

Available online at www.sciencedirect.com

Journal of Biomedical Informatics 39 (2006) 482–499

Journal of
Biomedical
Informaticswww.elsevier.com/locate/yjbin

Automatic generation of spoken dialogue from medical plans and ontologies

Martin Beveridge*, John Fox

Advanced Computation Lab, Cancer Research UK, 44 Lincoln's Inn Fields, London WC2A 3PX, UK

Received 1 June 2005

Available online 2 February 2006

Abstract

This paper presents some research undertaken as part of the EU-funded HOMEY project, into the application of intelligent dialogue systems to healthcare systems. The work presented here concentrates on the ways in which knowledge of underlying task structure (e.g., a medical guideline) can be combined with ontological knowledge (e.g., medical semantic dictionaries) to provide a basis for the automatic generation of flexible and re-configurable dialogue. This approach is next evaluated via a specific application that provides decision support to general practitioners to help determine whether or not a patient should be referred to a cancer specialist. The competence of the resulting dialogue application, its speech recognition performance, and dialogue performance are all evaluated to determine the applicability of this approach.

© 2006 Elsevier Inc. All rights reserved.

Keywords: Speech; Dialogue; Ontology; Task; PROforma; Conversational game; Cancer; Healthcare

1. Introduction

A typical approach to representing dialogue, especially in commercial voice-based systems, is the development of a prescriptive dialogue grammar that describes valid sequences of utterances. By employing dialogue grammars, a dialogue management system can be as simple as a graph or finite state machine where each node represents a prompt to the user with a set of options, and the user's response causes a transition to a new node. Such approaches have been particularly useful for systems where the dialogue structure very closely matches the task structure. In particular, since the system always takes the initiative it can restrict the number of options presented to the user and, to an extent, induce a valid user response via the phrasing of the prompts.

To allow some mixed-initiative dialogue, where the user can also take the initiative, extensions to graph systems

have been developed such as frame-based systems. In these entities are defined (e.g., a journey) which have slots to be filled (e.g., departure time, departure location, etc.) and at each node in the graph the dialogue manager has to ensure all mandatory slots are filled. This might be achieved by the system taking the initiative and prompting the user until all information has been gathered, or the user might take the initiative and fill more than one slot at once providing all the relevant information.

In contrast to dialogue grammars and frame-based systems, plan-based approaches to representing dialogue allow for much greater complexity in the dialogue. They take the approach that dialogue is goal-driven and so the aim of the dialogue manager is to infer these goals and respond appropriately. This approach allows for more complex phenomena such as indirect communicative acts where what is meant (illocution) is not the literal interpretation of what is said (locution), e.g., the case where a user asks a train timetable system "can you tell me when the last train to London leaves?" The correct response is for the system to inform the user of the departure time for the requested train and not to answer "yes" or "no."

* Corresponding author. Fax: +44 20 7269 3186.

E-mail address: martinbeveridge@slingshot.co.nz (M. Beveridge).

In order for a dialogue system to be able to reason about goals and their connection to utterances, a model of an agent's 'mental state' is required so that speech acts can be related to these mental states in the conversational participants. The model originally proposed involved describing the configuration of beliefs, desires, and intentions of an agent [1] and is often referred to as the BDI model. However, there are many dialogue phenomena which do not fit into the BDI model: dialogue control phenomena such as acknowledgements, pause-fillers, indicating turn-taking, etc., which maintain the dialogue and coordinate participants. More importantly BDI does not capture the notion of obligations [2] which seem to arise from social convention and include, for example, the fact that if someone asks you a question, it is considered unreasonable not to answer. In fact, speech act theory (and the BDI formulation) only deals with a single utterance and so cannot distinguish between a response to a request, or answer to a question, and a standard declarative used to initiate a conversation [3]. This inability to capture the local context of an utterance, and represent its function given that context, means that there is no way to capture the convention that answers follow questions or that people do not walk away in the middle of a conversation—things that, in fact, can be captured in dialogue grammars by distinguishing grammatical and ungrammatical dialogue structures.

To handle these sorts of problems, Traum and Allen [2] proposed an extension to the BDI model to include 'discourse obligations,' leading to what might be called a BDIO model [3]. Concomitantly, the types of speech that must be described becomes much wider than those described by speech acts, leading instead to the notion of 'dialogue act' [4] which includes both 'core speech acts' (i.e., the original set of speech acts) augmented with so-called 'argumentation acts' (e.g., answer, signal-understanding, utterance failure, etc.). Pulman [3] points out, however, that whilst this begins to capture simple conventions like question–answer it does not address the more general social pressure to respond, such as the hearer giving a reply to a speaker's question which does not constitute an answer but which the hearer hopes will be taken as a relevant response. Describing these more subtle phenomena within the BDI approach, however, probably requires the representation of more complex notions such as politeness and other aspects of human nature and society, which may, in the extreme, require a complete model of human agency.

Another approach, which can be seen as addressing some of the problems of speech act theory [3] is the description of dialogue in terms of conversational games [5–7]. This is primarily a descriptive approach to dialogue rather than a theory of 'rational agency' as the BDI approach is intended to be. For this reason, it circumvents some of the problems encountered by BDI since it starts from the premise of simply trying to describe the facts encountered in real dialogues rather than why they occur. To do this it represents dialogue at two functional levels: at the plan-based level are conversational games which are asso-

ciated with the mutual goals of the participants, and at the structural level are sequences of conversational moves which are intended to achieve those goals [6].

The notion of 'move' employed here extends speech acts to include acts such as reply, acknowledge, clarify, etc. Moves are either initiating moves of games (i.e., rather similar to speech acts) or responding moves. Dialogues are thought-of as being comprised of a series of games each aiming to achieve some sub-goal of the dialogue. Each game itself consists of a series of moves starting with an opening move and finishing with an end move. Importantly, the definition of a game includes moves by both participants, e.g., a request game includes a request by the initiating participant and a reply by the other participant, hence conventional links such as question–answer are captured by using a unit of discourse that spans multiple utterances [3]. The internal structure of a game is typically represented in a similar way to dialogue grammars. For example, a request game may consist of a request move from the speaker, followed by a reply move by the hearer and optionally a final acknowledgement from the speaker to indicate that the information in the reply is grounded. This can be represented as a finite-state network. Additionally, a game can have nested sub-games or a break. Sub-games account for phenomena such as clarifications, side sequences, etc., in which the sub-game contributes to the goals of the parent game. Breaks account for misunderstandings and indicate that either repair is needed to continue, or that the current game may have to be abandoned [6].

The notion of viewing dialogue in terms of games and moves therefore captures the fact that most conversations to achieve a task follow standard scripts (e.g., question–answer) to achieve a limited set of goals (e.g., getting some information, instructing someone) and so generates quite specific expectations regarding a participant's response to a conversational move [8]. At the same time, the recursive structure of games and sub-games allows complex mixed-initiative dialogues to be modelled [6]. This approach therefore combines aspects of plan-based approaches with aspects of dialogue grammars, with moves providing a model of the conventional structure of dialogue, and the higher-level model of plans and goals being represented in terms of games, hence allowing more complex reasoning about the motivations of the dialogue and conversational cooperation.

2. HOMEY cancer dialogue system

Whilst it is possible to hand-code dialogues directly using finite state network approaches (e.g., in a language such as VoiceXML [9]), it is an expensive process, especially for complex or flexible dialogues. The problem is that the possible dialogues must be specified in advance and so the system must either constrain the user to the responses required by the system in the order in which it expects them (ignoring any over-informative answers) or, to be more

flexible, must include additional questions and transitions to handle the range of possible user responses. The problem with the latter approach is that “as soon as the questions multiply, the number of transitions grows to unmanageable proportions. This problem is further augmented if adequate repair mechanisms are to be included at each node for confirmation or clarification of the user’s response” [10]. As an example, Zinn et al. [11] report a personal communication from Vocalis plc that “a typical industrial dialogue system in the area of banking has 1500 states.” In the medical domain, where the knowledge structures are particularly complex (e.g., the breast cancer ontology described below contains about 100,000 concepts), the number of states could run into millions. Moreover, this kind of approach leads to dialogues that are difficult to reconfigure for different domains and which do not lend themselves easily to dynamic adaptation according to the changing goals of the dialogue system.

One of the aims of the EU-funded HOMEY project¹ was to provide better linkage between underlying medical technologies and the dialogue system. In other work, Cancer Research UK has developed a system (ERA) for advising doctors on whether patients require urgent referral for suspected cancer [12]. The system is currently accessed by a standard web browser that generates web pages for collecting patient data and reporting on results as shown in Fig. 1. The cancer showcase used to evaluate the dialogue system developed as part of the HOMEY project uses the knowledge representation developed for ERA, along with a medical ontology, to provide a spoken dialogue interface for entering data into this system (see <http://www.acl.icnet.uk/lab/homey.html> for a video demonstration). This system was designed to be used over the phone by healthcare professionals who were not cancer experts (e.g., general practitioners and nurse practitioners) and formalised UK National Health Service guidelines for cancer referrals. An example interaction with the system is given below (‘S’ indicates system utterances and ‘U’ indicates user utterances).

