telephone. 2. Tape recorder for recording the names input by the

	caller. The caller's own name is replayed to the recipient of the call. And so is the recipient's name for use if recognition has difficulties. The tape recorder also takes messages.
Phone server	-
Acoustic models	Strings of phonemes constituting phrases which make up the names of the persons in the organisation.
Search	-
Vocabulary	The system currently has a vocabulary of 100 names + the word "yes". Transcriptions for these can be added and deleted at any time. The same vocabulary is active at all points, e.g. if a recognised name has been disconfirmed by the caller it still plays a part in the next recognition.
Barge-in	Only via DTMF by keying in the extension number.
Word hypotheses	The recogniser produces a best recognition with attached confidence score.
Grammar	There is no grammar in the speech recogniser.
Prosody	The system does not process input prosody.
Speech output	
Device(s)	Telephone.
Language(s)	English.
Input	Files with pre-recorded speech.
Lexicon	Approx. 100 pre-recorded names + the company roles of these people. No resources are shared between the speech synthesis module and other modules in the system.
Sound generation technique	Coded.
Prosody	No prosody processing has been included.
Voice character	Normal human voice; English; male or female (chosen by the system administrator). The voice has changed from mail to female due to client demand.
Pronunciation description units	-
Flexibility	-
Miscellaneous	-
User utterances	
Lexicon	Approx. 100 words (person names + "yes").
Grammar	None needed.
Parsing	None needed.

Style	Extremely terse: a name or “yes”.
Semantics	No semantic representation is being built (or needed).
Discourse, context	-
System utterances	
Generation	Is part of dialogue management.
Lexicon	Pre-defined pre-recorded names and phrases.
Grammar	None needed.
Semantics	None needed.
Style	Friendly.
Processing	-
Discourse, context	-
Multimodal aspects	
Device	Telephone keyboard.
Non-speech input	In addition to speech, the system accepts DTMF input.
Non-speech output	None.
Role(s)	0 for operator, 'star' to skip greeting, 'hash' to access voicemail. Voice mail is totally DTMF driven.
Evaluation	No evaluation.
Attentional state	
Focus, prior	None. The entire vocabulary is in active memory at any time.
Sub-task id.	The system does not do sub-task identification but requests the sub-task to be done by the user at any time.
Expectations	-
Intentional structure	
Tasks	Operetta picks up a ringing phone, greets the caller, offers operator fall-back, asks who they want to talk with and can ask for the caller’s name. It then rings the called party, and when they answer, it announces the call and puts the caller through - just like a human operator. If the switchboard supports music on hold, the caller will hear music. If the called party does not answer their phone, or the line is engaged, Operetta offers to take a message or to put the caller through to somebody else. The called party can access their messages whenever, and from wherever they choose. If a person has left it is up to the system admin. person to update the name database. If all goes well the caller will not be recognised (OOV) and will be routed to the operator to find out the person has left (again the 100% accurate hybrid system!)
Task complexity	Simple tasks; well-structured. No complexity measures have been applied.

Communication	Domain communication: system directed only. Highly constrained: only single phrases (names) and “yes” are allowed from the users. No explicit or echo system (domain) feedback. System-initiated meta-communication**:** If the recogniser returns a medium confidence, the caller is asked to confirm the recognised name, i.e. "was that <David Williams> " <>denote the name message recorded by David Williams. User-initiated meta-communication: None. Problems: Saying names is tricky. What are the cultural norms, e.g. surname first name, titles, etc. This is a problem for the opening prompts. It is not sufficient just to say 'full-name' as this is ambiguous. Other forms of system communication than domain and meta-communication: Tones signifying the 'Recogniser is listening' event.
Interaction level	Only one level is involved in the system initiated meta-communication. No graceful degradation is being used.
Implementation of dialogue management	The dialogue is a flow chart with recognition calls. Designers used a GUI dialogue tool eventually, though initially a form based tool (awful).
Linguistic structure	
Speech acts	The system does not need to identify speech (or dialogue) acts in the users’ input.
Discourse particles	The system does not need to identify discourse particles in the users’ input.
Co-reference	The system does not need to do co-reference resolution.
Ellipses	The system does not need to do any particular processing of ellipses.
Segmentation	The system does not need to do user turn segmentation.
Interaction history	
Linguistic	The system does not maintain a record of the surface language of the users’ utterances.
Topic	The system does not maintain a record of the order in which topics have been addressed through the interaction.
Task	The system does not maintain a record of the task-relevant information which has been exchanged.
Performance	The system does not maintain a record of the user’s performance during interaction.
Domain model	
Data	Internal phone book including names, staff positions and extension numbers (a maximum of 100). Operetta comes with a system

