Joachim Walter

Approaches to dialogue system development: TrindiKit vs. CSLU toolkit

Abstract: A number of toolkits are available that assist developers with the complex issues involved in constructing a spoken dialogue system. In this paper, I discuss two approaches: the Information State based approach and the finite-state approach.I present the architecture and toolkit for building dialogue managers currently beeing developed in the TRINDI project based on the notions of information state and dialogue move engine. I will also evaluate projects involving CSLU's RAD (Rapid Application Developer) and compare their finite-state approach with TRINDI's information-state based approach.

Hauptseminar: Dialogue management SS 2002
Dr. Ing. Ivana Kruijff-Korbayova
Computational Linguistics
Universität des Saarlandes, Germany

Content

1. Introduction

A number of toolkits are available that assist developers with the complex issues involved in constructing a spoken dialogue system. In this paper, I discuss two approaches: the Information State based approach and the finite-state approach.
I present the architecture and toolkit for building dialogue managers currently beeing developed in the TRINDI project based on the notions of information state and dialogue move engine.[1] I will also evaluate projects involving CSLU's RAD (Rapid Application Developer) and compare their finite-state approach with TRINDI's information-state based approach.[2]

2. Approaches to Dialogue Management

2.1 Information State based Approach

Staffan Larson and David Traum discuss a candidate model for best practice in the development of the dialogue management component of a spoken dialogue system.[3] They propose a view of dialogue management functions in terms of information state. Key to this approach is identifying the relevant aspects of information in dialogue, how they are updated, and how updating processes are controlled.

2.1.1 Information State Theory

The term Information state of a dialogue represents the information necessary to distinguish it from other dialogues, representing the culmative addidtions from previous actions in the dialogue, and motivating future actions. For example, statements generally add propositional information: questions generally provide motivation for others to provide specific statements.

---

[1] Staffan Larsson and R. Traum, David: *Information state and dialogue management in the TRINDI dialogue move engine toolkit*, 2000, Department of Linguistics, Göteborg University, Sweden

[2] Michael McTear: *Modelling spoken dialogues with state transition diagrams*: Experiences with the CSLU toolkit. In Proceedings of the 5th International Conference on Spoken Language Processing, Sydney, Australia, 1998.

[3] Staffan Larsson and R. Traum, David: *Information state and dialogue management in the TRINDI dialogue move engine toolkit*, 2000, Department of Linguistics, Göteborg University, Sweden

Information state is also referred to by similar names, such as „conversational score“, or „discourse context“ and „mental state“. Key to the approach is the notion of UPDATE of information state, with most updates related to the observation and performance of DIALOGUE MOVES.

It is important to distinguish information state approaches to dialogue modeling from other, structural, dialogue state approaches. These latter approaches conceive a „legal“ dialogue as behaving according to some grammar, with the state representing the results of performing a dialogue move in some previous state, and each state licensing a set of allowable next dialogue moves. The „information“ is thus implicit in the state itself and the relationship it plays to other states. On the other hand, it is very easy to model dialogue state as information state: the information is the dialogue state, itself. In the rest of this section, I will present the aspects of information state in a little more detail.

2.1.2 Informational Components

An information state theory of dialogue modeling consists of a description of the informational components of the theory of dialogue modeling, formal representations of these components, a set of dialogue moves that will trigger the update of the information state, a set of update rules, that govern the updating of the information state, given various conditions of the current information state, and an update strategy for deciding which rules to select at a given input.

It may be useful to distinguish components of information state into static and dynamic aspects. Examples of static information state components could include domain knowledge, knowledge of dialogue conventions, or similar sorts of information. This depends on the type of dialogue being modeled.

An example information state could be a simplified version of the dialogue game board proposed by Ginzburg.[4] There is some information assumed to be private (including beliefs (BEL), and an agenda of actions to perform in the dialogue (AGENDA)) and some that is assumed to be shared (propositions assumed to be shared beliefs (BEL), questions under discussion (QUD), and the latest dialogue move performed (LM)).