- (1) S: Hello, welcome to the breast cancer referrals system.
- (2) U: *Hello*
- (3) S: What is the patient’s age?
- (4) U: *She is thirty, female, and she has a bilateral nipple discharge*
- (5) S: Ok. And is it a bloodstained nipple discharge?
- (6) U: *No*
- (7) S: Ok. And is it a cloth-staining nipple discharge?
- (8) U: *No*
- (9) S: Ok. Does the patient have an acquired nipple deformity?
- (10) U: *What do you mean?*

- (11) S: For example: gestational inversion of nipple
- (12) U: *No, and she doesn’t have a lump, or any eczema*
- (13) S: Ok. Does the patient have a skin ulcer?
- (14) U: *Yes*
- (15) S: Ok. Does the patient have any intractable pain?
- (16) U: *Yes*
- (17) S: Ok. Does the patient have a breast abscess?
- (18) U: *Yes, and she has a cyst*
- (19) S: Ok. Do you mean a breast cyst?
- (20) U: *Yes*
- (21) S: Ok. So this patient is thirty, female, and has a skin ulcer, intractable pain, a breast abscess, a breast cyst, and a bilateral nipple discharge. Is that correct?
- (22) U: *No, she isn’t thirty*
- (23) S: What is the patient’s age?
- (24) U: *Thirty-five*
- (25) S: Ok. So this patient is thirty-five, female, and has a skin ulcer, intractable pain, a breast abscess, a breast cyst, and a bilateral nipple discharge. Is that correct?
- (26) U: *Yes*
- (27) S: I recommend urgent referral to breast specialist. Which action do you advise: urgent referral to breast specialist, support by general practitioner or non-urgent referral to breast specialist?
- (28) U: *Why do you recommend urgent referral?*
- (29) S: Urgent referral to breast specialist is indicated by the fact that this patient has a skin ulcer.
- (30) U: *What are the arguments for non-urgent referral*
- (31) S: Non-urgent referral to breast specialist is indicated by the fact that this patient has intractable pain and a breast abscess.
- (32) U: *Are there any arguments for support by general practitioner?*
- (33) S: There are no arguments for support by general practitioner.
- (34) U: *Ok, I advise urgent referral*
- (35) S: Ok. Do you mean urgent referral to breast specialist?
- (36) U: *Yes*
- (37) S: Ok. An urgent referral to breast specialist has been made for this patient.
- (38) U: *Ok*
- (39) S: Thankyou for using this service. Goodbye.
- (40) U: *Bye*

In utterances (3)–(26) above the system is acquiring data regarding the patient’s signs and symptoms. Utterance (4) shows how the user can provide more information than was requested in their answer, and (5) and (7) show how the system reacts by immediately following-up on the extra information provided. Utterance (10) shows a clarification sequence initiated by the user, (11) shows how the system reacts by re-phrasing and elaborating the question and (12) shows the use of negations by the user. Utterance (21) shows a system verification of the data collected and (22) shows a user-initiated repair. In utterances (27)–(36)

¹ Home Monitoring through an Intelligent Dialogue System (IST-2001-32424).

BREAST abort | instructions
Submit | Print | Reset | Restart | Dismiss

Patient Details:

Age: Gender: M F

Referral information (please tick boxes):

<p>Breast lumps:</p> <p>Discrete lump Yes <input type="checkbox"/> No <input checked="" type="checkbox"/></p> <p>Asymmetrical nodularity persistent at review after menstruation Yes <input type="checkbox"/> No <input checked="" type="checkbox"/></p> <p>Abscess Yes <input type="checkbox"/> No <input checked="" type="checkbox"/></p> <p>Persistent / refilling cyst Yes <input type="checkbox"/> No <input checked="" type="checkbox"/></p> <hr/> <p>Skin changes:</p> <p>Nodule Yes <input type="checkbox"/> No <input checked="" type="checkbox"/></p> <p>Distortion Yes <input type="checkbox"/> No <input checked="" type="checkbox"/></p> <p>Ulceration Yes <input type="checkbox"/> No <input checked="" type="checkbox"/></p>	<p>Pain:</p> <p>Intractable pain Yes <input type="checkbox"/> No <input checked="" type="checkbox"/></p> <hr/> <p>Nipple discharge / changes:</p> <p>Discharge Yes <input type="checkbox"/> No <input checked="" type="checkbox"/></p> <p style="padding-left: 20px;">Blood stained Yes <input type="checkbox"/> No <input checked="" type="checkbox"/></p> <p style="padding-left: 20px;">Large volume (sufficient to stain clothes) Yes <input type="checkbox"/> No <input checked="" type="checkbox"/></p> <p style="padding-left: 20px;">Bilateral Yes <input type="checkbox"/> No <input checked="" type="checkbox"/></p> <p>Eczema Yes <input type="checkbox"/> No <input checked="" type="checkbox"/></p> <p>Recent retraction or distortion (<3 mths) Yes <input type="checkbox"/> No <input checked="" type="checkbox"/></p>
--	---




Fig. 1. Screen-shot of the ERA breast cancer referrals application web interface.

the dialogue enters a decision negotiation phase where the system presents its recommendation (27), then the user can query the various arguments for or against the recommended decision (28, 30, 32), and the systems explains its reasoning (29, 31, 33). The final decision is made by the user (34). Finally, in utterance (37), the system informs the user of the action it has carried-out.

To link the dialogue with the underlying representation of the medical referral guidelines it was decided to introduce an intermediate layer, based on the framework of conversational games. This layer describes dialogue at both a plan-based level (in terms of games) and at a structural level (in terms of moves), hence bridging the gap between high-level domain tasks and low-level dialogue specifications such as VoiceXML, and allowing the dialogue to be dynamically adapted according to the current high-level context. This intermediate representation is referred to as the dialogue gameboard. The use of conversational games to model the dialogue was influenced by previous work [6,7] on the use of games for the discourse analysis of task-oriented dialogues. An alternative approach would be simply to model it in terms of atomic and non-atomic dialogue acts with the latter elaborated by recipes whose steps have ordering constraints between them as in the COLLAGEN system [13] amongst others.

The main difference between such systems and the approach taken here is not the use of conversational games per se, but rather the use of a multi-level specification of dialogue structure as an intermediate representation between underlying domain knowledge and the linguistic structure of the dialogue, hence providing a level at which both task and ontological constraints on dialogue structure can be captured.

2.1. Dialogue gameboard

The dialogue gameboard contains all the information required to generate a low-level specification for the next segment of dialogue. The set of game types proposed here for describing the current dialogue state are based on those described by [6,7] and includes games whose characteristic (initiating) moves are *inform* (presenting new information, e.g., “a referral has been made”), *instruct* (requesting that an action be carried-out by the user, e.g., “prescribe a course of tamoxifen”), *query-yn* (yes/no query) or *query-w* (query for a value) respectively. In addition to these moves, the games they initiate will also have response moves: *acknowledge*, *reply-yn*, and *reply-w*. Hence a typical *query-w* game will start with a *query-w* move by the initiating conversational partner, followed by a *reply-w* move by

the other partner and finally an optional *acknowledge* move by the initiating partner.

The gameboard is expressed as an XML language (see [14] for details) so that it can be easily mapped into VoiceXML via XSL Transformation templates. For example, consider the game specification shown below, which describes a game whose initiating move is to inform the user that the patient needs an urgent referral (the topic concept is taken from the domain ontology described later):

```
<Game id = "1" name = "Urgent Referral">
  <Topic concept = "PATIENT-IN-NEED-OF-URGENT-REFERRAL"/>
  <Move Type = "Inform"/>
</Game>
```

This can be realised as a VoiceXML document as shown below. This realises the *inform* move as a prompt and specifies a language model (grammar) for replies that can be mapped to an *acknowledge* move. This grammar can then be used by the speech recogniser to help interpret the speech signal. If either of the specified valid replies are recognised then the ‘filled’ event is fired and the attribute-value pair “*acknowledge = true*” (derived from the field name and the semantic tag assigned to the associated grammar) will be returned to the dialogue manager. Since an *acknowledge* move is optional, however, the user may not make any reply. In this case, the ‘noinput’ event fires and control returns to the dialogue manager to update the gameboard and generate the next system move.

```
<VXML>
  <form>
    <prompt>The patient is in need of urgent referral</prompt>
    <field name = "acknowledge">
      <grammar root = "rule1">
        <rule id = "rule1">
          <one-of>
            <item>ok</item>
            <item>right</item>
          </one-of>
          <tag>true</tag>
        </rule>
      </grammar>
    </field>
    <filled>
      <submit next = "..."/>
    </filled>
    <noinput>
      <submit next = "..."/>
    </noinput>
  </form>
</VXML>
```

As a further illustration, consider the specification of a *query-w* game given below which defines a request to know the patient’s sex:

```
<Game id = "27" name = "Patient Sex Enquiry">
  <Topic concept = "ORGANISM SEX STATE"/>
  <Move type = "Query-w">
    <Domain Multivalued = "False">
      <Option Concept = "MALE SEX"/>
      <Option Concept = "FEMALE SEX"/>
    </Domain>
  </Move>
</Game>
```

This can be mapped into a VoiceXML realisation as shown below. This again realises the *query-w* move as a prompt and the expected *reply-w* move is caught by the ‘filled’ event causing control to return to the dialogue system. The range of possible replies is defined by the speech grammar defined in the <grammar> element and each possible reply is assigned a semantic interpretation via a <tag> element. Hence, if the user replies “female” then an attribute-value pair “ORGANISM SEX STATE = FEMALE SEX” is submitted.

```
<VXML>
  <form>
    <prompt>What is the patient’s sex?</prompt>
    <field name = "ORGANISM SEX STATE">
      <grammar root = "rule1">
        <rule id = "rule1">
          <one-of>
            <item>male<tag>MALE SEX</tag></item>
            <item>female<tag>FEMALE SEX</tag></item>
          </one-of>
        </rule>
      </grammar>
    </field>
    <filled>
      <submit next = "..."/>
    </filled>
  </form>
</VXML>
```

Normally, the gameboard will specify more than one game, hence leading to an extended language model which allows the user to give replies other than just the answer to the question asked. Once the user’s response has been passed back to the interpreter it can be matched against the games in the gameboard to determine how it should be interpreted, i.e., what move the user intended to make and in which game.