	administration program which allows one to update the internal phone book.
Rules	There are no rules operating on the domain data.
User model	
Goals	Assumed to be to talk to a staff member of the company.
Beliefs	No user beliefs are handled on-line.
Preferences	No user preferences discovered through earlier interactions are being handled.
User group	No novice/expert distinction has been made.
Cognition	No specific cognitive characteristics of users have been taken into account, such as task load, limited memory, natural "response packages" or limited attention span.
System architecture	
Platform	Processor: Pentium 120MHz. Memory: 64 MByte. OS: SCO Unix 3.2v4.2. Disk: 550 MByte. Telephony: Card: Dialogic D41D. Hardware customisation: Telephony card Dialogic D41D. Depending on how many ports are installed, Operetta can answer up to 4 or 8 calls at the same time. The system consists of a stand-alone unit with its own keyboard and monitor. Operetta simply plugs into four (or eight) spare analogue extensions. There are several possible configurations of Operetta, depending on the switch available, cf. Figure 7.
Tools and methods	Describe the tools and methods used.
Generic	A runtime system (CAGE) provided the dialogue flow framework, steps specified procedurally, flow specified by goto's - distributed (dialogue, telephony, recognition).
No. components	Executable calls telephony and recog. libraries.
Flow	Recogniser returns result, telephony card returns DTMF, switch gives telephony events, e.g. flash hook, engaged tone etc., database returns numbers and extension, global settings, that's it really.
Processing times	N/A

11.2 Operetta dialogue

- Good morning. You are speaking to the Vocalis automatic switchboard. After the tone please clearly say the first and last name of the person you want and I will connect you. If you prefer to speak to the operator please stay on the line.
- Clyde Cox.

- Who is calling please?
- David.
- Please hold on whilst I transfer you.
- I am sorry, there is no answer for that extension. If you would like to leave a message for Clyde Cox after the tone please say “yes”; to speak to someone else please stay on the line.
- No.
- After the tone please clearly say the first and last name of the person you want, and I will connect you. If you prefer to speak to the operator please stay on the line.
- David Williams.
- Please hold on whilst I transfer you.
- I am sorry that extension is busy. If you would like to leave a message for David Williams, Human Factors Consultant, after the tone please say “yes”; to speak to someone else please stay on the line.
- Yes.
- Hey! This is David Williams. You are through to my voice mailbox. I am unable to take the phone right now so you can leave your message and I will get back to you when I can. Thanks very much.
- Hey there, David, it’s David here just leaving a test message to demonstrate the system.
- Thank you for calling Vocalis: We will get back to you as soon as possible. Good bye.

12. Operetta Life Cycle Questions

DISC partner: MIP

Authors: Laila Dybkjær and Niels Ole Bernsen

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

To develop a commercial call answering and routing system with voice mail. The benefits of the system should be no lost calls, calls are answered immediately, the system may off-load an overloaded switchboard operator thereby allowing the switchboard operator to more instantly handle non-routine calls and other tasks, users can just speak a name and do not need to remember extension numbers.

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

Historically, the SR platform specified was a Pentium 486 under UNIX with Dialogic telephony card.

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

The recogniser software was written in C so this was the specified language. The telephony stuff was off-the-shelf, everything else was home made.

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, i.e. under one type of constraint.)*

The system is not intended for one particular customer. The customer may be any organisation with a switchboard.

Other constraints: *(Were there any other constraints on the design process (e.g. on cost, manpower, purchase price, development time, standards conformation).)*

The product had to be competitive with other DTMF-based systems.

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

Basically the voice routing idea was the key idea.