---

[4] Ginzburg, J.:*Dynamics and the semantics of dialogue*. In: Logic, language and computation, vol 1, 1996

2.1.3 Formal Representations

After defining the aspects of the dialogue structure to model, the question arises as to how to model them. As an example, consider an aspect of information state such as actions to be performed in the dialogue. This includes an agenda, a plan, or other bundle of intentions. On the other hand, there is the question of representing a list: Should it be a FIFO queue, a LIFO stack, or some more open structure ? The example information state shown in Figure 1 is represented as a record.[5]

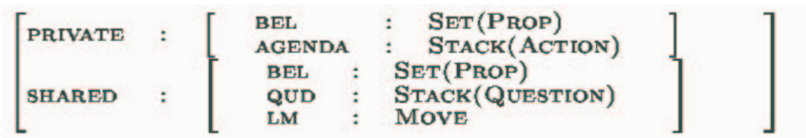

Figure 1: Information state represented as a record (Cooper & Larson)

2.1.4 Dialogue Moves

Dialogue moves are meant to serve as an abstraction between the large number of different possible messages that can be sent and the types of update to be made on the basis of performed utterances. Dialogue moves can also provide an abstract level for content generation. The set of dialogue moves to choose is also influenced by the task of language interpretation. Another problem is how to choose the inherent multi-functionality of utterances. Dialogue moves are often seen as something like speech-acts in the sense of Searle.[6]

2.1.5 Update Rules

Update rules formalize change in information state as the dialogue progresses. Each rule has a set of conditions and a set of effects. Effects are changes that will be made to the information state when the rule has been applied.

[5] Cooper, Robin & Larsson, Staffan, 1999: *Dialogue moves and information states*. In: Proceedings of the third international workshop on computational semantics.
[6] Searle, John R. 1969: *Speech acts*. New York: Cambridge University Press

Figure 2 shows the update rule integrateSysAsk, the rule for adding a question to QUD if an ASK move has been performed. The rule has two conditions: that the latest move was of type ASK, and that the top of AGENDA was the action of raising a question, the effects are to pop this item from AGENDA, and push onto the QUD, the question that is the content of both the RAISE AGENDA item and the ASK dialogue move.

$$
\begin{array}{ll}
\text{U-RULE:} & \textbf{integrateSysAsk} \\
\text{PRE:} & \left\{ \begin{array}{l} \text{val(SHARED.LM, ask(sys,}Q\text{))} \\ \text{fst(PRIVATE.AGENDA, raise-question(}Q\text{))} \end{array} \right. \\
\text{EFF:} & \left\{ \begin{array}{l} \text{push(SHARED.QUD, }Q\text{)} \\ \text{pop(PRIVATE.AGENDA)} \end{array} \right.
\end{array}
$$

Figure 2: integrateSysAsk

2.1.6 Update Strategy

Along with the set of update rules, a strategy for how to apply the rules is needed. Some types of update strategies are:

1. *Take the first rule that applies (iteratively until no rules apply)*
2. *Apply each rule (if applicable) in sequence*
3. *Apply rules according to class*
4. *Choose among applicable rules using probabilistic information*
5. *Present choices to user to decide (for development modes)*

A short question-answer exchange is illustrated in Figure 3. Before the exchange, the system has an AGENDA item ro raise the question about the user's destination. This meets the conditions for the update rule that selects an ASK move. After the system utterance, the update algorithm 1 will first apply the rule shown in Figure 2. After the user utterance, a rule will check that the answer matches the question topmost on QUD, and will then pop the question off to QUD, and integrate the proposition resulting from applying the question to the answer into the shared beliefs.

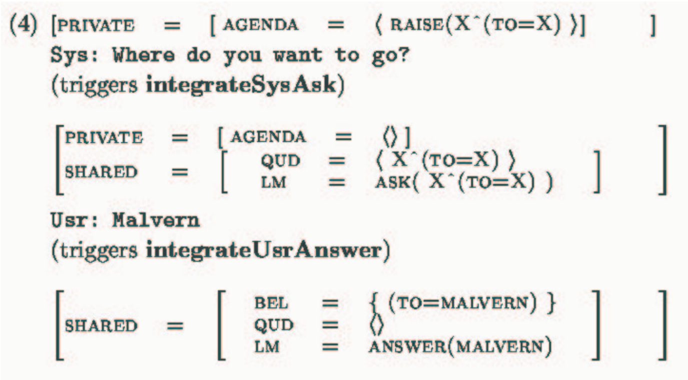

Figure 3: Question-answer exchange using algorithm 1

2.2 Finite-State dialogue models

Approaches to dialogue management can be broadly classified into finite-state methods, on the one hand, and self-organising or locally-managed approaches on the other.[7] In a finite model, the dialogue structure is represented in the form of a state transition network in which the nodes represent the system's questions and the transitions between the nodes determine all possible paths through the network, this specifying all legal dialogues.[8]

2.2.1 Criticisms of finite-state dialogue models

Finite-state models have been critized because of their inflexibility as well as their inability to cope with the requirements of more complex dialogues. Taking the example of a simple travel inquiery system, a natural order for the system's question might be: *destination* > *origin* > *date* > *time* .