It is not intended, however, that the gameboard should be authored directly. Instead, one of the aims of this work was to investigate use of existing knowledge representation schemas used in medicine as a basis for generating the dialogue. Hence, dialogue games are treated as primitives that are manipulated by higher-level knowledge representations: a domain plan (process specification) and ontology. These provide information of two distinct functional types: information on what task is to be accomplished (plan), and

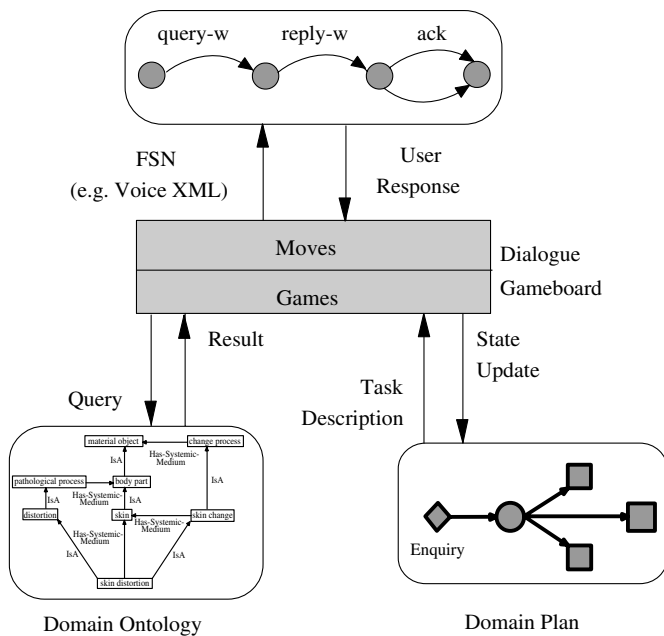


Fig. 2. Diagram showing the role of the dialogue gameboard in mediating between high-level domain and low-level dialogue representations.

information on the concepts associated with that task and their relations (ontology). This approach, shown diagrammatically in Fig. 2, is consistent with that taken by [15,16] which suggest a modular architecture with separate modules for the dialogue model, the task model and domain knowledge.

This distinction between dialogue specification and domain specification avoids any problems that might arise from a mismatch between representations suited to the task domain, e.g., clinical knowledge, and representations suited to language [17]. For example, Dahlbäck and Jönsson [15] distinguish two senses of the notion of ‘task’ as used in dialogue systems: firstly “some real-world non-linguistic activity that is directed towards achieving a particular goal” and secondly “the sequence of information that needs to be collected by the information providing system. ...[I]n the former case this knowledge is a separate structure, whereas in the latter it is intertwined with other aspects of the dialogue model.” For example, a medical guideline system will contain tasks such as retrieving data from a database, querying devices, decision-making, etc., all of which will occur independently of the dialogue. Conversely, the dialogue will contain tasks such as resolving misunderstandings, misrecognitions, etc., which will not be described in representations of clinical knowledge such as medical guidelines. Dahlbäck and Jönsson [15] therefore distinguish between underlying domain tasks (which in the approach described here are represented by the task specification) and the sequencing of dialogue-specific tasks (which are here represented by the high-level dialogue specification). Similarly, Flycht-Eriksson [16] argues that “domain knowledge reasoning should be clearly separated from dialogue management and performed by a separate

module.” This is also the approach taken in the TRIPS system [18] to maintain portability and flexibility.

Other approaches, such as [19], have conflated these levels of description. However, by using a single structure to represent both the domain and the dialogue, Maudet and Evrard [19] are forced to augment the set of dialogue moves with a series of distinct ‘logical moves,’ which allow the player to update the game board by application of logical rules. In particular, they propose a move ‘infer’ which represents the application of *modus ponens* to the player’s board. In their framework, such logical moves do not change the turn of the game and the same player continues to make subsequent moves until a dialogue move is made, at which point it becomes the other player’s turn. Hence, these moves have a different status to the dialogue moves. In the approach taken here, the dialogue gameboard only describes dialogue moves. Inference, e.g., applying rules that govern the system’s cognitive context, rules of rationality, or rules defining high-level strategies [19], is instead carried-out at the domain level by the task execution engine. The dialogue specification then changes indirectly as a reflection of changes in the domain plan.

The notion of dialogue gameboard adopted here is also similar in some respects to that of information state in the information state update model of dialogue [20]. In the approach described here, however, the level of description is games rather than moves (with moves instead represented by the low-level specification) and domain reasoning is delegated to a task specification layer. This architecture also has a lot in common with the TRIPS system [18]. The domain plan here plays a similar role to the TRIPS ‘Task Manager,’ which controls planning and scheduling, and the gameboard relates to what TRIPS refers to as the ‘Behavioural Agent’. This is responsible for the overall behaviour of the system, based on the goals of the system (e.g., task execution requests from the Task Manager), interpretation of user utterances, and any exogenous events (e.g., from monitors). TRIPS also has an ‘Interpretation Manager’ and ‘Generation Manager’ (where the Generation Manager can produce speech or graphics), but in TRIPS moves are conceptualised and interpreted/generated individually (as in the Information State model), whereas the approach described here is to generate low-level specifications to be realised by a separate client.

Finally, it should be noted that, whilst most of the games on the gameboard will be derived from underlying tasks, others will arise as a result of the dialogue itself in terms of tasks imposed by the user, e.g., to reply to a clarification request. These “communicative subgoals may also arise locally in the dialogue because of unanticipated responses and because of the complexity of the perceptual, understanding, evaluation, and other cognitive processes involved in interpreting and generating communicative behaviour” [21]. In balancing the demands from the domain plan on the one hand, and the user on the other, it is generally assumed that obligations imposed by user moves should be processed first [2]. Hence, the system must

respond to clarification questions or meta-level questions by the user before it can continue to pursue domain goals.

2.2. Domain plan

The domain plan determines the overall process to be followed and the individual clinical (non-linguistic) tasks required to achieve successful completion of the medical process. These tasks give rise to games on the gameboard that result in the dialogue system engaging in particular dialogues with the user to achieve those tasks. Once completed, it must then be possible for the dialogue system to update the state of the domain plan, e.g., to set the value of a data item that has been requested or to confirm the completion of some action. In addition, the domain plan must provide information regarding the relations between the various tasks in the plan. In particular, the dialogue system must know if one task is part of the decomposition of another or is dependent on another having completed, so that it can determine ordering relations between games on the gameboard. Such an approach is consistent with claims that dialogue structure is largely determined by task structure [22], or to put it another way: “engaging in a dialogue is typically not a goal in itself, but is motivated by some underlying task or goal one wants to achieve, and for which the dialogue is instrumental” [21].

The domain plan is currently implemented using Cancer Research UK’s *PROforma* toolset, which is a collection of tools for authoring, publishing and enacting processes specified in the *PROforma* language [23]. The toolset supports the definition of processes using four types of tasks that can be composed into networks representing plans to be carried out. Each task type is also associated with a graphical representation (icon), which is used in the graphical authoring tools for creating guidelines. These tasks are described in Fig. 3.

Each task type is a sub-class of an abstract ‘task’ super-class. The attributes of each sub-class determine the behaviour of its members during enactment of a guideline. All sub-classes inherit some generic attributes from the super-class which define: a goal that the task is to achieve, a trig-

ger that causes a task to be considered for enactment (asynchronously), pre-conditions for enacting the task, post-conditions that should hold after enactment, a cycling schema to control task iteration, and whether or not the task must be authorised by another agent before enactment. In addition to these generic attributes, each sub-class defines some class-specific attributes.

The plan class has additional attributes defining the tasks that compose the plan, constraints on the execution order of tasks, and conditions for successful and unsuccessful termination. The decision class has attributes defining decision candidates to be considered, arguments for/against candidates, and the scheme for combining arguments. Actions have an attribute to define a procedure to be carried-out by another agent, and enquiries have attributes to define the set of data items for which values are to be obtained from another agent.

To assist in the creation of *PROforma* specifications, the *PROforma* toolset contains a graphical authoring tool, which allows guidelines to be specified by drawing a high-level diagram depicting the tasks involved (using the icons shown earlier) and the relationships, e.g., scheduling constraints, between them (represented by arrows). The authoring tool also supports the definition of generic attributes (e.g., pre-condition, goal, etc.) and task-specific attributes for tasks (e.g., the data sources for an enquiry). An example guideline is shown in Fig. 4.

Once the guideline has been authored it can be submitted to the *PROforma* engine for enactment, at which point the individual tasks and attributes are used to generate procedures to carry-out. In the case of the example guideline given above, the enactment engine will request data regarding a patient’s symptoms, then make a decision based on that data as to whether the patient should have an urgent referral to a cancer specialist, a non-urgent referral, or no referral, then, depending on the decision, carry-out the appropriate action. Note that only a plan execution system is assumed here rather than full-blown dynamic AI planning. This is consistent with the general aim of finding a middle ground between the generality of AI-oriented approaches and computational efficiency [15].





Icon	Task	Description
	Plan	Sets of tasks to be carried out to achieve a critical goal. Plans may contain any number of tasks of any type (including other plans).
	Decision	Tasks which involve choices of some kind, such as choice of investigation, diagnosis or treatment.
	Action	Typically clinical procedures (such as the administration of an injection) which need to be carried out.
	Enquiry	Actions returning required information; typically requests for information or data from the user.

Fig. 3. The various task types used in the *PROforma* language.

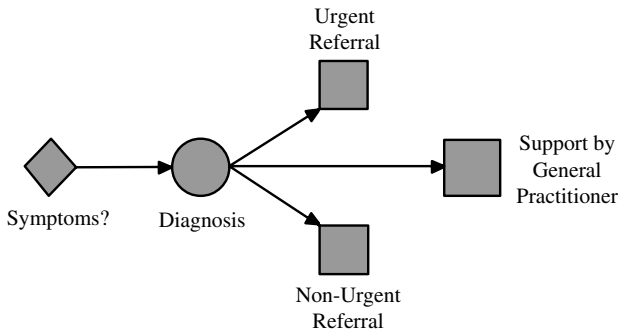


Fig. 4. An example PROforma plan as shown in the authoring tool.