Designer preferences: *(Did the designers impose any constraints on the design which were not dictated from elsewhere (e.g. programming language preferences, development methodology)?)*

The software which provides the run-time engine (CAGE) had its constraints.

Design process type: *(What is the nature of the design process (exploratory research, product development, redesign, other (explain))?)*

Product development.

Development process type: *(How was the system/component developed (e.g. through Wizard of Oz, using existing development methodology x, y, z).)*

Operetta developed from a technology basis, i.e. here is this interesting SI recognition, what can we do with it?. WOZ testing of dialogue was not used.

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

Yes, but not available. DW can filter information from them, see DISC web page.

Development process representation: *(Has the development process itself been explicitly represented in some way? How (e.g. bits and pieces of it can be found in scientific papers, the entire process was carefully documented in semi-formal notation, most of the process has been systematically represented in reports).)*

No, but DW thinks it is fairly representative of a new technology application. Operetta has developed in a piecemeal way.

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

The system is meant to meet organisations' needs of somebody answering the phone immediately (no queue) and no matter at which time people call. The system may also be configured to turn itself on and off to fit in with the customer's business practices. The activation times can be set up and changed by the system administrator. The idea is not to get rid of the switchboard operator but to reduce the workload on him/her.

Functionality criteria: *(Which functionalities should the system/component have (this entry expands the overall design goals, e.g. "allow users to do tasks X and Y", "include barge-in", "real-time"). Note that this entry is more general than, but may partially overlap with, the "grid" properties.)*

The system must allow people calling the organisation to be connected to the person to whom they wish to talk, to be connected to a human operator if the person is not available, or to leave a message even if the operator is not in; the system runs in real time; DTMF interrupt is possible by typing in the extension number of the person one wants to speak to. The caller may also press * during the greeting which allows him to interrupt and go straight to speaking the name of the person he wants. If a caller asks for a department the system can be configured to ring the number of a hunt or ring group, if they are supported by the switch.

Limitations are that not all switches are supported, the phone book can contain a maximum of 100 names, and the system only supports time break recall.

The DTMF driven menus of Operetta voice mail give the following features:

Change your greeting

Change your password

Message facilities:

- Listen to new messages

- Send messages

- Archive messages

- Reply to messages sent from other Operetta mailboxes

- Forward message with comment

Personal and global distribution lists

Local and remote access to voice mail

Messages are time and date stamped

Operetta can store up to 24.000 messages. It has disk space for a total of 15 hours of messages. There is no allocation of message space per person. The maximum message length is 2 minutes. Mailboxes are protected by a four digit password. Messages are not automatically deleted. Each person usually has his own mailbox. However, mailboxes may be shared but only entirely. It is not possible to share only non-personal voice mail.

Usability criteria: *(What are the aims in terms of usability (e.g. usable with no training, usable with training in Y).)*

No training needed for users calling.

Organisational aspects: *(Will the system/component have to fit into some organisation or other, how (e.g. partially replace the switchboard operator, require backup for difficult or incomprehensible queries)?)*

The system will partially replace the switchboard operator. It may extend the switchboard's opening hours to 24 hours per day and it is able to answer up to 8 calls at the same time thereby reducing/removing waiting time for callers.

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

Any organisation that wants to extend the switchboard opening hours, reduce waiting time for callers, and rationalise part of the switchboard task.

Users: *(What are the intended users of the system/component (e.g. users speaking High German, walk-up-and-use users, specialised user group X)?)*

We distinguish four user groups:

1. Any person who calls an organisation which has the system installed is a user; walk-up-and-use users; users must speak English.

2. People working in a company which has the Operetta system installed; training needed, cf. Figure 6.

3. Switchboard operator; training needed, cf. Figure 6.

4. System administrator; training needed, cf. Figure 6.

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

Mostly engineers with a linguist or two. Little HF input during the bulk of its lifetime.

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

In initial version of the system based on text input has been in operation since September 1992. Planned development: Longer than expected due to feedback from beta-trials and general receptivity. A product looking for a market.

Requirements and design specification evaluation: *(Were the requirements and/or design specifications themselves subjected to evaluation in some way, prior to system/component implementation? If so, how?)*

No. No time. Generally, requirements are captured from marketing surveys (what are competitors doing), from beta-trial results and from marketing hunches. There has been some effort to provide a better requirements capture method for configuring Operetta prior to installation. This used scenarios of use of different Operetta functionality.

