
Annotations for Instructional texts

Annie Zaenen Staffan Larsson

Distribution: PUBLIC



Task Oriented Instructional Dialogue

LE4-8314

Deliverable D1.2

February 1999

Task Oriented Instructional Dialogue

Gothenburg University

Department of Linguistics

University of Edinburgh

Centre for Cognitive Science and Language Technology Group, Human Communication
Research Centre

Universität des Saarlandes

Department of Computational Linguistics

SRI Cambridge

Xerox Research Centre Europe

For copies of reports, updates on project activities and other TRINDI-related information,
contact:

The TRINDI Project Administrator
Department of Linguistics
Göteborg University
Box 200
S-405 30 Gothenburg, Sweden
trindi@ling.gu.se

Copies of reports and other material can also be accessed from the project's homepage,
<http://www.ling.gu.se/research/projects/trindi>.

©1999, The Individual Authors

No part of this document may be reproduced or transmitted in any form, or by any means,
electronic or mechanical, including photocopy, recording, or any information storage and
retrieval system, without permission from the copyright owner.

Contents

1	Introduction	5
2	Aim of the annotation schemes	7
3	Dimensions of the annotation	9
4	The annotation schemes	15
4.1	The conceptual annotation schemes	15
4.1.1	The subgenre mark-up	15
4.1.2	Discourse Structure mark-up	16
4.1.3	The activity mark-up	17
4.2	The lexico-syntactic mark-up	19
4.3	Annotating instructional text with information states	21
4.3.1	Dialogue vs. text	21
4.3.2	Information states in text generation	21
4.3.3	Text structure and information state annotation	23
4.3.4	The annotated text	24

Chapter 1

Introduction

The goal of this deliverable is slightly different from that of deliverable 1.1 [Cooper *et al.*(1999)]. In the instructional text module of the project, we start from existing maintenance or repair manuals that are available in a traditional sequential format as written text and study ways to present the same content in interactive ways. The interactivity can take several forms. We intend to pursue two: hypertext linking for explanations and oral presentation for procedures.

For some parts of the text we hypothesize that the user could profit from a presentation in which he can ask for additional information about what he is processing. This additional information should be delivered in a form that takes into account what has already been processed. To do this, one needs to build a representation of the portions of the text processed up to the moment the information is asked for. Of course manuals tend not to be read from cover to cover. We can not assume that everything that precedes a particular passage has already been read. In fact what can be assumed to be read will depend in part on the type of reader. A beginner might read the introductory chapters which give a general overview and then go to the chapter he is interested in. A more experienced user, might go immediately to the subsection that contains the passage that she is interested in. We assume that there is a way to track which material the user has consulted within a specific consultation session. That material will be assumed to have become part of the common beliefs of the user and the “manual presenter” which act as the participants in the dialogue here.

The presentation of additional information seems to us to be most useful in the descriptive parts of the text and less in the purely instructional part. The instructional part, however, seems to be a good candidate for oral presentation. This needs a representation of the content derived from the original text and adequate to generate the same content in oral form. We assume that this is a more complex transformation than just reading the text that was meant to be read.

The general form of the exercise, then, is the following: mark-up the text to 'extract' from it a content representation that allows us to present it in a different form, either orally in dialogue or linked to supplementary information (in e.g. a dynamic hypertext environment).

We distinguish three types of mark-up: a conceptual one indicating the discourse structure; a linguistic one, indicating the elements that we hypothesize contribute to the discovery of the discourse structure, and a third one indicating the relation between information states and discourse structure. The last mark-up is also a first stab at giving an unified account of text generation and dialogue processing. The second mark-up is to be exploited if we want to transform documents from one mode of representation into another in a semi-automatic way. This deliverable is structured so that Chapters 2 and 3 are mainly geared towards the first and second mark-up schemes, while Chapter 4 includes a section on the third scheme.

Chapter 2

Aim of the annotation schemes

The text model should allow us to distinguish coherent instructional texts from less coherent ones. For less coherent text it will be more difficult to determine how it contributes to the content representation that we need to do the document transformations. Coherent text should also insure a more efficient update of the information state of the user.

