GENERALITY AND OBJECTIVITY
Central Issues in Putting a Dialogue Evaluation Tool into Practical Use

Laila Dybkjær, Niels Ole Bernsen and Hans Dybkjær
The Maersk Mc-Kinney Moller Institute for Production Technology
Odense University, Campusvej 55, 5230 Odense M, Denmark
emails: laila@mip.ou.dk, nob@mip.ou.dk, dybkjaer@mip.ou.dk
phone: (+45) 65 57 35 44   fax: (+45) 66 15 76 97

Abstract

This paper presents a first set of test results on the generality and objectivity of the Dialogue Evaluation Tool DET. Building on the assumption that most, if not all, dialogue design errors can be viewed as problems of non-cooperative system behaviour, DET has two closely related aspects to its use. Firstly, it may be used for the diagnostic evaluation of spoken human-machine dialogue. Following the detection of miscommunication, DET enables in-depth classification of miscommunication problems that are caused by flawed dialogue design and supports the repair of those problems, preventing their future occurrence. Secondly, DET can be used to guide early dialogue design in order to prevent dialogue design errors from occurring in the implemented system. We describe the development and in-house testing of the tool, and present the results of ongoing work on testing its generality and objectivity on an external corpus, i.e. an early corpus from the Sundial project in spoken language dialogue systems development.

1. Introduction

Spoken language technologies are being viewed as one of the most important next steps towards truly natural interactive systems which are able to communicate with humans the same way that humans communicate with each other. After more than a decade of promises that versatile spoken language dialogue systems (SLDSs) using speaker-independent continuous speech recognition were just around the corner, the first such systems are now in the market place. These developments highlight the needs for novel tools and methods that can support efficient development and evaluation of SLDSs.

There is currently no best practice methodology available which specialises software engineering best practice to the particular purposes of dialogue engineering, that is, to the development and evaluation of SLDSs. In June 1997, a European Concerted Action, DISC (Spoken Language Dialogue Systems and Components - Best Practice in Development and Evaluation), will be launched with the goal of systematically addressing this problem. DISC aims to develop a first detailed and integrated set of development and evaluation methods and procedures (guidelines, checklists, heuristics) for best practice in the field of dialogue engineering as well as a range of much needed dialogue engineering support concepts and software tools. The goals of dialogue engineering include optimisation of the user-friendliness of SLDSs which will ultimately determine their rank among emerging input/output technologies. The present paper will present ongoing work on one of the tools that are planned to result from DISC.

It is a well-recognised fact that the production of a new software engineering tool or method is difficult and time consuming. The difficulties lie not only in the initial conception of, for instance, a new tool, or in tool drafting and early in-house testing. Even if these stages yield encouraging results, there is a long way to go before the tool can stand on its own and be used as an integral part of best practice in the field. One central reason why this is the case is the problem of *generalisation.* A tool which only works, or is only known to work, on a single system, in a highly restricted domain of application, or in special circumstances, is of little interest to other developers. In-house testing will inevitably be made on a limited number of systems and application domains and often is subject to other limitations of scope as well. To achieve and demonstrate an acceptable degree of generality, the tool must be iteratively developed and tested on systems and application domains, and in circumstances that are significantly different from those available in-house. Achievement of generality therefore requires access to other systems, corpora and/or development processes. Such access is notoriously difficult to obtain for several reasons, including commercial confidentiality, protection of in-house know-how and protection of developers' time. A second reason why software engineering tool or method development is difficult and time consuming is the problem of *objectivity.* It is not sufficient that some method or tool has been trialled on many different cases and in widely different conditions. It must also have been shown that different developers are able to use the new method or tool with approximately the same result on the same corpus, system or development process. The benefits from using a new tool or method should attach to that tool or method rather than to its originators.