The game types used in the dialogue gameboard can be derived fairly directly from PROforma task classes. For example, a ‘Plan’ task can be used to generate a game that simply contains sub-games. Similarly, an ‘Action’ task can be used to generate inform or instruct games (depending on the particular task properties such as the procedure attribute). An ‘Enquiry’ task whose ‘Type’ attribute is boolean can be used to generate a Query-yn game and, similarly, a Query-w game can be derived from non-boolean enquiries, with the particular type derived from the definition of the associated data item (either a named domain or an enumerated set of allowable values).

In addition to deriving the types of games involved, the hierarchical relations between games can similarly be derived from the PROforma task decomposition. For example, if T1 is a task which is decomposed into two other tasks T2 and T3, then a gameboard structure in which a generated game G1 contains generated sub-games G2 and G3 can be inferred. Similarly, sequencing dependencies between games can arise from underlying task precondi-

tions. For example, if T2 is a task that depends on task T1 being in a completed state then the game G1 should be addressed before G2 in the dialogue.

2.3. Domain ontology

Dahlbäck and Jönsson [24] argue that task-specific knowledge must be augmented with a conceptual model that describes general information concerning the relationships between objects in a domain. For example, in a library system they suggest a conceptual model in which ‘book Is-a publication,’ ‘author is-aspect-of publication,’ etc. It is assumed here that such information should be specified in a domain ontology, such as the fragment shown in Fig. 5.

The relations useful for language are, however, generally at a more abstract level than such domain ontological relations. For example a rhetorical ‘elaboration’ relation between two concepts, C1 and C2, might arise from various ontological relations between C1 and C2 such as: C1 denotes an instance of the class denoted by C2, C1 denotes an attribute of the object denoted by C2, C1 denotes a part of the whole denoted by C2 and so on [25]. Hence, information relations, such as elaboration, are defined between conversational games on the basis of more specific associations between the topics of games in the domain ontology.

An example of the role of such relations is given below in the context of a medical dialogue system which is trying to determine whether a patient with suspected breast cancer should be referred to a specialist or not.

- a. S: Is there any nipple discharge? [Query-yn]
- b. U: Yes [Reply-y]
- c. S: Ok... [Acknowledge]

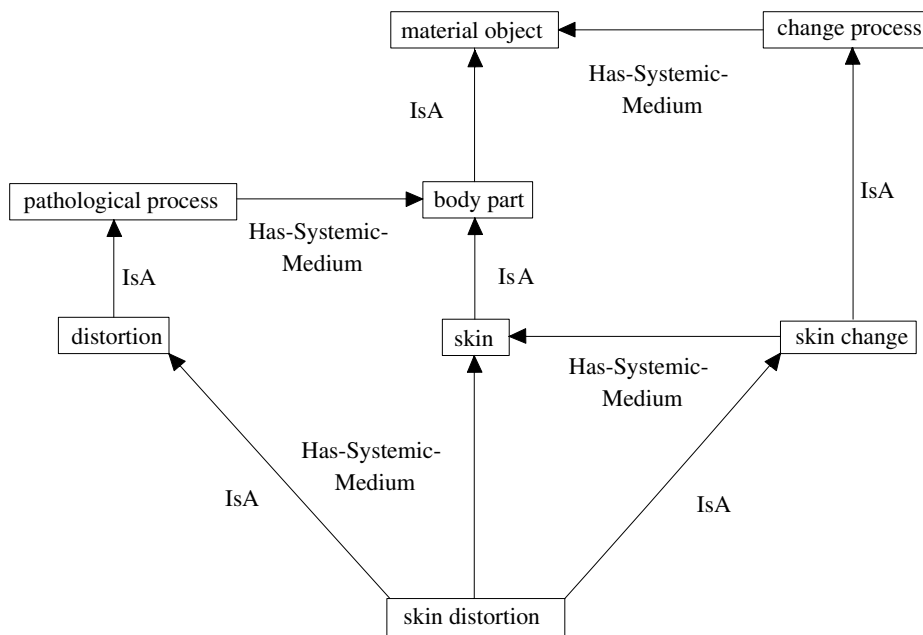


Fig. 5. A fragment of the ontology for the breast cancer domain.

- d. S: And is it a bloodstained nipple discharge? [Query-yn]
 e. U: No [Reply-n]
 f. S: Ok. [Acknowledge]

In this example, the second QUERYYN game (utterances d, e, and f) seeks to elaborate the information provided in the first game. This relation arises because the topic of the second game (bloodstained nipple discharge) is subsumed by the topic of the first game (nipple discharge). This relation is important for the purposes of dialogue management because it is the basis for the selection of the cue word “and” in utterance (d), and also licenses the use of an anaphor “it” to refer to the topic being elaborated. It is also important for the ordering of segments as a more coherent dialogue results if elaborating questions are asked immediately after questions that they elaborate (as above). The domain ontology is currently implemented using Language and Computing N.V.’s Ontology Browser [26].

3. Preliminary evaluation

For the breast cancer referrals showcase, the medical effectiveness of the underlying application (ERA) has already been determined in previous studies [12]. In future work it would be desirable to compare the usability of the web-based interface with that of the spoken dialogue interface developed here. So far, however, we have only carried out a preliminary validation study for the speech interface and so this section concentrates on evaluation metrics relevant to spoken dialogue systems.

3.1. Method

3.1.1. Subjects

The validation study was based on dialogues by 6 users who ranged from people familiar with the task domain through to people with no specific knowledge of the domain or any wider knowledge of medicine or healthcare.

3.1.2. Protocol

Due to the technical nature of the domain for the Breast Cancer Referrals demonstrator, the validation study was based on scripted interactions. Users were presented with the example transcript given earlier and asked to try to replicate that scenario. Note, however, that users were asked not to just blindly follow the script but to ensure that the information acquired by the system was correct according to the described scenario. Furthermore, the script contained examples of correcting misunderstandings, using help, etc., so users could quickly see how to use the system. No specific training was provided other than a review of the example transcript to point-out examples of taking initiative to ask clarification questions, querying decision recommendations, correcting errors at the verification stage and so on. Apart from that, users were simply expected to pick-up the range of possible dialogue moves implicitly from the script and from trying their own variations.

3.1.3. Variables

The following variables were considered as potentially important factors in evaluating the cancer referrals application: domain size and structure, degree of flexibility allowed at any point in the dialogue, verification strategy, variation in voices, and level of ambient noise. The following sections therefore describe the instantiations of these parameters in the evaluation study described here.

3.1.3.1. Domain size and structure. Domain size is a relevant parameter for the Cancer dialogue system because the number of domain concepts used in the dialogue determines the complexity of the language model generated for the speech recogniser: the higher the number of concepts, the more terms there will be in the speech grammar. In addition, hypernyms and hyponyms of these concepts are also included in the speech grammar to handle under-specified or over-specified user utterances, where the user refers to a more general or more specific concept, respectively, than the one that the system expected [27]. Domain structure is important because domain ontological relations place constraints on the dialogue system as to the order in which to request items. For example, if the user mentions that the patient has nipple discharge then the system should follow-up with elaborating questions regarding bloodstained nipple discharge and so on, rather than continuing with other unrelated questions, as described earlier. The ERA domain consisted of three basic tasks:

1. acquire the data values required to make a referral decision
2. make a recommendation to the user, allowing them to query the arguments for and against different decision candidates, and confirm the final decision advised by the user
3. inform the user when the appropriate action (urgent referral, non-urgent referral, etc.) was complete.

The data acquisition task (task 1 above) required values for 16 data items to be acquired by the dialogue system. These are listed in Table 1.

Note that there are ontological subsumption relations between some of the concepts associated with these data items: (a) *bilateral nipple discharge*, *bloodstained nipple discharge*, and *cloth-staining nipple discharge* are all subsumed by the concept *nipple discharge*, (b) *asymmetrical breast nodularity IS-A breast nodularity*, (c) *breast nodularity IS-A breast lump*, and (d) *gestational nipple retraction IS-A acquired nipple deformity*.

Once the above data has been collected, the system makes a recommendation to the user regarding the referral decision and allows the user to query the arguments for different candidates: urgent referral, non-urgent referral or no referral. If there are any arguments for urgent referral then that is recommended. If there are no arguments for urgent referral but there are some for non-urgent referral then that

Table 1
The data items to be acquired by the breast cancer referrals dialogue

Data name	Data type
Patient age	Integer
Patient sex	Male/female
Patient has nipple discharge	Boolean
Patient has bilateral nipple discharge	Boolean
Patient has bloodstained nipple discharge	Boolean
Patient has cloth staining nipple discharge	Boolean
Patient has breast cyst	Boolean
Patient has breast lump	Boolean
Patient has breast nodularity	Boolean
Patient has asymmetrical breast nodularity	Boolean
Patient has intractable pain	Boolean
Patient has acquired nipple deformity	Boolean
Patient has gestational nipple retraction	Boolean
Patient has nipple eczema	Boolean
Patient has breast abscess	Boolean
Patient has skin ulcer	Boolean

is recommended. If there are no arguments for either urgent or non-urgent referral then ‘no referral’ is recommended. Once the user confirms one of the candidates the system carries-out the advised task (although, for the purposes of this demonstrator, no action is actually performed) and informs the user when it is complete.

3.1.3.2. Degree of flexibility of user response. The degree to which the domain size and structure affects the performance of the dialogue system largely depends on the degree of flexibility of user response that is allowed. The more flexible the range of possible responses, the more scope there is to make use of ontological relations in the dialogue planning. In the current study, a wide form of mixed-initiative was used in which the user can not only answer more than one question with a single utterance, but can also explicitly take the initiative by asking questions of the system (e.g., clarification questions). The range of possible replies also includes references to over-specified and under-specified concepts [27]. This allows a complex dialogue structure as shown in the example below.