A second aim is to investigate which parts of an instructional text are good candidates for oral presentation and which can be enriched through links to additional information. This will depend on the type of activities that the text aims to provoke. This requires that we know what the aims of the various subparts of the text are. We postulate that general descriptions meant to help the user understand the functioning of a device will profit from links to additional information but not critically from oral presentation and that instructions (procedures) would profit from oral presentations as such a presentation frees the hands and the eyes of the user for other tasks than reading, i.e. for the maintenance or repair itself. We need to get from the text to an activity model that then in turn will allow us to develop the required interactive model for the appropriated parts of the text. This modelisation could then contribute to the generation of these interactive modes for instructional text.

Within this project, then, instructional texts are marked up to determine three things:

- their internal coherence,
- which information is contained in the various subparts so as to make it possible to adapt the additional information requested to what is already read.
- whether they lend themselves to oral presentation, and if so, which changes they should undergo when presented in that mode.

The second and third issue depends in part on the activities that the various subparts of the text are intended to provoke. Given this, the aim of the annotation schemes is to mark the elements that are relevant

- for determining the information structure of the text,
- for indicating the type of activity aimed at in the different parts of the text.

Chapter 3

Dimensions of the annotation

To annotate the descriptive parts of the text, we combine features of the model proposed in [Polanyi(1988)], the Linguistic Discourse Model (LDM), with aspects of Rhetorical Structure Theory as adapted by [Knott and Dale(1993)].

A text is subdivided in Discourse Constituent Units (DCU). Normally a clause forms the smallest type of DCU but small DCU's can be combined into larger ones.

DCU's come in four types:

- *Sequences*, which are composed of DCU's of the same type. Sequences come in two subtypes:
 - Lists*: the sequence of clauses expresses values of the same function for alternative arguments;
 - Topic chains*: a series of distinct predications about the same entity.
- *Expansions*, which are composed of a main and subordinated DCUs.
- *Structures*, in which there are two elements connected with an explicit logical coordinator such as or, if... then, because. We will distinguish the following subtypes:
 - purpose (to, in order to)
 - cause (because)
 - result
 - motivation (as)
 - condition (if ... then)

As is well known, several connectives can have a purely temporal value or a causal value. We will annotate them as causal if that seems their main function. Purely temporal relations are annotated handled in the activity scheme.

- *Interruptions.* We distinguish the following interruptions:
 - examples
 - digressions
 - restatements

We will indicate the level of embedding of DCU's through a sequence of +'s followed and preceded by a number: a DCU that is an expansion of the previous one will have the same number as the one it expands (its 'nucleus' in RST terms) followed by + followed by another number. The numbers indicate the position of a DCU within a sequence: the first one is numbered 1, the second 2, and so forth. Interruptions will be indicated with a letter instead of a number. The numbering scheme encodes a tree structure formed by the discourse.

The LDM parses a discourse from left-to-right on a clause-by-clause base into a discourse tree. The Discourse Parse Tree (DPT) is formed by attaching incoming material to the accessible nodes. The clausal dcu's form the leaves of the tree. Only the nodes at the right frontier of the tree are accessible but DCU's can be integrated with material that follows them before they get integrated with the already created tree. As far as the tree structure goes, interruptions will be integrated in the same way as expansions.

The tree structure that is created in this way can be seem as reflecting the plan of the section and indeed another way to get to the same structure is to raise questions as part of a plan. This alternative encoding which shows the relation between the discourse structure and the information states as defined in the Gothenburg contribution in [Cooper *et al.*(1999)]. is illustrated in Section 4.3.

Given the aims of this project, it is not enough to characterize textual units, we also want to understand what the formal clues are that allow us to come to the given characterisation. The syntactic structure of the sentences plays some role here and the choice of what has been called 'discourse markers'. According to a conservative view on discourse-markers they are mainly sentence connectives. But as has been noticed in the literature (find ref.) in fact other lexical items can also give information about the rhetorical relation between different propositions. As there is no set list of relevant phenomena available, we start with an experimental list that will most like change in the course of the project. It is our assumption that the information found in these lexico-syntactic tags can be automatically extracted from an adequate sentence parse and later in the project we will experiment with automatic parsing to find this information. We start of with (machine aided) manual tagging because we do not want to depend on the quality of the chosen parser.