Prior to the start of DISC, we have developed and tested a tool for dialogue design evaluation on an in-house SLDSs project (Bernsen et al. 1996, Bernsen et al. 1997a). The paper will present first test results on the generality and objectivity of this tool called DET (Dialogue Evaluation Tool). Building on the assumption that most, if not all, dialogue design errors can be viewed as problems of *non-cooperative* system behaviour, DET has two closely related aspects to its use. Firstly, it may be used as part of a methodology for diagnostic evaluation of spoken human-machine dialogue. Following the detection of human-machine miscommunication, DET enables in-depth classification of miscommunication problems that are caused by flawed dialogue design. In addition, the tool supports the repair of those problems, preventing their occurrence in future user interactions with the system. Secondly, DET can be used to guide early dialogue design in order to prevent dialogue design errors from occurring in the implemented system. The distinction between use of DET for diagnostic evaluation and as design guide mainly depends on the stage of systems development at which it is being used. When used prior to implementation, DET acts as a design guide; when applied to an implemented system, DET acts as a diagnostic evaluation tool. In what follows, we describe the development and in-house testing of the tool (Section 2), present ongoing work on testing its generality and objectivity (Section 3), and conclude the paper taking a look at the work ahead (Section 4).

2. Tool Development

DET was developed in the course of designing, implementing and testing the dialogue model for the Danish dialogue system (Bernsen et al. 1997b). The system is a walk-up-and-use prototype SLDS for over-the-phone ticket reservation for Danish domestic flights. The system's dialogue model was developed using the Wizard of Oz (WOZ) simulation method. Based on the problems of dialogue interaction observed in the WOZ corpus we established a set of guidelines for the design of cooperative spoken dialogue. Each observed problem was considered a case in which the system, in addressing the user, had violated a guideline of cooperative dialogue. The WOZ corpus analysis led to the identification of 14 guidelines of cooperative spoken human-machine dialogue based on analysis of 120 examples of user-system interaction problems. If those guidelines were observed in the design of the system's dialogue behaviour, we assumed, this would increase the smoothness of user-system interaction and reduce the amount of user-initiated meta-communication needed for clarification and repair.

The guidelines were refined and consolidated through comparison with a well-established body of maxims of cooperative human-human dialogue which turned out to form a subset of our guidelines (Grice 1975, Bernsen et al. 1996). The resulting 22 guidelines were grouped under seven different *aspects* of dialogue, such as informativeness and partner asymmetry, and split into *generic* guidelines and *specific* guidelines. A generic guideline may subsume one or more specific guidelines which specialise the generic guideline to a certain class of phenomena. The guidelines are presented in Figure 1.

The consolidated guidelines were then tested as a tool for the diagnostic evaluation of a corpus of 57 dialogues collected during a scenario-based, controlled user test of the implemented system. The fact that we had the scenarios meant that problems of dialogue interaction could be objectively detected through comparison between expected (according to the scenario) and actual user-system exchanges. Each detected problem was (a) characterised with respect to its *symptom*, (b) a *diagnosis* was made, sometimes through inspection of the log of system module communication, and (c) one or several *cures* were proposed. The 'cure' part of diagnostic analysis suggests ways of repairing system dialogue behaviour. The diagnostic analysis may demonstrate that new guidelines of cooperative dialogue design must be added, thus enabling continuous assessment of the scope of DET. We found that nearly all dialogue design errors in the user test could be classified as violations of our guidelines. Two *specific* guidelines on meta-communication, SG10 and SG11, had to be added, however. This was no surprise as meta-communication had not been simulated and thus was mostly absent in the WOZ corpus.

3. Generalising the Tool

As pointed out in Section 2, success in early tool development is not enough if the aim is to be able to recommend the tool to other SLDS developers on a solid basis. The early development phase focused on one SLDS with one particular dialogue structure, in one particular domain of application, designed for a particular type of task, i.e. reservation, in one particular development phase, i.e. evaluation of an implemented system, and in circumstances of controlled user testing where we had available the scenarios used by the subjects as well as the full design specification of the system. To test and increase the generality of the tool, we are currently applying DET as a dialogue design guide to a WOZ corpus from the Sundial project (Peckham 1993).