When answering the system's question concerning destination, the user might reply with a destination as well as the departure time. A finite-state based system would simply progress through its set of predetermined questions, ignoring or

[7] Fraser, N.M. and Dalsgaard P.: *Spoken Dialogue Systems: A European Perspective,* Proceedings of International Symposium on Spoken Dialogue, Philadelphia, 1996

[8] Michael McTear: *Modelling spoken dialogues with state transition diagrams*: Experiences with the CSLU toolkit. In Proceedings of the 5th International Conference on Spoken Language Processing, Sydney, Australia, 1998.

failing to process the additional information and and then asking an irrelevant question concerning the departure time.

2.2.2 Empirical studies of finite-based dialogue systems

The strengths and weaknesses of system-led dialogue control and finit-state models have been investigated in several empirical studies.[9] The conclusion from these studies is that system-led dialogue using state transistions would appear to be suitable for simple tasks with a flat menu structure and a small list of options, bringing also the advantage of less complex spoken language and dialogue modelling technology. The lack of flexibility and naturalness may be justified as a trade-off against these technological demands.

3. Dialogue managemnt: TRINDIKit vs. CSLU ToolKit

3.1 TRINDIKit

3.1.1 Dialogue Move Engine (DME)

The implementation of a theory using the information-state approach described in Section 2.1 is named a Dialogue Move Engine (DME). Its main functions are updating information state based on the observance of moves and selecting moves to be performed. The DME, together with some connective material, forms the dialogue management and discourse tracking aspects of dialogue system. For the following functions, the dialogue system would need additional modules:

- user interface	to receive input from and present output to the user.
- interpretation	to calculate from the input which dialogue moves have been performed, adding these to a special latest move (LM) part of the information state.

[9] Hone, K.S., and Baber, C.: *Using a simulation method to predict the transaction time effects of applying alternative levels of constraint to utterances within speech interactive dialogues*, ESCA Workshop on Spoken Dialogue Systems, Vigso Denmark, 1995

- generation to take the contents of the special next move part of information state, and produce the output.
- control to wire together the other modules, either serially or in parallel

Larsson et al. developed a DME toolkit called TrindiKit (Task Oriented Instructional Dialogue), which provides the basic architecture as well as facilities for implementing theories of informatuion state.[10] The general architecture of TrindiKit is shown schematically in Figure 4.