The information contained in these mark-up schemes, together with some general assumptions about the knowledge of the intended user, should allow us to articulate how the linguistic information determines the information state of the user and what non linguistic information is assumed to be present. Given some ideas about efficient updates of information states this will allow us to develop a model for text coherence.

To determine which parts of the texts would lend themselves to oral presentation, we need to

determine which type of activities they are intended to provoke. In this part of the enterprise we start from a recent study of [Lux(1998)] who observes that the language found in aircraft maintenance manuals falls in four distinct subgenres:

- *Function Descriptions*
- *Tasks*
- *Warnings*
- *Notes*

These various subgenres can be distinguished by the linguistic devices that they use. They correspond to subparts of the manuals. The sections on generalities, device descriptions and descriptions of operations exhibit one type of language; the procedures another and the warnings are characterised by a third type of sublanguage. The notes do not show any linguistic characteristic different from the general language.

We summarise here Lux's findings, adapting them to English:

in *Function Descriptions*, we find:

- a rich inventory of anaphora and other co-reference indicators;
- most anaphora and their antecedents are subjects, which leads to syntactic parallelism;
- negatives are exceptional
- most sentences are declaratives in the present tense
- passives are frequent, agents are rarely expressed
- the nominal groups are complex.

in *Task Descriptions*, we

- imperatives
- few anaphora
- few coordinations
- nearly no negations
- no impersonal forms
- no passives
- no modal indicators (except in embedded clauses)
- frequent use of verbs indicating a change in position
- few verbs taking clausal complements
- object drop
- drop of the article, especially in prepositional phrases

- frequent use of coordinated adjectives but few coordinated sentences

in *Warnings*, we find

- Negative imperatives
- Deontic modals
- Impersonal verbs expressing deontic modalities
- Manner adverbs
- Nominal and adjectival predications

Notes don't have specific linguistic characteristics.

Our annotation scheme intends to verify and refine these observations for instructional texts. Within this project we stay quite closely to the type of text that was the object of Lux's study: maintenance and repair manuals. It is clear that the model will need to get developed further if one wants to take other material into account take, e.g. textbooks.

We will add the hypotheses of Lux, adapted to English, to our lexico-syntactic mark-up scheme and on the other hand mark the text for the types of sections that correspond *mutatis mutandis* to Luxas classification. In fact in most cases this mark-up might be part of the non textual information that is given for the documents we work on.

The decision to present orally some of the material that is proposed to be read by the user in our manuals, is not only dependent on its classification as a function description, a task list or a warning or note. First of all, it will depend on the environment, e.g. in a noisy environment, an oral presentation might not be feasible. More to the point of this study, the way the material is presented might not lend itself directly to an oral presentation, e.g. the description of a complex device might be better understood when read than when heard. Some material, e.g. procedures lend themselves in principle to oral presentation but they might be described in the text in a way that makes it impossible because such a presentation strains the user's memory too much. In this case the language needs to be adapted to the new mode of presentation. To check for these factors a further type of mark-up is useful, one that helps us construct an activity model corresponding to the text. Again the hypothesised linguistic cues that help us determine the model are added to the lexico-syntactic mark-up.

The core of the type of text that we are studying are procedures. The aim of the activity mark-up is to distinguish between the parts of text that are non procedural and those that are. For the part that is procedural, it should be determined whether the procedure is given in an acceptable form.

A procedure consists of a number of steps to be executed in a fixed order and each describing an action that the agent has to execute given a certain (initial) state of the device, using a specified set of tools and products in view of reaching a certain result state. It is a sequence of actions. The sequence can be subdivided in preparatory actions, the procedure proper and "wrap-up" actions. Each action involves an agent, the addressee of the manual, an object acted upon (might be too strong), instruments with which to perform the action (if not specified, the agent's body can be considered to be the instrument), and possibly subsidiary

objects. An action changes the state of the device, or subdevice, into a new state. For the action to be appropriate the device has to be in a given initial state (e.g. dirty) and end up in a final state. Each action takes place in time and the conditions of anteriority, posteriority or temporal overlap are crucial for the right execution of the tasks.