Ideally, testing DET on the Sundial corpus will increase the generality that can be claimed for the tool in four different ways: (1) the *system dialogue* is different from that of the Danish dialogue system; (2) the *task type* is different, i.e. information vs. reservation; (3) the *test type/tool purpose* pairs are different: whereas in the case of the Danish dialogue system, DET was used for diagnostic evaluation in a controlled user test, the tool is being used as an early dialogue design guide in the case of Sundial; and (4) *circumstances* are different because

Dialogue Aspect	**GG No.**	**SG No.**	**Generic or Specific Guideline**
Group 1: **Informativeness**	GG1		*Make your contribution as informative as is required (for the current purposes of the exchange).
		SG1	Be fully explicit in communicating to users the commitments they have made.
		SG2	Provide feedback on each piece of information provided by the user.
	GG2		*Do not make your contribution more informative than is required.
Group 2: **Truth and evidence**	GG3		*Do not say what you believe to be false.
	GG4		*Do not say that for which you lack adequate evidence.
Group 3: **Relevance**	GG5		*Be relevant, i.e. be appropriate to the immediate needs at each stage of the transaction.
Group 4: **Manner**	GG6		*Avoid obscurity of expression.
	GG7		*Avoid ambiguity.
		SG3	Provide same formulation of the same question (or address) to users everywhere in the system's dialogue turns.
	GG8		*Be brief (avoid unnecessary prolixity).
	GG9		*Be orderly.
Group 5: **Partner asymmetry**	GG10		Inform the dialogue partners of important non-normal characteristics which they should take into account in order to behave cooperatively in dialogue. Ensure the feasibility of what is required of them.
		SG4	Provide clear and comprehensible communication of what the system can and cannot do.
		SG5	Provide clear and sufficient instructions to users on how to interact with the system.
Group 6: **Background knowledge**	GG11		Take partners' relevant background knowledge into account.
		SG6	Take into account possible (and possibly erroneous) user inferences by analogy from related task domains.
		SG7	Separate whenever possible between the needs of novice and expert users (user-adaptive dialogue).
	GG12		Take into account legitimate partner expectations as to your own background knowledge.
		SG8	Provide sufficient task domain knowledge and inference.
Group 7: **Repair and clarification**	GG13		Enable repair or clarification meta-communication in case of communication failure.
		SG9	Initiate repair meta-communication if system understanding has failed.
		SG10	Initiate clarification meta-communication in case of inconsistent user input.
		SG11	Initiate clarification meta-communication in case of ambiguous user input.

Figure 1. Guidelines for cooperative system dialogue. GG means generic guideline. SG means specific guideline. The generic guidelines are expressed at the same level of generality as are the Gricean maxims (marked with an *). Each specific guideline is subsumed by a generic guideline. The left-hand column characterises the aspect of dialogue addressed by each guideline.

we do not have the scenarios used in Sundial and do not have access to the early design specification of the Sundial system. If DET works well under circumstances (4), we shall know more on how to use it for the analysis of corpora produced without scenarios, such as in field tests, or without the scenarios being available.

The important generalisation (4) poses a particular problem of objectivity. When, as in controlled user testing, the scenarios used by subjects are available, it is relatively straightforward to detect the dialogue design errors that are present in the transcribed corpus using objective methods. The objectivity problem then reduces to that of whether different analysers arrive at the same classifications of the identified problems. When, as in many realistic cases in which DET might be used, no scenarios exist or are available, an additional problem arises of whether the corpus analysers are actually able to detect *the same* problems in a dialogue prior to classifying them. If not, DET will not necessarily be useless but will be less useful in circumstances in which the objective number of dialogue design errors matters. In the test case, objectivity of detection will have to be based on the empirical fact, if it is a fact, that developers who are well-versed in using the tool actually do detect the same problems.

