A coding manual for the Linköping dialogue model

Nils Dahlbäck Arne Jönsson
Department of Computer and Information Science
Linköping University
S-581 83 LINKÖPING, SWEDEN
nilda@ida.liu.se, arnjo@ida.liu.se

January 5, 1998

Abstract

This paper describes the coding model and coding conventions used for the LINDA and LINLIN dialogue models developed in Linköping. Both dialogue and focus structure are included in the analysis. The model is primarily intended for the dialogue component of natural language interfaces and other similar dialogue situations. The underlying assumptions are described briefly, and the criteria for coding of the different categories are described.

1 Introduction

Computational dialogue models for natural language interfaces need to account for two concurrent tasks: *dialogue structure* and *focus structure* (cf. Grosz & Sidner, 1986). Dialogue structure involves managing relationships between segments in the dialogues. Focus structure is concerned with the recording and structuring of entities mentioned in the discourse to allow a user to refer to them in the course of the interaction.

The LINLIN model(Jönsson, 1993) is an extended version of the LINDA-model (Dahlbäck, 1991) which accounts for both dialogue and topic structure. For the dialogue structure, an initiative-response (IR) structure similar to the LINDA model is assumed. The model for dialogue management assumes that a dialogue is divided into three main levels on the basis of structural complexity. One level corresponds to the size of a dialogue (D), another level to the size of a discourse segment and a third level to the size of a single speech act, or dialogue move. The definition of the categories is solely based on local information, which makes them domain independent.

It is important to realize that the model is based on a number of assumptions, which are known not to hold for all kinds of human dialogues. The model is thus not intended as a general dialogue model, but for a restricted class of dialogue situations. It was originally developed for teletyped NLI dialogues, but has recently been shown to be applicable also for spoken NLI dialogues in similar applications.

The basic discourse segment unit is an initiative-response-unit (IR-unit), i.e. an initiative (which opens a segment) and a response (which closes the segment). Recursive embeddings of IR-units are allowed. An intitiative is basically a directive in standard speech-act terminology. This can be seen as an act where the speaker introduces a goal, such as a perameter requiring a value, and the expectation that it shall be satisfied. A response is an act where the goal is satisfied, i.e. the value is given or the expectation is eliminated by other means. The IR-units are sub-classified along a number of different dimensions (see below).

The two basic categories of the model is turn and move. A turn is a continuous period of the dialogue where one participant speeks. It is also that part of the dialogue where one participant has the floor. This means that he has the right, and sometimes the obligation, to say something. In human dialouges there is a need for several distinctions, such as utterance (=continuous period of speech by one person) and turn (=continuous period where one person holds the floor). However, in most NLI-systems, a turn is co-extensive with an utterance by user or system, since interruptions do not occur (or can be treated as taking place 'outside' the dialogue).

Each turn consists of one or more moves. A move consists of one (speech) act. It has been pointed out by Levinson (Levinson, 1981) and others, that different speech-acts within an utterance need not be separately assigned to different subparts, but that one part can perform more than one such act. A rather strong assumption of the LINDA/LINLIN model is that this need not be true for man-computer dialogues.

Other similar models, for instance, LOKI (Wachtel, 1986) and SUNDIAL (Bilange, 1991), use also an intermediate analysis level, characterized by having a common topic, e.g. an object whose properties are discussed over a sequence of segments. This is, however, not used in the present model. The reason for this is that we found in our empirical analysis the notion of topic problematic (Ahrenberg, Jönsson, & Dahlbäck, 1990) by being too simple. A sequence of segments may be connected in a number of different ways, e.g. by dealing with one object for which different properties are at issue. But it may also be the other way around, so that the same property is topical, while different objects are talked about. Therefore, instead of using a single notion of topic, a nomber of topical parameters are distinguished, e.g. CurrentObject, CurrentAttribute and others (se below).

Another topic related aspect concerns the topical domain. Three basic such domains are identified; task related (T), discourse related (D), and situation related (S). These can be further sub-divided as necessary for particular applications or situations. In for instance a travel information system where the user also can order tickets, the task will be sub-classified into obtaining information and placing an order.

2 Coding for the LINLIN dialogue model

The initial step in the analysis is assigning the IR-structure. This is then used as the backbone for the further analysis as described below.

2.1 Dialogue structure

- I: If the move is seen as introducing a goal it is scored as an initiative,
- R: If a move is a goal-satisfying move, it is scored as a response.

The initiatives and responses are then connected into IR-units, building a binary tree of such units.

2.1.1 Subclassifying the dialogue structure

To specify the functional role of a move we use the parameters Type and Topic.