Our assumption is that easy to follow procedures should be simple, i.e. the user should be asked to perform only one action at once, that their conditions of completion should be clear and that the mode of execution should be unambiguously specified. We will argue that only action descriptions of that type can be presented reliable in an oral mode but also that only action descriptions of this type are easy to execute regardless of the mode of presentation of the material. We will code the instructional text for these characteristics. Again, however, the simplicity of the action does not necessarily coincide with the simplicity of the linguistic encoding. E.g. “clean X by rubbing it with Y ”describes one action as rubbing X is what cleans it (see [Di Eugenio(1992)] for discussion). We can consider that in a sentence fragment like this, the verb clean actually determines the condition of completion. What would be meant then is “rub X until clean ”which is more explicit.

The structure of the text should help detect the structure of procedure. One would expect to find the same level of granularity in the description of preparatory actions and in that of the those that follow the main action. A cursory inspection of instructional texts suggest, however, that this is not the case.

Again we need two types of mark-up, one based on the understanding of the text and one isolating the linguistic cues.

The mark-up needed to determine which additional information should be given upon request is designed to build a representation of the information the user has already gotten. This will be used to extract the appropriate information out of a database of facts about the devices, products and procedures under discussion. This database will need to be built up using information extracted out of the document itself. In a further elaboration of this idea we will study the ILEX project which has a similar scheme to deliver contexted information but does not start from the analysis of existing text.

These four mark-up schemes should help us in constructing a model for the oral presentation of (parts of) the annotated text. In the next sections we describe each of the schema’s in more detail.

Chapter 4

The annotation schemes

4.1 The conceptual annotation schemes

4.1.1 The subgenre mark-up

As said above, [Lux(1998)] propose a subdivision in different genres of different parts of technical manuals. Her study is based on Aerospatiale maintenance manuals. Generalising after inspection of some Xerox material we come the following tentative list of subgenres. It is not clear at this point which ones are obligatory and which ones are optional:

Function Description: this is a general overview, in general in the form of running text, of the way the device functions, what its functionality is, what the reason for the repair procedure is and how it needs to be approached. Encoding: **FDG**

Parts list: self evident. Encoding **PLG**

Maintenance Schedule: self evident. Encoding **MSG**

Tasks: here the maintenance or use instructions themselves are given. The structure can be recursive and this recursivity should be indicated. Encoding: **TSG**

Warnings: two types need to be distinguished and are sometimes labeled differently:

- About device damage: if you do that, the machine will break
Encoding: **WDG**
- About damage to user: if you do that, you will hurt yourself.
Encoding: **WUG**

Notes: asides that are not warnings. Encoding: **NTG**

These marks will be put around the relevant parts of text in the style of SGML marks, e.g. <NTG.../NTG>.

4.1.2 Discourse Structure mark-up

The mark-up is meant to give the discourse structure of the expository parts of the text. This markup takes subsections as the entities to be annotated: the subgenre mark-up gives the higher level structure whereas the discourse structure mark-up gives the details down to the clausal level.

Expansion 1 Change Paper Rolls

Structure 1+1+1+1 when paper roll finished,

Expansion 1+1+1 'End of Paper' alarm generated
print engine stop.

Expansion 1+1 A new roll should be mounted.

Sequence 1+2 change rolls

Structure 1+2+1 before the mounted roll is finished

Expansion 1+2+1+1 (change of paper grade).

Sequence 1+2+2+1 premature roll change results in paper loss

Sequence 1+2+2 group the printing of jobs on their paper needs.

Sequence 2 Roll checklist before printing

Expansion 2+1 Check roll for splices

Expansion 2+1+1 Check whether the roll is guaranteed splice free. .

Expansion 2+1+1+1 If so, a label is attached to the paper roll
a guarantee is indicated.

Expansion 2+1+1+2+1 If not, a splice is visible

Sequence 2+1+1+2 Proceed with step 2 to print on such a roll

Expansion 2+1+1+2+2 Stop printing

Structure 2+1+1+2+2+1 splice lies within a few mm of...

Sequence 2+1+1+2+3 Cut the paper at the splice table.

Sequence 2+1+1+2+4 Unwind the roll until ...

Sequence 2+1+1+2+5 Make a new splice ...

Sequence 2+1+1+2+6 Proceed printing.

Sequence 2+2 Check for tight roll windings

.....