- (1) S: What is the patient’s age?
- (2) U: she is 35, female, and she has a nipple disorder
- (3) S: ok, what type of nipple disorder does the patient have?
- (4) U: What do you mean?
- (5) S: for example: nipple anomaly or nipple discharge
- (6) U: she has some nipple discharge
- (7) S: ok, ...

Here, when the user responds to the system’s question regarding patient age in utterance (2), they also specify some additional information: the patient’s sex and the fact that they have a nipple disorder. The concept ‘nipple disorder’ is under-specified (too general) and so the system responds in (3) by following-up this information with a clarification question. The user responds in (4) by issuing

their own clarification request to the system and the system provides an appropriate response (5). The user then answers the system’s clarification question (6), which is acknowledged by the system (7) before continuing with its dialogue plan.

3.1.3.3. Verification strategy. Whilst, it would be expected that a high degree of flexibility in user responses would provide the most usable system, flexibility also requires a complex language model and so increases the chance of misunderstandings. Another important variable was therefore the verification strategy employed to handle mis-recognitions, etc.

The verification strategy employed in this study was a ‘batch’ or ‘global’ strategy in which all the data required by the system to make a referral recommendation is collected without verification, and is then verified in a single batch before being used to make a decision. This is demonstrated in the example below.

- (1) S: So this patient is thirty, female, and has a skin ulcer, intractable pain, a breast abscess, a breast cyst, and a bilateral nipple discharge. Is that correct?
- (2) U: No, she isn’t thirty
- (3) S: What is the patient’s age?
- (4) U: Thirty-five
- (5) S: Ok. So this patient is thirty-five, female, and has a skin ulcer, intractable pain, a breast abscess, a breast cyst, and a bilateral nipple discharge. Is that correct?
- (6) U: Yes
- (7) S: I recommend. . .

Here, the system verifies all the collected data in (1), and the user indicates that part of the data is incorrect (2) leading the system to re-request that item (3). Once the new value is acquired in (4) the system repeats the verification with the new data (5) which is confirmed by the user (6) and the system then uses that data to make a decision regarding its recommendation for referral or non-referral (7).

3.1.3.4. Variation in voices. Since the speech recogniser is speaker-independent, speaker variation may also have an impact on system performance. In the current study, all the speakers were male and only one had a non-standard accent (i.e., not Received Pronunciation). All spoke with standard prosody and at a normal speaking rate.

3.1.3.5. Level of ambient noise. The level of ambient noise in a particular environment is also a factor in system performance. In the current study, the system was used in an office environment so there was a fairly high level of mainly unstructured noise (i.e., background noise such as doors opening and closing, typing, coughing, etc.) but also a small amount of structured noise (e.g., from other members of the office talking).

3.2. Results

The results of the validation study are broken into three main sections: an evaluation of dialogue manager competence, results for speech recogniser performance and results relating to dialogue manager performance.

3.2.1. Dialogue manager competence

Two previous projects, TRINDI² and DISC,³ have provided criteria for evaluating a dialogue manager's competence in handling certain dialogue phenomena. These are the TRINDI tick-list and the DISC dialogue management grid. Both of these are considered below.

3.2.1.1. TRINDI tick-list. The TRINDI Tick-List [26] consists of three sets of questions that are intended to elicit explanations describing the extent of a system's competence. The first set consists of nine questions relating to the flexibility of dialogue that the system can handle, the second set consists of five questions relating to the overall functionality of the dialogue system and the third set contains just two questions relating to the ability of the dialogue system to make use of contextual/domain knowledge to provide appropriate responses to the user. These are given below.

1. Can the system deal with answers to questions that give more information than was requested?
2. Can the system deal with answers to questions that give different information than was requested?
3. Can the system deal with answers to questions that give less information than was actually requested?
4. Can the system deal with negatively specified information?
5. Can the system deal with 'help' sub-dialogues initiated by the user?
6. Can the system reformulate an utterance on request?
7. Does the system deal with 'non-help' subdialogues initiated by the user?
8. Can the system deal with inconsistent information?
9. Can the system deal with belief revision?
10. Can the system deal with noisy input?
11. Can the system deal with barge-in input?
12. Can the system deal with no answer to a question at all?
13. Can the system check its understanding of the user's utterance?
14. Does the system only ask appropriate follow-up questions?
15. Is utterance interpretation sensitive to dialogue context?
16. Can the system deal with ambiguous designators?

An analysis of the current system and a comparison with other systems that have also used this evaluation metric is given in Table 2. Here, the columns 'P,' 'A,' and 'T' refer to results for the Philips Train Timetable System [29], the SRI-Autoroute System [30] and the Trains-95 system [31]. The Philips system is a research demonstrator based on the specific domain of train timetables, the SRI-Autoroute system is a natural language interface to a PC-based route finder package that uses a dialogue manager based on conversational games, and the Trains system is a collaborative planning system that employs plan recognition to determine the intentions underlying users' utterances. The results presented here are taken from the empirical tests reported in [28], but where empirical results were not available theoretical results from published papers, also reported in [28], are given instead (these are indicated by the use of a question mark after the value, e.g., 'Y?' means 'yes, in theory'). The column marked 'S' refers to results reported for the Siridus baseline architecture demonstrator in [32]. This is a general architecture intended to be used by researchers and based on the earlier TrindiKit architecture [20]. The columns marked 'G' and 'D' refer to results for the GoDiS and Delfos systems, respectively, as reported in [33]. These are both instantiations of the Siridus architecture: Delfos has been used to implement a telephone operator application, whilst GoDiS has been used to implement a wide range of different applications from travel agent to home control to telephone operator. Not all evaluations used all questions given above—where a result is not known a question mark is used in the relevant table column. Where the answer to a question is both yes and no (e.g., when a feature is only partially implemented) the value 'YN' is used.

3.2.1.2. DISC Dialogue Management grid. The DISC Dialogue Management grids [34] include a set of nine questions that are intended to elicit some factual information regarding the potential of a dialogue system. These are given below.

1. What initiative can the system cope with? (System/User/Mixed)
2. Free or bound order of main tasks?
3. Does the system initiate repair dialogues?
4. Does the system initiate clarification dialogues?
5. Can the user initiate repair dialogues?
6. Can the user initiate clarification dialogues?
7. Can indirect speech acts be handled?
8. Is there any difference between the system's use of speech acts and its ability to do topic spotting?
9. Does the system deal with ellipsis?

Question 8 above refers to whether the system determines speech acts based on keyword-spotting, e.g., the use of wh-pronouns such as "what", "where," etc., to indicate a question, or whether it uses wider knowledge of the sen-

² Task Oriented Instructional Dialogue, European Telematics Applications Programme project LEA-8314.

³ Esprit Long-Term Research Concerted Action No. 24823.

Table 2

Trindi Tick-List evaluation of the CR–UK dialogue system and comparison with other systems (P, Philips; A, Autoroute; T, Trains; S, Siridus; G, GoDiS; D, Delfos; Y, yes; N, no; YN, partially; Y?, yes in theory; and ? not known)

Q#	HOMEY cancer demonstrator	P	A	T	S	G	D
1	Yes, both extra information and over-specified replies	Y	Y	Y?	Y	Y	Y
2	Yes, it can accept direct, over-specified or under-specified answers to any question in the dialogue state	Y	Y?	Y?	Y	Y	Y
3	Yes, the system will then issue clarification questions as necessary	Y	Y	Y?	N	Y	Y
4	Yes, e.g., “she doesn’t have a cyst”	YN	YN	N	YN	Y	Y
5	Yes, e.g., “what do you mean?”	N	?	?	N	Y	Y
6	Yes, see previous question	?	?	?	?	N	Y
7	Possible but not yet implemented	Y	N	Y	N	N	Y
8	Yes, the most recent information is taken as being correct and older inconsistent information is removed	N	N	N	Y	Y	Y
9	Yes, see previous question	?	?	?	Y	Y	N
10	Yes, the user is asked to repeat their utterance if it couldn’t be matched by the recogniser	Y	Y	Y	Y	N	N
11	Yes	?	?	?	N	?	?
12	Yes, the user is prompted again to supply an answer	YN	YN	Y	Y	Y	Y
13	Yes, a ‘batch’ verification strategy is used	?	?	?	N	?	?
14	Yes, based on the current state of the domain process	Y?	N	N	Y	Y	Y
15	Yes, elliptical utterances are interpreted according to the current dialogue state	YN	YN	Y	Y	Y	Y
16	Yes, e.g., under-specified utterances. The system issues clarification questions	Y	N	Y	N	N	Y

tence structure, e.g., reversal of subject and verb to indicate a question so that “Is there a flight” is handled differently from “I would like a flight.”

An analysis of the current system and a comparison with other systems that have also used this evaluation metric is given in Table 3. Results are given here for the SIRIDUS baseline architecture demonstrator reported in [32] and for the GoDiS and Delfos systems as reported in [33].

For further details on the CR-UK dialogue system competence, the reader is referred to [35]

3.2.2. Speech recogniser performance

The following metrics were employed to evaluate speech recognition performance: word accuracy, sentence recognition, concept accuracy, and semantic recognition.

3.2.2.1. Word accuracy and sentence recognition. A commonly used measure of speech recognition performance is the accuracy of the system in recognising individual words. This is typically calculated using the formula below [36]:

$$WA = 100 \left(1 - \frac{W_s + W_i + W_d}{W} \right) \%$$

This measures accuracy in terms of the number of word substitutions (W_s), deletions (W_d), and insertions (W_i) relative to the total number of words (W) in the actual spoken utterances. Here, *substitution* means that a different word was recognised from the one spoken, *deletion* means that a word was spoken but not recognised, and *insertion* means that a word was recognised even though it was not spoken. In this case the word accuracy was 71.8% (693 errors out of 2459 words). Another measure is the percentage of sentence strings that were completely correctly recognised (i.e., where every word in the sentence was correctly recognised). In this case this was 59.2%.