- Type corresponds to the illocutionary type of the move. In simple service systems two sub-goals can be identified (Hayes & Reddy, 1983, p. 266): 1) "specify a parameter to the system" and 2) "obtain the specification of a parameter". Initiatives are categorized accordingly as being of two different types:

- Update, U: where users provide information to the system
- Question, Q: where users obtain information from the system.
- Answer, A: Responses from the system or answers to clarification requests.
- Discourse opening, DO: Discourse management moves initiating a discourse such as *Welcome to WingHolidays. What can we do for you?*
- Discourse continuation, DC: Discourse management moves for continuing the discourse such as *Can I help you with anything more?*
- Discourse ending, DE: Discourse management moves ending discourse such as *Bye.*

The last three need not be included in a full binary IR-unit.

- Topic describes which knowledge source to consult. In information retrieval applications three different topics were used
  - Task, T: the database for solving a task
  - System, S: acquiring information about the database, system-related,
  - Dialogue, D: the ongoing dialogue
  - Order, O: If the background system allows ordering of a specified item.

2.2 Focus structure

In information retrieval dialogues the most common user initiative is request for information from the database. Users specify a database object, or a set of objects, and ask for concept information, e.g. the value of a property of that object or set of objects. Two focal content parameters, termed Objects and Properties, account for the information structure of a move (query).

- Objects denote a set of primary referents
- Properties a complex predicate ascribed to this set, cf (Ahrenberg, 1987).

These are focal parameters in the sense that they can be in focus over a sequence of IR-units.

The focus structure depends on how the information in the user initiative and the answer provided from the database specify the values to the focal parameters Objects and Properties. A move can fully specify both Objects and Properties. However, many utterances provide only a partial specification to the focal parameters, which means that contextual information is needed to make them fully specified. Our hypothesis is that the values of Objects and Properties for a previous segment provide an initial local context for the next segment. These values are changed (or made more specific, as the case may be) with values from thenew initiative and the ensuing response from the background system.

The coding scheme needs to account for three types of specifications, termed FS (FullySpecified), LC (LocalContext) and GC (GlobalContext).

- Fully Specified, FS: denotes utterances which are fully specified or can not be further specified using context information. Thus, not only correct utterances, are considered FS but also vague or erroneous utterances.

- Local Context, LC: is used for user utterances which can be specified as regards Properties or Objects from the local context. The information should be found in local focus, i.e. the current segment. This means that focal information from the previous IR-node provides the correct information.
- Global Context, GC: denote underspecified specified utterances that cannot be specified from local focus.

2.3 An example illustrating the coding scheme

To illustrate the coding scheme consider the utterance *How rust prone is Volvo 244?*. This is coded $Q_T FS$[1] as it is an initiative of type Question querying information from the background system, i.e. Topic T and both Objects and Properties are specified. Let us assume that the next user utterance after the systems answer is *Mercedes 200*. This is coded $Q_T LC$ as it is a task-related question where information on Properties, i.e. rust prone, is needed and found in local focus.

Another example is *which 10 car models are most spacious*, where the provided aspect of the Property, i.e. spacious, is ambiguous. However, this is also coded $Q_T FS$ as context information would not further disambiguate the utterance. A dialogue manager would initiate a clarification subdialogue such as: *Information on space is either coupe or boot. Please be more specific.* This is coded $Q_D FS$. The utterance is Topic D because the requested information is not to come from the background system or system information base, but from the ongoing dialogue with the user. Thus, a user response such as *best coupe* would be coded $Q_D FS$ as it provides dialogue information. Both clarification subdialogue moves are coded FS as they are fully specified in the context in which they are uttered, i.e. best coupe is FS in a Topic D subdialogue. The values for the focal parameters are updated and holds the required information. A special case further illustrates this. It occurs when the answer is too large to print on the screen. In such cases the system initiates a clarification subdialogue, coded $Q_D FS$, asking the user to restrict the number of items to be printed, for example, *There are 76 car models which satisfy your requirements.* CARS *normally only shows 25 cars at a time. Do you want to see them all?*.

An important user initiative which do not depend on the values to Objects or Properties is Q_S[2]. These are initiatives where the user explicitly request information about the system, such as *What does rust 5 mean?*.

The code A_S is used for system answers which provides information about the database system. Information can be provided on various aspects on what type of information there is in the database and what type of questions that can be used to elicit this information. Typical examples are: CARS *can not answer questions on the shape of the car models* and *WingHoliday arranges trips to the following places in Greece: Ikaria, Lefkada, Naxos, Crete, Samos and Tolon.*

2.4 Illustrating the dialogue model

To illustrate the behaviour of the customized Dialogue Manager consider the dialogue fragment in figure 1. Initially, the Dialogue Manager creates an instance of a dialogue-node, to capture the whole dialogue, which in turn creates an IR-node, IR_1, and inserts it in the dialogue tree. IR_1 creates an instance of a move-node which interprets the first utterance and integrates it into the dialogue tree making the information available to IR_1. The first utterance, U1, is a task-related user initiative,

[1]For brevity, when presenting the dialogue grammar, the Topic of a move is indicated as a subscript to the Type.

[2]There is no need to code the specification of the focal parameters for such user initiatives or answers from the system. However, they can be coded FS as that category is least demanding.