4. The Simulated System

The Sundial dialogues are early WOZ dialogues in which subjects seek time and route information on British Airways flights and sometimes on other airline flights as well. The emerging system seems to understand the following types of domain information:

1. Departure airport including terminal.
2. Arrival airport including terminal.
3. Time-tabled departure date.
4. Time-tabled departure time.
5. Time-tabled arrival date.
6. Time-tabled arrival time.
7. Flight number.
8. Actual departure date (not verified).
9. Actual departure time.
10. Actual arrival date (not verified).
11. Actual arrival time.
12. Distinction between BA flights which it knows about, and other flights which it does not know about but for which users are referred to airport help desks, sometimes by being given the phone numbers of those desks.

By contrast with the Danish dialogue system, the Sundial system being developed through the use of the analysed corpus uses a *delayed feedback* strategy. Instead of providing *immediate feedback* on each piece of information provided by the user, the system waits until the user has provided the information necessary for executing a query to its database. It then provides *implicit feedback* through answering the query. Until the user has built up a full query, which of course may be done in a single utterance but sometimes takes several utterances to do - the system would only respond by asking for more information or by correcting errors in the information provided by the user. The delayed feedback strategy is natural in human-human communication but might be considered somewhat dangerous in SLDSs because of the risk of accumulating system misunderstandings which the user will only discover rather late in the dialogue. We would not argue, however, that the delayed feedback strategy is impossible to implement and successfully use for flight information systems of the complexity of the intended Sundial system. Still, this complexity is considerable, in particular, perhaps, due to the intended ability of the system of distinguishing between timetabled and actual points in time. It is not an easy design task to get the system's dialogue contributions right at all times when this distinction has to be transparently present throughout.

Another point about the corpus worth mentioning is that the simulated system understands the user amazingly well and in many respects behaves just like a human travel agent. The implication is that several of the guidelines in Figure 1, such as GG11/SG6/SG7 on background knowledge, and GG13, SG9/SG10/SG11 on meta-communication are not likely to be violated in the transcribed dialogues. It should be added that it is not accidental that exactly these guidelines are not likely to be violated in the transcribed dialogues. The reason is that it is difficult to realistically simulate the limited meta-communication and background-understanding abilities of implemented systems. As to the novice/expert distinction (SG7), this is hardly relevant to sophisticated flight information systems such as the present one. A final guideline which is not likely to be violated in the transcriptions, is SG1 on user commitments. The reason simply is that users seeking flight information do not make any commitments: they merely ask for information.

5. Methodology and Results

The Sundial WOZ corpus comprises approx. 100 flight travel information dialogues concerning British Airways flights. The corpus was produced by 10 subjects who each performed 9 or 10 dialogues based on scenarios selected from a set of 24 scenarios. We do not have these scenarios. The transcriptions came with a header which identifies each dialogue, markup of user and system utterances, consecutive numbering of the lines in each dialogue transcription, and markup of pauses, ahs, hmms and coughs. For the first generality test of DET, we have selected 33 dialogues. Three dialogues were used for initial discussions among the two analysers. The remaining 30 dialogues were split into two sub-corpora of 15 dialogues each. Each sub-corpus was analysed by the two analysers.

Methodologically, we analysed each system utterance in isolation as well as in its dialogue context to identify violations of the guidelines. Utterances which reflected one or more dialogue design problems were annotated with indication of the guideline(s) violated and a brief explanation of the problem(s). Using TEI, we have changed the existing markup of utterances to make each utterance unique across the entire corpus. In addition, we have added markup for guideline violation. An example is shown in Figure 2.