3.2.2.2. Concept accuracy and semantic recognition. Another useful measure is the accuracy of the system in acquiring concepts (i.e., degree of semantic understanding). Based the standard measure of word accuracy given above [37] proposes the following formula to calculate concept accuracy:

$$CA = 100 \left(1 - \frac{SU_s + SU_i + SU_d}{SU} \right) \%$$

Table 3

DISC evaluation of the CR–UK dialogue system and comparison with other systems (Y, yes; N, no; and ?, not known)

Q#	HOMEY cancer demonstrator	Siridus	GoDiS	Delfos
1	Mixed, can answer any question or ask for clarification at any time	Mixed	Mixed	Mixed
2	Free, see previous question	Free	Free	Free
3	Yes, e.g., for no input, non-matching input and verification	Y	Y	Y
4	Yes, e.g., for under-specified replies	N	Y	Y
5	Yes, by providing negative feedback	Y	Y	Y
6	Yes, the user can ask for clarification of a question, e.g., “what do you mean?”	N	N	N
7	Yes, e.g., the declarative “I don’t understand” is interpreted as a query	?	?	?
8	Yes, speech acts are determined from the sentence structure not just keywords	N	?	?
9	Yes, e.g., short answers such as “35” “she is 35” etc., are interpreted in the context of the question asked	Y	Y	Y

This measures accuracy in terms of the number of substitutions (SU_s), insertions (SU_i), and deletions (SU_d) of semantic units relative to the total number of semantic units uttered (SU).

In this case, there were a total of 1084 Semantic Units, of which 941 (86.8%) were correctly recognised, with 239 errors giving a concept accuracy of 78.0%. The total number of utterances was 687, of which 523 were correctly interpreted, giving a semantic recognition rate of 76.1%.

3.2.3. Dialogue manager performance

The performance of the dialogue manager was evaluated using the following metrics: degree of success in achieving the desired task, cost of successful completion, and the overall usability of the system.

3.2.3.1. Task success. The following aspects of task success were considered: the number of users who managed to complete a dialogue, the correctness of the data acquired from the user [38], and the correctness of data provided by the system (transaction success) [39].

In the majority (80.8%) of cases, users successfully completed the dialogue. Hence, any errors that occurred were, in general, not severe enough to prevent the user from reaching the end of the dialogue. Of those cases where users hung up, about half were due to consistent mis-recognition of a single lexical item (“lump”) by the speech recogniser. The other half were due to mis-recognition of the user’s final decision as new data (as illustrated by the first example in the section on usability below), causing the system to repeat the verification and decision stages and giving the impression that the system had started again from the beginning.

For those dialogues that successfully reached the decision stage (86.7%), the accuracy of the system in acquiring the data items necessary to make a decision (patient details, health symptoms, etc., as described above) was evaluated. Since a verification strategy was used by the system to check the correctness of these items, a high degree of accuracy was observed. In fact of the total of 416 data items for which values were acquired, only 10 values were incorrect, hence the overall accuracy was 97.6%. In the cases where incorrect values were acquired this seemed to be due to the user mis-hearing or not properly attending to the verification prompt and confirming that data was correct even though there were errors. This may be due to the verification prompts being over-long or because of unclear pronunciation of some values by the speech synthesizer.

Finally, turning to transaction success, it was found that, of the dialogues that were completed by users, the referral task was successfully achieved (i.e., an appropriate referral was made for the patient being described) in 85.7% of cases. Hence the overall transaction success (including dialogues that were not completed) was 69.2%. Unlike the acquisition of patient and health symptom data, the referral decision acquired from the user was not verified by the system and so, in the cases where an incorrect refer-

ral was made, the error was generally due to a mis-recognition by the system during the negotiation of the decision (e.g., interpreting a user utterance as confirming a candidate when it was actually a question regarding the arguments for that candidate). The success of the verification strategy for data acquisition, however, suggests that it should be extended to also include verification of decisions.

3.2.3.2. Dialogue costs. The task success of a dialogue system needs to be weighed against the costs in using the system. For example, a system that verified every data item immediately at the point where it was acquired would have a high task success but at the cost of a long and tedious dialogue. The following measures of dialogue ‘cost’ were considered: system response time, overall amount of time required to complete a dialogue, the number of turns required to complete a dialogue [39], and the proportion of turns that were spent correcting errors such as misrecognitions, misunderstandings, etc., i.e., a ‘correction rate’ [39].

The median response time for the dialogue manager (from receiving a voice browser request to sending a response) was 531 ms (ranging from a minimum of 16 ms to a maximum of 12,047 ms). High response times occurred only at the start of a dialogue in cases where the system had previously been reset and so the domain ontology had been removed from memory. In these cases, this data had to be reloaded from file. Once loaded, however, the system response time returned to average levels, and subsequent dialogues were at this level throughout until the server was reset.

The median total number of turns (including both user and system moves) was 50. The median time taken to successfully complete a dialogue was 277 s (i.e., 4 min and 37 s) and ranged between 188 and 444 s. This metric correlates closely with the number of turns per dialogue (correlation coefficient $r = 0.88$). The fastest dialogues were those where there were few mis-recognitions (and hence a lower number of corrections) and where the user took advantage of mixed-initiative to provide a lot of information in one utterance (e.g., “she is thirty, female and has a bilateral nipple discharge”) rather than waiting to be prompted for each item. Conversely, the longest dialogue was one in which the speech signals were very noisy and there were many speech recognition errors (and hence corrections).

This dialogue duration metric becomes more meaningful when normalised according to the complexity of the dialogue, e.g., as measured by the average number of concepts acquired. In this study, the median number of concepts acquired in an interaction was 37 (in a range between a minimum of 13 and a maximum of 58, with the lowest values recorded for dialogues that were not successfully completed). Hence, the median time to acquire a concept can be estimated as 7 s per concept (ranging between a minimum of 4.6 s and a maximum of 11.5 s). These times include both user and system turns, and so it appears that the data acquisition accuracy reported earlier was not achieved at

the cost of dialogue efficiency as measured by dialogue duration.

Finally, turning to the correction rate metric, it was found that the median number of turns involving spontaneous (i.e., non-scripted) error corrections in each dialogue was 2, and that the median proportion of turns spent correcting errors was 8.2%. Hence, the dialogues also had a low cost in terms of the amount of time spent by the user in correcting system errors.

3.2.3.3. Usability. The following usability metrics were considered: the number of times a user made use of ‘help,’ the contextual appropriateness of system responses [39], the quality of user responses [40], and user report.

In general, there was no spontaneous (i.e., non-scripted) use of system help during a dialogue (median number of help utterances was zero). In fact, help was only used nine times in the entire set of dialogues. For those dialogues in which it was used, the median proportion of turns spent on help in a dialogue (a notional ‘help rate’) was 5.3% (ranging from a minimum of 2.6% to a maximum of 10.5%).

The quality of system responses was evaluated according to the SUNDIAL ‘Contextual Appropriateness’ metric [39]. Each response was labelled as either appropriate (AP), borderline appropriate/inappropriate (AI), completely inappropriate (IA), incomprehensible (IC) or a total failure (TF) (e.g., failure to respond at all to the user’s utterance). It was found that, of 691 system responses, 547 (79.2%) were classified as AP, 32 (4.6%) as AI, 100 (14.5%) as IA, 8 (1.2%) as IC, and 4 (0.6%) as TF.

Most of the responses in the IA category occurred as a result of mis-recognitions of the user’s last reply (e.g., recognising “she has an ulcer” as “she has a lump” and then asking “what type of lump does she have?”). Of those responses included in the AI category, most were cases where the response was appropriate according to local context (e.g., the user’s last utterance) but not really appropriate in the wider context (the current dialogue segment). This is illustrated by the following fragments (taken from the dialogues recorded in this study).

- | | |
|--|----|
| (1) S: I recommend urgent referral ... What do you advise? | |
| (2) U: ok urgent referral | |
| (3) S: So this patient is 2. Is that correct? | IA |
| (4) U: no she is forty | |
| (5) S: So this patient is 14. Is that correct? | AI |
| (6) U: no | |
| (7) S: In what way is it incorrect? | AI |
| (8) U: she is forty | |
| (9) S: So this patient is 40. Is that correct? | AI |
| (10) U: yes | |

In this fragment, the system is in the decision phase of the dialogue (1) and the user advises that an urgent referral is necessary (2). Unfortunately, the user utterance is mis-

cognised as a correction to the patient’s age (in this case the string returned by the speech recogniser was “ok her age is two or”) and the system then tries to verify the new data (3). This system response is therefore inappropriate (IA) in the context of the user’s last utterance. However, the user then goes on to correct the misunderstood data (4, 6, 8, and 10) which entails further system responses (5, 7, and 9) which are appropriate in the local context given that the user now really is changing the age value, but are inappropriate given the wider context of confirming a referral decision (AI).

The following dialogue fragment provides a similar example.

- | | |
|--|----|
| (1) S: What kind of lump does the patient have? | IA |
| (2) U: no lump | |
| (3) S: And is it a breast nodularity? | IA |
| (4) U: wh...what do you mean? | |
| (5) S: For example: pre-existing breast nodularity, or symmetrical breast nodularity | AI |
| (6) U: Neither | |
| (7) S: Does the patient have a skin ulcer? | AP |

In this fragment, the system has mis-interpreted a user utterance as meaning that the patient has a lump (1) and the user is trying to correct the data (2). Unfortunately, the user utterance is mis-recognised (the string returned by the speech recogniser in this case was “a lump”) leading to another inappropriate (IA) utterance (3) trying to clarify the type of lump. The user then asks for help (4) and so the system response (5) providing an explanation is appropriate in the local context of the user’s last utterance but inappropriate in the wider context in which the user is trying to indicate that the patient does not have lump at all (AI).