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

No human factors criteria. Recognition targets set.

Evaluation: *(At which stages during design and development was the system/component subjected to testing? How (describe the methodologies used, e.g. glassbox, blackbox, diagnostic, performance, adequacy, acceptance)? What were the results?)*

The real system was evaluated internally at various stages. External beta-trialling provides the bulk of development advice.

Human Factors testing was used in a limited way once the system was on the Vocalis switchboard. This test analysed what people said after the 'say a name' prompt. The results showed most people said the right thing. The opening prompt was, changed to sound less like an answering machine which caused people to hang up. Testing was done again and showed improvement. The results were not documented. There has been no analysis of the voicemail dialogue.

Recently, particular yes/no recognition has been studied by listening to recordings from beta-sites. Generally, dialogue changes are evaluated on an expert basis by me and then feedback is gained from beta-sites. Beta-sites in turn provide case by case input often in a closely coupled. This will continue until the system is selling in volume, i.e. until we have got it right.

Beta sites involved companies of between 10 and 70 people.

Feedback from users is in terms of questionnaires and word of mouth. Negative comments relate mostly to recognition performance, speed and accuracy. Dialogue problems are mostly related to the opening prompt which is either too long or not informative enough.

The input prompt is shown in Figure 4. Figure 5 shows what the called party hears. If Operetta thinks it knows who the caller wants but is not sure, it will ask the caller "Was that X?". If the system is totally unsure it will ask the caller to hold, then play the switchboard operator a recording of what the caller said, so that the operator can either transparently transfer him/her to the correct extension or ask the caller who s/he wants to talk to.

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?)*

Speech technologists, system integrators and algorithm specialist. Product moved out of R&D into operations but there was still a need for speech specialist participation. Team has expanded to include document write, customer care specialist and trainer.

Problems during development and evaluation: *(Were there any major problems during development and evaluation? Describe these (e.g. problems of collaboration in the team, major delays caused by ?, difficulties in satisfying specification requirement X, developer Y left the team, lack of quality of what was delivered by some in the team).)*

Currently switch integration is a big problem - there are so many. Vocalis is less committed to Operetta so there is less 'engineering' resource available.

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.)*

-new technology - what to do with it?

-new product, making a market

-how to sell it - Focus groups organised

-Beta testing

-Hasn't really sold well enough - What now

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

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

Operetta includes high, medium and OOV confidence levels. Error detection dialogue is used for medium confidence, OOV means a switch to the operator. This may be hidden from the caller, i.e. the operator does the routing manually.

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

Easy to update name DB with transcriptions. System admin. interface allows this. Full training course given to system admin. person.

Portability: *(How easily can the system/component be ported? (e.g. OS dependencies, machine dependencies).)*

Move to NT.

Modifications: *(What is required if the system is to be modified)*

Some modifications can be done by the system administrator, others must be done by Vocalis, e.g. new recordings. Operetta uses pre-recorded speech output. Most prompts are predefined. An exception is the greeting message, e.g. "Welcome to the Vocalis automated switch board....". This will be recorded for each customer to be shorter or longer as required. Additional client specific prompts may be recorded for 'out-of-business hours' announcements. This process is time consuming and it is envisaged that once Operetta moves fully out of the beta-phase it will be sold with predefined prompts (apart from the company name).

Integration with better voicemail systems is planned.

Additions, customisation: *Has a customisation of the system been attempted/carried out (e.g. modification of a part of the vocabulary, new domain/task, etc.)? Has there been an attempt to add another language? How easy is it (how much 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 to enforce that the optimal sequence of update steps is followed (e.g. a menu-driven update interface, etc.)? Comment on any peculiarities from the pov. of best practice.*

The Operetta platform clearly has other uses, anything which requires name identification. Currently a variant is being used as a directory enquiries system. - obviously a much larger vocabulary. The REWARD project uses an Operetta box with Spanish, Danish, Dutch variants.