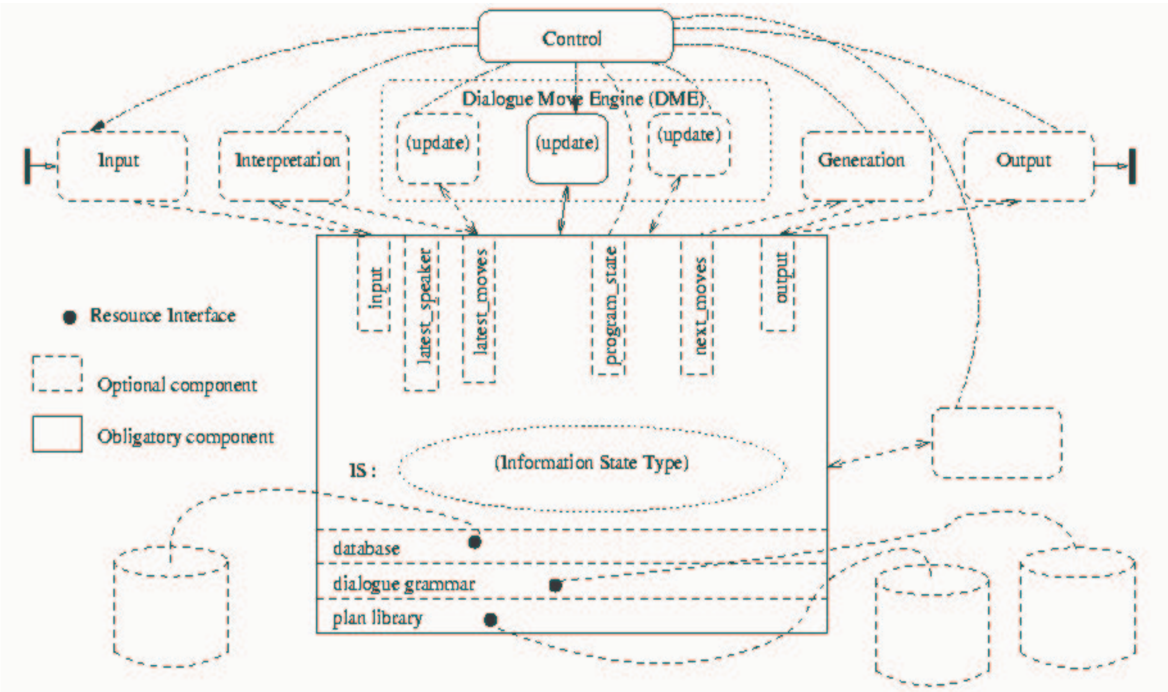

Figure 4: TrindiKit architecture

3.1.2 Implementations using TrindiKit

A number of systems are currently beeing developed using the TrindiKit. I will show one in more detail: GoDiS[11], which uses an extension of the information

[10] Larsson, Staffan, Bohlin, Peter, Bos, Johan & Traum, David, 1999: *TrindiKit manual*. Tech. Rept. D2.2 – Manual. Trindi.
[11] Bohlinm Peter, Cooper, Robin, Engdahl, Elisabet & Larsson, Staffan, 1999: *Information states and dialogue move engines*. Pages 25-31 of: Proceedings of the ijcai99 workshop.

state theory used as an example in Section 2.1.

3.1.2.1 GoDiS

GoDiS uses fairly simple algorithms for control, update and selection modules, keyword-based interpretation and template-based generation. The system distinguishes 8 dialogue move types: *ask, answer, repeat, request_repeat, greet, goodbye, thank* and *quit*. The main division in the information state is between information that is private to the system and that which is assumed to be shared between the dialogue participants. The SHARED field is divided into three subfields: set of propositions which the agent assumes for the sake of the conversation, a stack of questions under discussion (QUD) and information about the latest move (speaker, move type and content). The PRIVATE subfield contains four subfields: BEL, AGENDA, PLAN and TMP. The update rules in GoDiS include an ability to perform several kinds of accomodation when the appropriate expected structure are not in the QUD or PLAN subfield. A sample dialogue handled by GoDiS is hown in Figure 5.

```
Sys: welcome to the travel agency!
Usr: flights to paris
Sys: what city do you want to go from?
Usr: london
Sys: when do you want to travel?
Usr: in april, as cheap as possible
Sys: do you want a return ticket?
Usr: yes please
Sys: the price is $300

PRIVATE = [ BEL    = {}
            AGENDA = ⟨⟩
            PLAN   = ⟨ RAISE(R^(RETURN=R)),
                       RAISE(M^(MONTH=M)),
                       RAISE(C^(CLASS=C)),
                       RESPOND(P^(PRICE=P)) ⟩
            TMP    = (same as SHARED) ]
SHARED  = [ BEL = {(TO=PARIS),(HOW=PLANE)}
            QUD = ⟨X^(FROM=X)⟩
            LM  = ASK(SYS,Y^(FROM=Y)) ]
```

Figure 5: sample dialogue by GoDiS

3.1.2.2 Other TRINDIKit systems

The EDIS system[12] ueses a notion of information state based on Poesio & Traum[13]. The informational components consists of a common ground part, a semi-public part, and a private part as with GoDiS. The common part includes four types of information: OBL, SCP, DH and COND. The semi-public part are a collection of discourse units, which represent coherent bundles of information. Private information includes the intensions of the agent being modeled.

The MIDAS system uses the DRS structures of DRT[14] as a major component of its information state. Using a simplified version of the theory proposed in Poesio and Traum, the root DRS represents the assumed common ground, which will subordinate DRS representing events mentioned in the dialogue as well as DRS for tracking grounding. An example of a MIDAS information state can be seen in Figure 6.