U1: show mercedes
S2: Wait...
Cars can answer questions about a number of car models in regard to make, model designation, year, country of manufacture, susceptibility to rust, and size class, as well as questions about cost, driving and collision safety, interior (cabin) and baggage (boot) space, and technical data.
Any special requests?
U3: cost and space
S4: Wait...
Checking...

Manufacturer	Model	Year	price	cabin	boot
Mercedes	200	1982	66400	4	370
Mercedes	200	1985	90000	4	370
Mercedes	190	1988	149000	3	320
Mercedes	200	1988	180300	4	370

Figure 1: From the CARS corpus, dialogue 1.

a Q_T[3], which correctly partly specifies an object Objects=(Manufacturer: *Mercedes*, Model: , Year:). However, it does not specify domain concept information[4].

Thus, IR_1 initiates a clarification subdialogue by creating a new IR-node, IR_{11}, and connects this to IR_1. The action plan of IR_{11} creates an instance of a system move-node. From the lack of value to the Properties parameter, IR_{11} concludes that the user needs to provide domain concept information and S2 is generated. The system initiative S2 consists of one part providing system information and one part providing an explicit request for user information. The next action of IR_{11} is to create an instance of a user move-node which interprets the response from the user. IR_{11} is an IR-node of type Q_D/A_D which means that a discourse-related question should be followed by a discourse-related answer. The first part, the Q_D, was generated by the system. The second part, A_D, carries the information that only an answer is regarded as a valid response. The Aspect slot is used to hold the parameter for which the system wants an answer and the Value slot is used for the user's answer. If the user answers correctly, the values for Properties in the initiating IR-unit are updated. A Q_D/A_D-unit is identified from the type information, i.e. the Type of the response from the user is A. Otherwise the user move is regarded as not being an answer to the system request. In this case, the user, in U3, provides an answer to the clarification request, an A_D. The user move-node is connected to IR_{11} and updates its Properties parameter with the values *cost* and *space*.

This information is made available to IR_1 which can submit a request to the database access module. The response from the database manager updates the Objects parameter of IR_1 to the four objects presented and also updates the Properties parameter. Thus, the Objects parameter now consists of a set of four objects, not one as in the initial request, U1. IR_1 creates an instance of a generate-move, an A_T to present the result to the user, S4. Finally, IR_1 signals to the top-level dialogue node, D, that it is done and the interaction proceeds from the D-node which again instantiates an IR-node, IR_2, ready to interpret a new user initiative. Initially IR_2

[3]For brevity, when presenting the dialogue grammar, the Topic of a move is indicated as a subscript to the Type.

[4]One could argue that the user asks for all information. This would, however, be too much information to display and thus a similar clarification subdialogue would be initiated from the system.

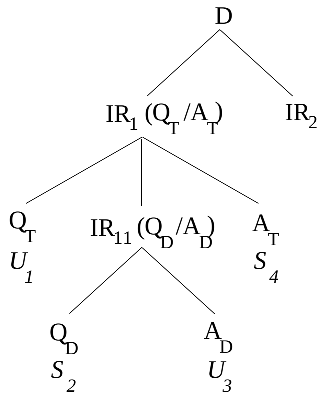

Figure 2: The dialogue tree for the dialogue in figure 1

has values to its context parameters copied from IR_1 as described above. Figure 2 shows the resulting dialogue tree.

References

Ahrenberg, L. (1987). *Interrogative Structures of Swedish. Aspects of the Relation between grammar and speech acts.* Ph.D. thesis, Uppsala University.

Ahrenberg, L., Jönsson, A., & Dahlbäck, N. (1990). Discourse representation and discourse management for natural language interfaces. In *Proceedings of the Second Nordic Conference on Text Comprehension in Man and Machine, Täby, Sweden.*

Bilange, E. (1991). A task independent oral dialogue model. In *Proceedings of the Fifth Conference of the European Chapter of the Association for Computational Linguistics, Berlin.*

Dahlbäck, N. (1991). *Representations of Discourse, Cognitive and Computational Aspects.* Ph.D. thesis, Linköping University.

Grosz, B. J. & Sidner, C. L. (1986). Attention, intention and the structure of discourse. *Computational Linguistics*, *12*(3), 175–204.

Hayes, P. J. & Reddy, D. R. (1983). Steps toward graceful interaction in spoken and written man-machine communication. *International Journal of Man-Machine Studies*, *19*, 231–284.

Jönsson, A. (1993). *Dialogue Management for Natural Language Interfaces – An Empirical Approach.* Ph.D. thesis, Linköping University.

Levinson, S. C. (1981). Some pre-observations on the modelling of dialogue. *Discourse Processes*, *4*, 93–116.

Wachtel, T. (1986). Pragmatic sensitivity in nl interfaces and the structure of conversation. In *Proceedings of the 11th International Conference of Computational Linguistics, University of Bonn*, pp. 35–42.