Having independently analysed the two sub-corpora of 15 dialogues each, the analysers discussed each of the 384 claimed guideline violations and sought to reach consensus on as many classifications-by-guideline as possible. This lead to the following 10 *status descriptors* for the claimed guideline violations:

```
<u id="U1:7-1">
    (0.4) #h yes I'm enquiring about flight number bee ay
    two eight six flying in later today from san francisco
    (0.4) could you tell me %coughs% 'scuse me which
    airport and terminal it's arriving at and what time (9)
    %coughs% (2) %coughs%
...
<u id="S1:7-6">
    (10) flight two eight six from san francisco arrives at
    london heathrow terminal four at thirteen ten
<violation ref="S1:7-6" guideline="SG2"> Date not men-
    tioned. The tabled arrival time is probably always the
    same for a given flight number but there may be days
    on which there is no flight with a given number.
<violation ref="S1:7-6" guideline="GG7"> It is not clear
    if the time provided is that of the timetable or the ac-
    tual (expected) arrival time of the flight.
```

Figure 2. Markup of part of a dialogue from the Sundial corpus. The excerpt contains a user question and the system's answer to that question. The user's query was first misunderstood but this part of the dialogue has been left out in the figure (indicated as: ....). The system's answer violates two guidelines, SG2 and GG7, as indicated in the markup.

(id) Identity = The same design error case identified by both annotators.

(c) Complementarity = A design error case identified by one annotator.

(cv) Consequence violations = Design error cases that would not have arisen had a more fundamental design error been avoided.

(us) User symptoms = Symptoms of design errors as evidenced from user dialogue behaviour.

(a) Alternatives = Alternative classifications of the same design error case by the two annotators.

(rc) Reclassification = Agreed reclassification of a design error case.

(rce) Reclassification to already identified case = Agreed reclassification of a design error case as being identical to one that had already been identified.

(ud) Undecidable = Agreed undecidable design error classification.

(deb) Debatable = The annotators disagreed on a higher-level issue involved in whether to classify a system utterance as a design error case.

(rej) Rejects = Agreed rejections of attributed design error cases.

Based on the consensus discussion, the analysers created two tables, one for each sub-corpus. The tables were structured by guideline and showed the violations of a particular guideline that had been identified by one of the

Guide-line	NOB-NOB	Comments	NOB-LD
GG1		ud: 2 different interpretations possible of S8:1-5	S8:1-5
	S8:1-6	id+deb: offer/give phone no.	S8:1-6
		c+deb: offer/give phone no.	S8:3-3
	S8:6-3	c: scheduled not stated	
		c+deb: offer/give phone no.	S8:9-2
	S8:9-4	c: scheduled not stated	
	S9:1-3	c: actual not stated	
	S9:6-2	a: actual not stated + GG7	
		c: S should specify desired information	S9:9-2
	S9:9-3	c: actual not stated	
	S9:10-2	a: scheduled not stated + GG7	
	S10:1-3	id+deb: offer/give phone no. + rc: from SG8	S10:1-3
	S10:1-4	id+deb: offer/give phone no. + rc: from SG8	S10:1-4
		c+deb: offer/give phone no.	S10:1-5
		id+deb: offer/give phone no. + rc: from SG8	S10:1-9
		rej: no need to mention arrival airport	S10:6-2
	S10:6-3	a: failed S clarification + GG5	
		id+deb: offer/give phone no. + rc: from SG8	S10:9-2

Figure 3. Table of claimed violations of GG1. NOB-NOB is NOB's annotation of the NOB sub-corpus. LD-NOB is LD's annotation of that sub-corpus. The table contains 18

cases of which 16 are agreed violations of GG1 (id, c and a), one is undecidable (ud) and one was rejected (rej). The table shows that 4 cases were reclassified (rc), that the two cases of alternative classifications involved GG1 and GG7, and that an agreed classification involved a debate on a component issue (id/c+deb).

two analysers, each violation being characterised, in addition, by its unique utterance identifier, its status descriptor and a brief description (Figure 3).