Sequence 2+3 Check for dust on roll sides and surface

Interruption 2+3+a NOTE: Do not put the brake onto the roll before ...

Expansion 2+3+1 Remove dust from the roll sides using a special cleaning tool.

Sequence 2+3+2 Remove dust from the roll surface by removing the first winding...

.....

4.1.3 The activity mark-up

The activity mark-up itself will comprise:

Procedural unit: a piece of text that is interpreted as describing a complete procedure. Procedures can be composed of other procedures. They are encoded as **PUA**

Core action: **CA**

Preceding the core action, we find:

- *Enablers*: doing one action brings about the conditions necessary for the second (precondition); this is encoded as **EA**;
- *Condition statements*: if ...; they are encoded as **IA**;

Termination condition for core action:

- *Statement of purpose*, encoded as **PA**
- *Other*, encoded as **TA**

Concurrent actions. They can be core, preparatory or terminating. We distinguish two different cases:

- *Generation*: the two actions are simultaneous, one is not part of the other, one of them together with the set of conditions that holds implies the other, This is encoded as **GA**
- *Non generation*, encoded as **NA**

Sequences of actions. They can be core, preparatory or terminating, encoded as **SA**

For each atomic action we collect further information about instruments and aspect:

- *Instrument used*, encoded as **IN**
- Aspectual information [Moens and Steedman(1987)]:

- Event:
 - +/- *consequence* (+/- telic)
 - +/- *extended*

This gives rise to the following codes

EV-CONS-EXT EV+CONS-EXT EV-CONS+EXT EV+CONS+EXT

- *State*, encoded as **STATE**

Note that there is a difference between the aspectual information encoded in the action schema and that encoded in the lexico-syntactic schema: the action schema takes coercion into account, whereas the linguistic one encoded only the straight linguistic information. This distinction might be difficult to maintain and we might drop it later.

These markers will be put around the text in the style of SGML marks. Some can be embedded in each other and the same text fragment can have more than one code.

Example:

```
<PA>Cleaning the Coronas of the Paper Supply
* <EA><SA><1>1. <EV+CONS+EXT>Open the top cover of the paper supply.
    </EV+CONS+EXT></1>
*     <2>2. <EV+CONS+EXT>Remove the right cover of the paper supply.
    </EV+CONS+EXT></2>
*     <3>3. <EV+CONS-EXT>Locate the corona assembly.</EV+CONS-EXT></3>
* <WDG>NOTE: When handling the corona assemblies, make sure never to
    touch the corona wires.</WDG>
*     <4>4. <EV+CONS+EXT>Sliding the corona carefully out of the
    guide bars.</EV+CONS+EXT></4></SA></EA>
* <CA><SA><1>5. <GA><EV+CONS+EXT>Rub the housing of the corona with a
    <IN>cotton tip moistened with soap and water</IN>
    until <PA>it is clean.</EV+CONS+EXT></PA>.<GA></1>
*     <2>6. <EV-CONS+EXT> Rub the housing of the corona <IN>with a dry
    cotton tip<?IN>.</EV-CONS+EXT></2>
*     <3>7. <GA><EV+CONS+EXT>Rub the corona wires with a <IN>cotton tip
    moistened with soap and water</IN> until
    <PA>it is clean</PA>. /EV+CONS+EXT></GA></3>
*     <4>8. <EV-CONS+EXT> Rub the corona wires with a <IN>dry cotton
    tip</IN>.</EV-CONS+EXT></4></SA></CA>
* <TA><SA><1>9. <EV+CONS+EXT>Insert the corona assembly back into the guides.
    </EV+CONS+EXT></1>
*     <3>10. <EV+CONS+EXT>Replace the covers.</EV+CONS+EXT></3></SA></TA>
</PA>
```

These are the types that we anticipate to find in the procedures part. In the function description and the notes we expect to find other types, e.g. reasons, obstacles, criteria, etc. We do not intend to mark-up these sections for actions at this point.

4.2 The lexico-syntactic mark-up

The different elements that we have isolated can be indicated in different ways in language, for instance temporal relations can be expressed by complementisers or prepositions (before, after) or the order of the clauses or adverbs. The causal relations by complementisers or prepositions (to, by) or order. The modalities (necessity, possibility) are indicated by modal verbs, negation and complementisers (not, if, may, etc.)