It is possible to combine some of the categories given above to get a broader characterisation of system responses. For example ‘adequate responses’ could be defined as those that are contextually appropriate or borderline appropriate (AP + AI), and ‘inadequate responses’ could be defined as those that are inappropriate, incomprehensible or failures (IA + IC + TF). Given these categories, 83.8% of responses would be classified as adequate and only 16.2% as inadequate. Hence, the majority of system responses were sufficiently appropriate (given the context) to be understandable by the user (and hence most dialogues were successfully completed, as described earlier).

As well as evaluating system responses, the quality of user responses was also analysed using the Behavioural Coding Scheme [40]. This measures the degree to which user answers could be characterised as ‘concise and responsive’ (e.g., “S: what is the patient’s age? U: 35”), ‘usable but not concise’ (e.g., “S: what is the patient’s age? U: her age is 30”), ‘responsive but not usable’ (e.g., “S: what is the patient’s age? U: she’s middle-aged”), ‘not responsive’ (e.g., “S: what is the patient’s age? U: I don’t know”), or as containing no speech (e.g., just noise or silence).

It was found that, of a total of 687 user responses, 347 (50.5%) could be classified as concise and responsive (C), 300 (43.7%) as usable but not concise (U), 14 (2%) as responsive but not usable (R), 3 (0.4%) as not responsive (NR) and 23 (3.4%) as not containing any speech (NS). Note that a very high proportion of category C compared to category U would be characteristic of a primarily system-initiated dialogue with the user simply providing answers as briefly as possible. On the other hand, the pattern observed here of almost equal numbers of category C and U responses is indicative of a mixed-initiative system in which the user can provide more data than was requested, phrase it in more complex ways than just a single-word reply, and ask clarification questions of the system. Hence, the analysis of user responses is indicative of a high degree of dialogue flexibility.

The first three of these categories (C + U + R) can be amalgamated into a single class of ‘Adequate Answer’ and the last two (NR + NS) into a class of ‘Inadequate Answer’ [40]. Under this scheme, 96.2% of user responses were adequate and only 3.8% were inadequate. Hence, the prompts provided by the system were sufficient to elicit adequate (responsive, usable, and/or concise) answers from the user.

The final metric typically employed to measure usability is ‘user report’. Since the cancer showcase was a prototype system, however, no systematic investigation of users’ qualitative assessments of usability was undertaken. However, anecdotally, users did report finding the system reasonably easy to use and, in particular, were impressed with the ease with which they could correct incorrect data items, e.g., using a single utterance such as “no she is forty and she does not have a cyst but she does have an ulcer.”

Almost all users also reported problems with repeated mis-recognitions of certain words, which required some perseverance to get past, and problems with the lack of verification of the final referral decision (as noted earlier). The last problem was compounded by the fact that all data slots were kept open (i.e., fillable) until the end of the dialogue, so last-minute errors could cause previously set slots to be overwritten and the verification and decision phase of the dialogue to be repeated just when the dialogue appeared to be complete. However, except for a small number of cases, these problems did not prevent the user from completing the dialogue.

3.2.4. Performance comparison

A comparison of the performance results described above with other systems reported in the literature for a common subset of the objective metrics (i.e., task success and dialogue cost) is summarised in Table 4. Talk’n’Travel is a system for making air travel plans over the telephone, using mixed-initiative with both open-ended and directed prompts and a mixture of implicit and explicit verification. The results quoted here are from an independent evaluation of the system, involving untrained subjects, conducted as part of the DARPA Communicator program [41]. The

LMSI RAILTEL system is a system for accessing French rail service (SNCF) information. The results quoted here are from field trials with naïve subjects on two prototype scenarios [42]. The Let’s Go bus information system is a research system that was adapted to be usable by the general public. The results reported here are based on public usage during the first 3 weeks of operation [43]. Finally, the Pain Monitoring Voice Diary (PMVD) system was developed by Spacegate Inc for monitoring chronic pain patients. It is a system-initiated dialogue system using a finite state model with system prompts carefully crafted to restrict the range of user responses and an immediate explicit verification strategy so that no data value is accepted without being explicitly confirmed by the user [44].

It is difficult to make definitive comparisons between systems since the domains, tasks and user populations are quite varied,⁴ but it can be seen that the cancer dialogue system described here has a similar completion and task success rate to the DARPA Communicator Talk’n’Travel system, despite poorer speech recognition performance. It is difficult to compare the dialogue costs of the two systems since only overall dialog completion time is given for Talk’n’Travel, whereas a more meaningful metric would be the time to acquire a concept. Presumably for the travel domain the number of concepts acquired in a dialog is quite low (as with the RailTel and bus info systems) so, relative to the complexity of the task, the time given for the Talk’n’Travel system seems quite high.

The RailTel system performs less well than the CR–UK system in terms of dialog completion and task success, and also has a very high dialogue completion time (this seems to be due to some problems encountered in the interpretation of times by the system). The Let’s Go bus system also performs less well on task success, although no data on time taken per dialogue is reported. Unlike the other systems the test population for this system was unconstrained and included elderly people and non-native speakers in noisy conditions, e.g., on mobile phones standing on busy streets, hence the relatively poor speech recognition results.

The PMVD system has very high dialogue completion and task success rates. This is achieved, however, through a strict system-initiated dialogue with a conservative verification strategy. The cost of this approach is that the time per turn and time to acquire a concept is about twice that of the CR–UK cancer dialogue system.

Finally, it should be noted that the small number of users and the use of scripted interactions in the CR–UK system evaluation make it difficult to judge how general the preliminary results reported here are. A further more extensive investigation is therefore required before definitive statements can be made.

⁴ This is a problem that has been addressed by approaches such as the PARADISE metric [walker et al.] but this relies on having a metric of user satisfaction to use to determine the weightings of task success and cost functions via linear regression. In this case we do not have user satisfaction scores so cannot properly apply the PARADISE approach.

Table 4

A comparison of the CR–UK cancer dialogue system with four other systems on objective performance metrics

Metric	CR–UK cancer system	Talk 'n' Travel	RailTel	Let's Go Bus Info	PMVD	Median value
Word error rate	28.2%	21%	21.6%	60%		24.9%
Concept error rate	22%	10%	8.35%	48.1%		16%
Dialogue completion rate	80.8%	82%	73%		95.8%	81.4%
Transaction success	69.2%	70.5%	59.5%	43.6%	93.9%	69.2%
#Turns per dialogue	50		8	14.6	14.8	14.7
Combined correction and help rate	11.5%			20%		15.8%
Time to complete a dialogue	277 s	246 s	219 s		105.6 s	232.5 s
Time per turn	3.5 s		27.4 s		7.19 s	7.19 s
#Concepts acquired per dialogue	37		4	3.5	7.85	5.9
Time to acquire a concept	7 s		54.7		13.46 s	13.46 s

The final column shows the median average of row values.

3.2.5. Reconfigurability

The reconfigurability of the dialogue system for use in other domains was also tested by using it to implement a new, larger, and more complex application, namely genetic risk assessment⁵ (RAGs). The domain plan for breast cancer referrals was replaced with another pre-existing *PROforma* process specification for genetic risk assessment and a domain ontology was created by hand to represent family relations. Finally, the sentence and speech grammar generation templates were extended to cover the wider range of system and user utterances for the new domain. Although, the RAGs application has not yet undergone any performance evaluation, the limited nature of the changes that were required in porting the original dialogue system to the new domain suggests that the framework developed here is reasonably general and that reconfiguration for other domains can be achieved with probably much less effort than would be required to build a new dialogue application from scratch. Further exploration of this issue is obviously required, however, before any definitive statements can be made. In particular, it would be desirable to compare the efficiency of developing an application using this framework as compared to hand-crafting it using VoiceXML or visual IVR design tools.

4. Conclusions

This paper has presented an approach to building spoken dialogue systems in which the dialogue model is split into high-level and low-level representations, with the latter generated dynamically from the former via an intermediate representation based on conversational games. This approach can therefore make use of current voice-based standards such as VoiceXML to achieve independence from specific speech recognition and synthesis technologies, whilst also utilizing high-level notions of domain tasks and concepts that form the basis of much research into discourse structure. It was further proposed that the high-level

representation can be derived from a domain plan and ontology, hence removing the need to author dialogues directly, and providing a degree of reconfigurability, as well as allowing greater integration with the application domain and non-dialogue tasks.

A particular application of this approach was described, namely a system for advising medical practitioners whether or not a potential cancer patient should be referred to a specialist. An analysis of the competence of the breast cancer demonstrator indicates that applications implemented using this framework can handle a wide range of dialogue phenomena. This high level of competence derives from basing the dialogue system on high-level knowledge representations, which allow more sophisticated reasoning about dialogue structure than the simple task lists typically employed in dialogue systems.

Finally, our evaluation of the overall dialogue system performance, though tentative, suggests that these benefits have not been gained at the expense of performance. The cancer application demonstrates good speech recognition performance (concept accuracy of 78% with 86.8% of concepts correctly understood), and good performance in acquiring data (97.6% correct) and successfully completing transactions (80.8% of dialogues completed, of which 85.7% achieved the dialogue goal). Moreover this was achieved whilst maintaining a fast response time from the dialogue manager (on average 531 ms from request to response), and an efficient dialogue from the user's point of view (on average 7 s to acquire a concept, including both system and user turns) with a low correction rate (on average 8.2% of turns spent correcting errors).