Of the 384 claimed guideline violations, 344 were agreed upon as constituting actual guideline violations, comprising the status descriptors identity, complementarity, consequence violations, user symptoms, alternatives, reclassification (rc) and reclassification (rce). 40 claimed guideline violations were undecidable, not agreed upon or jointly rejected by the analysers. These figures are not very meaningful in themselves, however, because many identified design guideline violations were identical. This is illustrated in Figure 3 in which the case of *offer/give phone no.* recurs no less than 8 times. The analysers agreed that the system should always offer the phone number of an alternative information service when it was not itself able to provide the desired information, instead of merely telling users to ring that alternative service. The analysers disagreed, however, on whether the system should start by offering the phone number or provide the phone number right away (cf. *deb* in Figure 3). What we need as SLDS developers is not a tool which tells us many times of the same dialogue design error but a tool which helps us find as many different dialogue design errors as quickly as possible. We take this to mean that *when annotating spoken dialogue transcriptions, it can be waste of time and effort to annotate the same design error twice.* A single annotation, once accepted, will lead to a different and improved design. However, if resource limitations enforce restrictions on the number of dialogue design errors which can be repaired, the number and severity of the different dialogue design errors will have to be taken into account.

Guide-line	No. of agreed violations	No. of types
GG1	16+11	6
SG1	Not relevant in information systems	
SG2	6+10	3
GG2	2+1	3
GG3	8+7	1
GG4	1+0	1
GG5	8+5	6
GG6	1+2	2
GG7	7+9	7
SG3	30+39	1
GG8	The system is successful in this respect	
GG9	The system is successful in this respect	
GG10	Massively violated in SG4 and SG5	
SG4	21+18	1
SG5	15+20	1
GG11	The “system” understands	
SG6	too well	
SG7	for these to be violated	
GG12	Violated in SG8	
SG8	6+3	1
GG13	The “system” understands	
SG9	too well	
SG10	for these guidelines to be	
SG11	violated	

Figure 4. Cases and types of dialogue design errors sorted by guideline violated. Note that Figure 4 does not include the cases and types that were either undecidable, disagreed, or rejected (see Figure 5).

Following the reasoning of the preceding paragraph, the analysers proceeded to distil the different types of guideline violations or dialogue design errors identified in the corpus. This led to a much simpler picture, as shown in Figure 4.

Figure 5 shows the nature of the types of guideline violation referred to in Figure 4 as well as the types that were undecidable, disagreed upon or rejected. It should be noted that the term “type” is in this context rather vague. Some of the types of guideline violation in Figure 5 are very important to the design of a habitable human-system spoken dialogue, such as the demand for a more informative opening presentation of what the system can and cannot do, others are of less importance because they appear to be rather special cases, such as when the system offers a phone number to a user who already told the system that s/he had this phone number; some types cover a wealth of different individual cases, such as the many differences in phrasing the same message to the user, others cover just a single case or a number of identical cases; and, of course, some types are more difficult to repair than others. However, common to all these guideline violations is that they should be remedied in the implemented system if at all possible.

Jointly, Figures 4 and 5 show that 15 guideline violation types were found by both analysers, 9 types were found by one analyser only, one type, in fact, a single case, was undecidable on the evidence provided by the transcription, 3 types were disagreed upon, and 6 types were rejected during the consensus discussion. No types were found that demanded revision or extension of the guidelines. The Sundial corpus was analysed by two of the DET tool developers. It cannot be excluded, therefore, that others might in the corpus have found types that demanded revision or extension of the guidelines. This will have to be tested in a future exercise. However, on the evidence provided, the guidelines generalise well to a different *dialogue* and *task type* (cf. Section 3). We also found that the guidelines generalise well to the different *test type/tool purpose* pair of the Sundial corpus. In fact, it is not much different to use the guidelines for early evaluation during WOZ and using the guidelines for diagnostic evaluation of an implemented system. In both cases, one works with transcribed data to which the guidelines are then applied.

Turning now to the objectivity or intersubjectivity of the performed analysis, we mentioned earlier that this raises two issues wrt. the Sundial corpus: (a) to which extent do the analysers identify the same cases/types of guideline violation? and (b) to which extent do the analysers classify the identified cases/types in the same way? During DET development, we never tested for objectivity of annotation.