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

Documentation:

Operetta System Administrator's Guide: How to turn lines on/off; how to respond to silence; activation times for call routing and for voice mail; how to adjust activation times; how to specify business hours; types of transfer (all calls are routed via Operetta, but some options can affect the way a call is received (blind transfer, checked transfer, call screening, voice mail); how to add new names and extensions, including speech recogniser updates. The parameters which can be configured by the system administrator are shown in Figure 1.

Parameter	**Default**
Number of rings before giving up on the Operator	8
Operator's extension number	0
Operator's mailbox number	100
Press 0 for the Operator	on
Action to take when the Operator's phone is engaged	Try another extension
Action to take when the Operator doesn't answer	Try another extension
Alternative extension for the Operator	231
Number of rings before giving up on the night bell	30
Number of rings before giving up	5
Music on hold	on
Call screening	on
Department names	on
Announcing who a call is for	on
Message waiting indication	on
Paging configuration	Full announcement
Phone book's order	By last name
Confirmation of name recognition	off
Ask who do you want to speak to? If confirm fails	off
Say name associated with mailbox	on
Change time and date	Current time and date

Figure 1. A list of the Operetta parameters that can be configured, and their default value.

We have flow charts which show what happens in the following cases:

Extension with no voice mail; call screening on.

Extension with voice mail; call screening on.

Extension with no voice mail; call screening off.

Extension with voice mail; call screening off.

Extension with checked transfer.

Extension with blind transfer.

We have call routing flow charts which describe the call routing depending on how certain parameters (see Figure 2) are configured. And we have a speech routing dialogue flow chart.

Parameter	**Behaviour when ON**	**Behaviour when OFF**	**Default**
CONFIRM NAME RECOG	Always asks "Was that X?" when positive or middle confidence.	Only asks "Was that X?" when middle confidence.	OFF
CONFIRM SECOND CHANCE	After the caller has replied "No" to "Was that X?" asks them to repeat the name of the person they want.	After the caller has replied "No" to "Was that X?" tells the caller they will be transferred by the operator.	OFF
REPEAT NAME RECOG WITH OOV	If the confidence is low (OOV) will ask the caller to repeat the name of the person they want.	If the confidence is low (OOV) will tell the caller to hold and transfer transparently to the operator.	OFF
CALL SCREENING	After Operetta has ascertained the name of the called party, asks the caller for their name.	Does not ask the caller for their name, does not announce name to called party.	ON

Figure 2. Parameters and behaviour when on/off.

The flow charts cover the combinations (call screening is assumed to be on) shown in Figure 3.

Parameter	ON/OFF							
CONFIRM NAME RECOG	ON	ON	ON	ON	OFF	OFF	OFF	OFF
CONFIRM SECOND CHANCE	ON	OFF	ON	OFF	ON	OFF	ON	OFF
REPEAT NAME RECOG WITH OOV	ON	ON	OFF	OFF	ON	ON	OFF	OFF
CALL SCREENING	ON	ON	ON	ON	ON	ON	ON	ON

Figure 3. Parameter combinations covered by flow charts.

Good morning/afternoon/evening you're speaking to the [company] automatic switchboard. After the tone please clearly say [the name of the department or] the first and last name of the person you want and I will connect you. If you prefer to speak to the operator, please stay on the line.

Figure 4. Standard form of the Operetta greeting. It can be changed to suit the customer's circumstances.

A call for [called party] from [caller].

Figure 5. If the system is configured to ask for the caller's name the called party hears the sentence shown in the figure. Otherwise 'from [caller]' is left out. The words in square brackets are replaced by a recording of the caller's voice. The call announcement allows the called party to make call screening, i.e. to interrupt a call announcement and make it appear to the caller as if the phone was not answered.

Session	Duration	Max. attendees
System administrator	2-3 hours	3
Switchboard operator	1_-2 hours	5
User (call routing and voice mail)	1-1_ hours	10

User (call routing)	30-40 minutes	10

Figure 6. Training sessions needed.

Configuration	Description
Front end	All incoming calls go straight to Operetta.
Overflow	Operetta picks up calls when the switchboard operator is busy or does not answer.
Out of hours	Operetta answers calls after the switchboard operator has gone home.

Figure 7. Different configurations of Operetta.