In terms of usability, the majority of system responses were found to be contextually appropriate (79.2%, with a further 4.6% borderline cases), and elicited user responses that were almost all (96.2%) adequate (i.e., responsive, usable and/or concise). Possibly for these reasons, users did not, on average, make use of system help, and when they did it was only for a small proportion (5.3%) of turns.

A more extensive evaluation now needs to be performed to determine the extent to which these results generalise to larger user populations, to more fully investigate the

⁵ A video demonstration of this application is available at <http://www.acl.icnet.uk/lab/homey.html>.

advantages of this approach over hand-coding in languages such as VoiceXML, and to compare the usability of the speech interface developed here with the original web interface.

Acknowledgments

This work was funded by the European Union under the 5th Framework HOMEY Project, IST-2001-32434 (see <http://www.acl.icnet.uk/lab/homey.html>). Thanks to the project partners for many useful discussions and advice.

References

- [1] Cohen P, Perrault CR. Elements of a plan-based theory of speech acts. *Cogn Sci* 1979;3(3):177–212.
- [2] Traum DR, Allen JF. Discourse obligations in dialogue processing. In: Proceedings of the 32nd ACL, Las Cruces, New Mexico; 1994. p. 1–8.
- [3] Pulman SG. Conversational games, belief revision and Bayesian networks, CLIN VII. In: Landsbergen J, et al., editor. Proceedings of the seventh computational linguistics in the Netherlands meeting, 1996, 1997. p. 1–25.
- [4] Poesio M, Traum D. Towards an axiomatization of dialogue acts. In: Hulstijn J, Nijholt A, editors. Proceedings of the twente workshop on the formal semantics and pragmatics of dialogues, Enschede; 1998. p. 207–22.
- [5] Power R. The organization of purposeful dialogues. *Linguistics* 1979;17:107–52.
- [6] Kowtko JC, Isard SD. Conversational games within dialogue, research paper 31, Human Communication Research Centre, Edinburgh, 1993.
- [7] Carletta J, Isard A, Isard S, Kowtko J, Doherty-Sneddon G. HCRC dialogue structure coding manual. Technical Report HCRC/TR-82, University of Edinburgh, UK: Human Communication Research Centre; 1996.
- [8] Poesio M, Mikheev, A. The predictive power of game structure in dialogue act recognition: experimental results using maximum entropy estimation. In: Proceedings of ICSLP98, 1998.
- [9] McGlashan S, Burnett DC, Carter J, Danielsen P, Ferrans J, Hunt A, Lucas B, Porter B, Rehor K, Tryphonas S. Voice Extensible Markup Language (VoiceXML) Version 2.0, W3C Recommendation, 16th March. <<http://www.w3.org/TR/voicexml20/>>; 2004.
- [10] McTear MF. Modelling spoken dialogues with state transition diagrams: experiences with the CSLU toolkit. In: Proceedings of the fifth international conference on spoken language processing (ICSLP), Sydney, Australia, 1998. p. 30 November–4 December.
- [11] Zinn C, Moore JD, Core MG. A 3-tier planning architecture for managing tutorial dialogue. In: Proceedings of the sixth international conference on intelligent tutoring systems (ITS), Biarritz, France; 2002. p. 574–84.
- [12] Bury J, Humber M, Fox J. Integrating decision support with electronic referrals. In: Rogers R, Haux R, Patel V, editors. Medinfo. Amsterdam: IOS Press; 2001.
- [13] Rich C, Sidner C. COLLAGEN: when agents collaborate with people. In: Proceedings of the first international conference on autonomous agents. California: Marina del Rey; 1997.
- [14] Beveridge M, Milward D. Definition of the high-level task specification language. Deliverable D11, EU 5th Framework HOMEY Project, IST-2001-32434, 2003 <<http://www.acl.icnet.uk/lab/homey.html>>.
- [15] Dahlbäck N, Jönsson A. Knowledge sources in spoken dialogue systems. In: Proceedings of Eurospeech'99, Budapest, Hungary, 1999.
- [16] Flycht-Eriksson A. A domain knowledge manager for dialogue systems. In: Proceedings of the 14th European conference on artificial intelligence, ECAI 2000. Amsterdam: IOS Press; 2000.
- [17] Hovy EH. In defense of syntax: informational, intentional, and rhetorical structures in discourse. In: Proceedings of the workshop on intentionality and structure in discourse relations. ACL-93, Columbus, OH, 1993.
- [18] Allen J, Ferguson G, Stent A. An architecture for more realistic conversational systems. In: Proceedings of the intelligent user interfaces 2001 (IUI-01), Santa Fe, NM, 2001. January 14th–17th.
- [19] Maudet N, Evrard F. A generic framework for dialogue game implementation. In: Hulstijn J, Nijholt A, editors. Proceedings of the second workshop on formal semantics and pragmatics of dialogue, May 13–15, Netherlands: University of Twente, Enschede; 1998.
- [20] Larsson S, Traum D. Information state and dialogue management in the TRINDI dialogue move engine toolkit. *Natural Language Engineering* 2000;6:323–40. Special issue on spoken language dialogue system engineering.
- [21] Bunt H. Dynamic interpretation and dialogue theory. In: Taylor M, Neel F, Bouwhuis D, editors. The structure of multimodal dialogue, vol. 2. Amsterdam: John Benjamins; 1996.
- [22] Grosz B, Sidner C. Attention, intention and the structure of discourse. *Comput Linguist* 1986;12(3):175–204.
- [23] Fox J, Beveridge MA, Glasspool D. Understanding intelligent agents: analysis and synthesis, AI communications, 16, Amsterdam: IOS Press; 2003. p. 139–52.
- [24] Dahlbäck N, Jönsson A. Integrating domain specific focusing in dialogue models. In: Proceedings of EuroSpeech'97, Rhodos, Greece, 1997.
- [25] Mann WD, Thompson SA. Rhetorical structure theory: towards a functional theory of text organization. *Text* 1988;8(3):243–81.
- [26] Ceusters W, Martens P, Dhaen C, Terzic B. Link factory: an advanced formal ontology management system. In: Proceedings of the interactive tools for knowledge capture workshop, KCAP-2001, Victoria, BC, Canada; 2001.
- [27] Milward D, Beveridge MA. Ontologies and the structure of dialogue. In: Proceedings of the CATALOG, eighth workshop on the semantics and pragmatics of dialogue, Barcelona, Spain; 2004, 19th–21st July.
- [28] Bohlin P, Bos J, Larsson S, Lewin I, Matheson C, Milward D. Survey of existing interactive systems. Deliverable D1.3, TRINDI Project, LE4-8314; 1999.
- [29] Aust H, Oerder M, Siede F, Steinbiss V. A spoken language enquiry system for automatic train timetable information. *Philips J Res* 1995;49(4):399–418.
- [30] Lewin I. The autoroute dialogue demonstrator. Technical report CRC-073. SRI Cambridge Computer Science Research Centre; 1998.
- [31] Allen J, Miller B, Ringger E, Sikorski T. A robust system for natural spoken dialogue. In: Proceedings of the 34th ACL, Santa Cruz; 1996a; p. 62–70.
- [32] Berman A, Cooper R, Ericsson S, Hieronymus J, Jonson R, Larsson S, Milward D, Torre D. Implemented SIRIDUS system architecture (Baseline). Deliverable 6.2, SIRIDUS Project IST-1999-10516; 2000.
- [33] Larsson S, Jonson R, Amores G, García C, Quesada JF. Evaluation of contribution of the information state based view of dialogue. Deliverable 3.4, SIRIDUS Project IST-1999-10516; 2002.
- [34] Heid U, Bernsen N, Dybkjaer L. Current practice in the development and evaluation of spoken language dialogue systems. Deliverable D1.8, DISC project, esprit long-term research concerted Action No. 24823; 1998.
- [35] Beveridge M, Giorgino T, Falavigna D, Gretter R. Validation report, public deliverable D19, EU HOMEY Project, IST-2001-32434; 2004 <<http://www.acl.icnet.uk/lab/homey.html>>.
- [36] De Mori R. Spoken dialogues with computers. New York: Academic Press; 1998.
- [37] Boros M, Eckert W, Gallwitz F, Görz G, Hanrieder G, Niemann H. Towards understanding spontaneous speech: word accuracy vs.

- concept accuracy. In: Proceedings of the ICSLP'96, Philadelphia, PA; 1996. p. 1009–12.
- [38] Walker MA, Litman DJ, Candace AK, Abella A. Evaluating spoken dialogue agents with PARADISE: two case studies. *Comput Speech Lang* 1998;12(3).
- [39] Simpson A, Fraser. Black box and glass box evaluation of the SUNDIAL system. In: Proceedings of the third European conference on speech communication and technology (Eurospeech'93), Berlin, Germany; 1993.
- [40] Sutton S, Hansen B, Lander T, Novick DG, Cole R. Evaluating the effectiveness of dialogue for an automated spoken questionnaire. Technical report CS/E95-12, Department of Computer Science and Engineering, Oregon Graduate Institute of Science and Technology; 1995.
- [41] Stallard D. Evaluation results for the Talk'n'Travel System. In: Proceedings of the first international conference on human language Technology, ACL, San Diego, 2000; pp 1–3.
- [42] Bennacef S, Devillers L, Rosset S, Lamel L. Dialog in the RAILTEL telephone-based system. In: Proceedings of the fourth international conference on spoken language processing (ICSLP), Philadelphia, PA, USA; 1996. p. 550.
- [43] Raux A, Langner B, Bohus D, Black AW, Eskenazi M. Let's Go Public! Taking a Spoken Dialog System to the Real World. In: Proceedings of interspeech, September 4–8, Portugal; Lisbon; 2005. p. 885–8.
- [44] Levin E, Levin A. Spoken dialog system for real-time data capture. In: Proceedings of interspeech, September 4–8, Portugal; Lisbon; 2005.