Agreed Status	Types of Case	Guideline
Found by A1 + A2: identity + alternatives (including consequence violations)	actual arrival/departure not stated	GG1,GG7
	scheduled arrival/departure not stated	GG1,GG7
	failed S clarification	GG1,GG5
	S should offer phone no.	GG1
	no feedback on arrival/departure day, on BA and/or on route	SG2
	missing/ambiguous feedback on time	SG2,GG7
	U: has phone no. S: offers phone no.	GG2, GG5
	departure time instead of arrival time provided	GG5
	phone number provided although user has it already	GG5,GG2
	S: handles all flights - “BA does not handle Airline X.”	GG5
	S: “no flights are leaving Crete today”	GG6,GG7
	scheduled vs. actual arrival/dep. time not distinguished	GG7
	AM and PM not distinguished	GG7
	many variations in S’s phrases	SG3
	too little said on what system can and cannot do: BA often missing, time-table enquiries always missing	SG4, SG8
Found by A1: complementarity (including reclassifications)	S should specify the information it needs	GG1
	S provides insufficient information for the user to determine if it is the wanted answer	GG1
	S repeats more than the 4 phone no. digits asked for	GG2
	“flight info.” known to be false: S knows only BA	GG3
	S: encourages inquiry on airline unknown to it	GG5
	S: “flights between London and Aberdeen are not part of the BA shuttle service, there is a service from London Heathrow terminal one” (rc from GG5 to GG6)	GG6
Found by A2: complementarity + user symptoms (including reclassifications)	U: arriving flights?, S: leaving flights: imprecise feedback	SG2
	system says it is not sure of the information it provided	GG4
	open S intro requires interaction instructions on waiting, verbosity etc.	SG5
Undecidable	2 different interpretations possible of S8:1-5	
Disagreed	whether to just offer or actually give phone no.	GG1
	delayed feedback strategy	SG2
	BA to Zurich: when open meta-communication	GG7

Rejected	no need to mention arrival airport	GG1
	no S-goodbye: U hung up!	SG2
	delayed feedback strategy could defend this case	SG2
	response package OK	GG2
	no BA flights from Warsaw	GG7
	the system needs not have recent events info	SG4, SG8

Figure 5. Cases and types of dialogue design errors sorted by guideline violated.

As to (a), the comparatively high number of guideline violation types found by one analyser but not by the other, i.e. 9 types compared to the 15 types found by both analysers, either shows that we are not yet experts in applying the guidelines to novel corpora, or that the tool is inherently only “62.5 % objective”. This needs further investigation. However, a different consideration is pertinent here. Consider, for instance, analyser A1. A1 found the 15 guideline violation types which were also found by A2 plus another 6 guideline violation types. Compared to these 21 types, analyser A2 only managed to add 3 new guideline violation types. Suppose that, on average, either of two expert analysers find equally many guideline violation types not found by the other. In the present case, this number would be 4.5 guideline violation types. A single expert in using the tool would then find 19.5/24 or 81 % of the guideline violation types found by two analysers together. Still, we don’t know how many new guideline violations a third expert might find and whether we would see rapid convergence towards zero new guideline violations. It would of course be encouraging if this proved to be the case. The 3 types disagreed upon and the 6 rejected types illustrate, we suggest, that dialogue design is not an exact science! Taken together, however, the 4.5 guideline violation types added by the second analyser and the 9 disagreed or rejected types suggest the usefulness of having two different developers applying the tool to a transcribed corpus. Finally, the single undecidable case was one in which the (non-transcribed) prosody of what the user said might have made the difference. Following the system’s statement that “I’m sorry there are no flights leaving Crete today”, the user asked “did you say there aren’t any flights leaving Crete today?” One analyser took the user’s question to be a simple request to have the system’s statement repeated, in which case no guideline violation would have been committed by the system. The other analyser took the user’s question to be an incredulous request for more information (“did you say there AREN’T ANY flights leaving Crete today?”), in which case the system’s subsequent reply “Yes” would have been a violation of GG1.

