

# GoDiS - Issue-based dialogue management in a multi-domain, multi-language dialogue system\*

Staffan Larsson and Stina Ericsson

{sl, stinae}@ling.gu.se

Department of linguistics, Göteborg University

Box 200, SE-405 30 Göteborg, Sweden

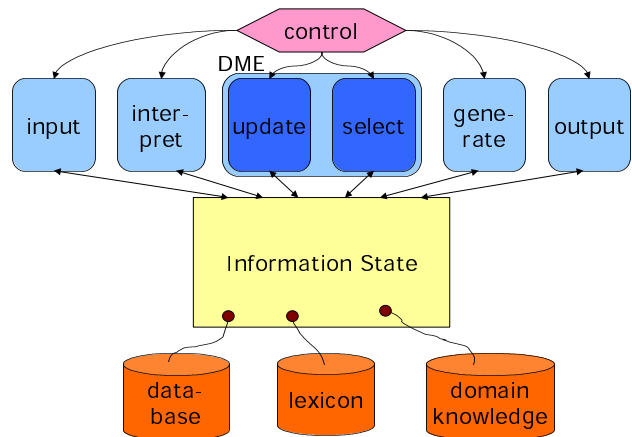
## 1 Introduction

GoDiS (Gothenburg Dialogue System) is an experimental dialogue system utilizing dialogue management based on Ginzburg's concept of Questions Under Discussion (QUD). GoDiS is implemented using the TrindiKit, a toolkit for implementing dialogue move engines and dialogue systems based on the Information State approach. While originally built for fairly simple information exchange dialogue, it is being extended to handle action-oriented and negotiative dialogue. A revised version has recently been implemented (Larsson(2002)).

One of the goals of the information state approach is to encourage modularity, reusability and plug-and-play; to demonstrate this, GoDiS has been adapted to several different dialogue types, domains, and languages, including information exchange dialogue in travel agency and autoroute domains, and action-oriented dialogue when acting as an interface to a mobile phone or VCR. GoDiS has also been enhanced with speech input and output, and is able to switch between Swedish and English in an ongoing dialogue.

## 2 GoDiS architecture

The GoDiS architecture shown below is an instantiation of the general TrindiKit architecture.



The components of the architecture are the following:

- the Information State (IS)
- domain-independent modules, operating according to module algorithms
- the Dialogue Move Engine (DME), consisting of two modules (Update and Select); the DME is responsible for updating the IS based on observed moves, and selecting moves to be performed by the system.
- a controller, wiring together the other modules, either in sequence or through an asynchronous mechanism.
- three domain-dependent resources: Database, Lexicon, and Domain Knowledge

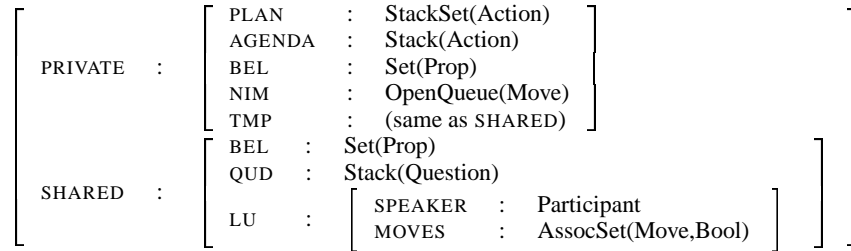


Figure 1: GoDiS information state type

The TIS is accessed by modules through conditions and operations. The types of the various components of the TIS determine which conditions and operations are available.

### 3 GoDiS information state

The notion of information state we are putting forward here is basically a modified version of the dialogue game board proposed in (Ginzburg(1996)). We represent information states of dialogue participants as records of the type in figure 1.

The main division in the information state is between information which is private to the agent and that which is shared between the dialogue participants. The PLAN field contains a dialogue plan, i.e. is a list of dialogue actions that the agent wishes to carry out. The TMP field keeps track of shared information that has not yet been grounded. QUD is a stack of questions under discussion. These are questions that have been raised and are currently under discussion in the dialogue.

### 4 Features

Adaption to new domains and languages is simplified by the clear separation of domain-dependent and domain-independent components. Given a DME which can handle the dialogue type required by the domain, e.g. information exchange, adaption to a new domain amounts to building resources for that domain. Adaption to a new language only requires adaption of the lexicon resource. Also, resources can be switched on-line, in principle allowing the system to change domain and language during dialogue.

*Question and task accommodation* allows GoDiS to adapt to the user's utterances, e.g. allowing the user to answer questions before they have been

asked, and providing information before any task has been established. Given a simple schema specifying the reimplementaion of menu-based user interfaces in the form of dialogue plans, these mechanisms allow a more natural and flexible interface than a simple menu.

The information state update rules include rules for question and task accommodation, as well as rules for handling grounding and rules for integrating the latest move with the IS.

As an example, the u-rule for accommodating questions (which is necessary when the user answers a question that the system has yet to ask) is shown below.

```

U-RULE: accommodateQuestion
PRE: {
  val(shared.lm.speaker, usr)
  in(shared.lm.moves, answer(A))
  empty(shared.qud)
  in(private.plan, raise(Q))
  domain: relevant_answer(Q, A)
EFF: {
  del(private.plan, raise(Q))
  push(shared.qud, Q)

```

Accommodation allows the system to handle dialogue such as this one:

```

system> Phone interface speaking, how can
I help you?
user > Jim
system> OK, Jim. Do you want to activate,
divert, add a new number, change a
name, or delete a name?
user > add new number
system> What number?

```

### References

- Ginzburg, J. (1996). Interrogatives: Questions, facts and dialogue. In *The Handbook of Contemporary Semantic Theory*. Blackwell, Oxford.
- Larsson, S. (2002). *Issue-based Dialogue Management*. Ph.D. thesis, Göteborg University.