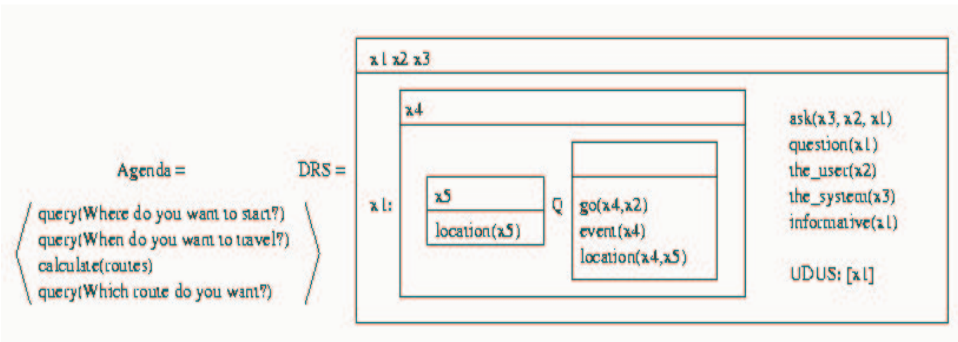

Figure 6: MIDAS information state

[12] Matheson, Colin, Poesio, Massimo & Traum, David, 2000: *Modelling grounding and discourse obligations using update rules*, In: Proceedings of the first conference of the north american chapter of the association for computational linguistics.

[13] Poesio, Massimo & Traum, David R, 1998: *Towards an axominatization of dialogue acts*. In: Proceedings of twendial'98, 13th twente workshop on language technology: Formal semantics and pragmatics of dialogue

[14] Kamp, H. & Reyle, U. 1993: *From discourse to logic*. Dodrecht: D.Reidel

3.2 CSLU ToolKit

3.2.1 Directory assistance

There are many directory inquiry systems. I will only present the part of a directory inquiery dialogue which the systems attemps to elict the name of the person to be called. To complete this task, a first and a last anme are required. So, the task can be completed in a single step like *Request First and Last Name* or in a series of steps like *Request Surname* > *Request Spelling of Surname* > *Request First Name* > *Confirm First and Last Name*. In a system implemented using RAD at the University of Ulster, the dialogue was designed to maximize transaction success at the possible expense of transaction. The main characteric of this task and ist dialogue model is that there is a minimal amount of information to be elicited. This task lends itself easily to implementation using a finite-state dialogue model. A finite-state model can also be used for similar tasks such obtaining weather forecasts, football scores, ordering items from a catalogue, ormaking simple bank transactions.

3.2.2 Questionnaires

Dialogues for questionnaires are also highly structured, so that the user can be constrained through carefully designed prompts to produce an acceptable range of responses. Finite-state models can be used for tasks such as eliciting a person's personal details for financial transactions or obtaining information for insurance quotes.

3.2.3 Travel inquiries

For travel inquiries there is also in the simplest case a fixed set of parameters to be acquired and a natural sequence for their elicitation. E.g. *destination* > *origin* > *date* > *time*. Such a system can also be implemented using a finite-state model.

4. References

Bohlinm Peter, Cooper, Robin, Engdahl, Elisabet & Larsson, Staffan, 1999: *Information states and dialogue move engines*. Pages 25-31 of: Proceedings of the ijcai99 workshop.

Cooper, Robin & Larsson, Staffan, 1999: *Dialogue moves and information states*.In: Proceedings of the third international workshop on computational semantics.

Fraser, N.M. and Dalsgaard P.*: Spoken Dialogue Systems: A European Perspective*, Proceedings of International Symposium on Spoken Dialogue, Philadelphia, 1996

Ginzburg, J.:*Dynamics and the semantics of dialogue*. In: Logic, language and computation, vol 1, 1996

Hone, K.S., and Baber, C.: *Using a simulation method to predict the transaction time effects of applying alternative levels of constraint to utterances within speech interactive dialogues*, ESCA Workshop on Spoken Dialogue Systems, Vigso Denmark, 1995

Kamp, H. & Reyle, U. 1993: *From discourse to logic*. Dodrecht: D.Reidel

Matheson, Colin, Poesio, Massimo & Traum, David, 2000: *Modelling grounding and discourse obligations using update rules*, In: Proceedings of the first conference of the north american chapter of the association for computational linguistics.

Poesio, Massimo & Traum, David R, 1998*: Towards an axominatization of dialogue acts*. In: Proceedings of twendial'98, 13th twente workshop on language technology: Formal semantics and pragmatics of dialogue

Searle, John R. 1969: *Speech acts*. New York: Cambridge University Press

Larsson, Staffan, Bohlin, Peter, Bos, Johan & Traum, David, 1999: *TrindiKit manual*. Tech. Rept. D2.2 -Manual. Trindi.

Staffan Larsson and R. Traum, David: *Information state and dialogue management in the TRINDI dialogue move engine toolkit*, 2000, Department of Linguistics, Göteborg University, Sweden

Michael McTear: *Modelling spoken dialogues with state transition diagrams: Experiences with the CSLU toolkit*. In Proceedings of the 5th International Conference on Spoken Language Processing, Sydney, Australia, 1998.