As to (b), Figure 5 shows that the two analysers produced several alternative classifications. It should be noted, however, that the number of these disagreements has been exaggerated by the data abstraction that went into the creation of a small number of types as shown in Figures 4 and 5. In fact, alternative classifications were only made in 7 cases. It appears to be a simple fact that there will always be data on guideline violation which legitimately may be classified in different ways. Depending on the context, the fact that the system says too little about what it can and cannot do can be a violation of either SG4 or SG8. If it says so up front, this is an SG4 but if it later demonstrates that it has said too little, this should be an SG8 but it is comparatively innocuous if an analyser happens to classify the violation as an SG4. GG1 (say enough) and GG7 (don’t be ambiguous) are sometimes two faces of the same coin: if you don’t say enough, what you say may be ambiguous. Similarly GG1 (say enough) and GG5 (be relevant), may on occasion be two faces of the same coin: if you don’t say enough, what you actually do say may be irrelevant. The same applies to GG2 (superfluous information) and GG5 (relevance): superfluous information may be irrelevant information. SG2 (provide feedback) and GG7 (don’t be ambiguous) may also overlap in particular cases: missing feedback on, e.g., time may imply that the utterance becomes ambiguous. Finally, GG6 (avoid obscurity) and GG7 (don’t be ambiguous) may on occasion be difficult to distinguish: obscure utterances sometimes lend themselves to a variety of interpretations.

6. Conclusion and Future Work

We find the results reported in this paper encouraging. The tool has generalised well to the Sundial corpus and some amount of objectivity has been demonstrated with respect to type identification and classification. As this was out first attempt at using the tool independently of one another, we intend to repeat the exercise using the insights gained. Two times 15 Sundial dialogues will be used for the purpose. Following that, we plan to repeat the experiment with a small sub-corpus of the Philips corpus which comprises 13.500 field test dialogues concerning train timetable information (Aust et al. 1995). This will add a new dialogue and task type, as well as the new circumstances of a field trial to the generality test of the tool. If and when convincing generality and a satisfactory degree of objectivity in using DET have been achieved, a final *transfer* problem must be addressed. This problem concerns how to transfer DET to other developers in some "packaged" form which does not assume person-to-person tuition. This should enable other SLDS developers to quickly and efficiently learn to use DET at the same level of objectivity as has been achieved during the tests of the tool. Only then will DET be ready for inclusion among the growing number of dialogue engineering best practice development and evaluation tools. As a first step in addressing the transfer problem, we have recently included a DET novice in the team. He is an experienced computational linguist but with little experience in SLDS development. We are investigating what it takes to make him an expert in using DET by having him analyse the same Sundial sub-corpus as was reported on above and we hope that he will participate in the planned second Sundial sub-corpus exercise. Following these steps, the final task will be to define an explicit and simple training scheme for how to become an expert in using the tool.

References

Aust, H., Oerder, M., Seide, F. and Steinbiss, V.: The Philips Automatic Train Timetable Information System. *Speech Communication* 17, 1995, 249-262.

Bernsen, N.O., Dybkjær, H. and Dybkjær, L.: Cooperativity in human-machine and human-human spoken dialogue. *Discourse Processes*, Vol. 21, No. 2, 1996, 213-236.

Bernsen, N.O., Dybkjær, H. and Dybkjær, L.: What should your speech system say to its users, and how? Guidelines for the design of spoken language dialogue systems. To appear in *IEEE Computer*, 1997a.

Bernsen, N.O., Dybkjær, H. and Dybkjær, L.: Designing Interactive Speech Systems. From First Ideas to User Testing. Springer Verlag, to appear, 1997b.

Grice, P.: Logic and conversation. In Cole, P. and Morgan, J.L., Eds. *Syntax and Semantics,* Vol. 3, *Speech Acts,* New York, Academic Press, 1975, 41-58.

Peckham, J.: A new generation of spoken dialogue systems: Results and lessons from the SUNDIAL project. *Proceedings of Eurospeech '93*, Berlin, 1993, 33-40.