It is well known that these linguistic elements give only an approximation to the discourse relations we are after: for instance the mood of the clause gives us direct speech acts but not indirect ones: “Could you open the door?” is in general used as a request and not as a question about the abilities of the addressee.

Given that there is no one-to-one correspondence between function and form, we code for both separately. In what follows we will make the simplifying assumption that only two types of noun phrases have to be added to the clausal inventory to get acceptable results:-ing forms and expressions of means, i.e. NP after with, by means of, etc. This assumption will be evaluated later in the project and, if necessary, changed.

The lexico-syntactic mark up, then, includes the following. I have grouped the various expressions according to the discourse function that they are the most often connected to but this grouping is only for expository reason. As said above: if the connection were one-to-one we would need only one type of mark-up.

Temporal indicators: before, after, when, then, while,...

They will be represented by the words themselves in upper case followed by the type of complement they take. E.g. **BEFORE+THAT**

Cause/purpose indicators: because, to, by, for, so,...

They will be represented by the words themselves followed by the type of complement they take, e.g. **TO+INF**

Instrument: with, use X to ..., by means of, ... Encoded as **WITH+NP**, **USE-TO+INF**, etc.

Speech acts indicators

- Imperative: encoded as **IMP**
- Indicative + tense: encoded as **IND-PRES**, **IND-PAST**
- Other: encoded as **OTHER-SAI**

Modalities and Moods

- Passive without or without agent: encoded as **PASS-AGENT** and **PASS+AGENT** respectively
- Conditional: encoded as either **IF-COND** or **INV-COND** (were he here, he would ...)

- Negation: not, never, no,... Encoded by the lexical item in **UPPERCASE**
- Deontic/epistemic modalities: must, has to, should, it is necessary, don't, may, could, can,...
Encoded by the lexical items in **UPPER-CASE**

Anaphoric relations

- Pronouns with their grammatical function.
Encoded as citation form of the pronoun in upper-case followed by SUB, OBJ, etc.
Fore instance **HE+SUB**
- Definite NP's if they have an antecedent, with their function.
Encoded as **DEFNP+FUNCTION**
- Antecedent with function for pronouns and NP's
Encoded as **DEFNP,INDEFNP,HE ... +FUNCTION**
- The coreference links themselves will be encoded with indices.

Coordination: and, or, neither...nor,... Encoded as the lexical item in **UPPER-CASE**

Deletions

- Missing object: encoded as **OBJ-DEL**
- Missing article: encoded as **ART-DEL**

That/if/whether-complement with governing verb: encode with the complementizer preceded by the verb. E.g. **SAY+THAT**.

Free adjuncts

- -ing forms used as FA, encoded as **+ING-FA**
- others, encoded as **-ING-FA**

Aspect

- State: encoded as **STATE**
- Event: encoded as **EVENT**
+/- consequent
+/- extended
So we get the following possibilities: **EV-CONS-EXTL EV+CONS-EXTL EV-CONS+EXTL EV+CONS+EXTL**

Example:

```
<IMP><EV-CONS+EXT>Rub
  <DEFNP+OBJ 1>the housing of
    <DEFNP+OFOBJ 2>the corona </DEFNP+OFOBJ></DEFNP+OBJ>
```

```

    <WITH+NP> with a cotton tip moistened
        <WITH+NP> with soap and water </WITH+NP></WITH+NP>
</EV-CONS+EXT>
    <UNTIL> until <1>it</1>
        <STATE> is clean</STATE></UNTIL></IMP>.

```

4.3 Annotating instructional text with information states

4.3.1 Dialogue vs. text

The basic idea of annotating instructional text with information states is to see text as a “degenerate case” of dialogue. This means that text generation is a special case of utterance generation. We annotate the text with the assumed information state of a dialogue system engaged in (monologically) generating the text. Interactive text generation, useful in e.g. hypertext navigation, can be seen as a middle ground between dialogue and monological text generation.

4.3.2 Information states in text generation

As said above, the structuring given in 4.1.2 can also be achieved by adding the notion of plan to the Gothenburg mark up scheme. On the Cooper & Larsson theory [Cooper *et al.*(1999)], the information state of a dialogue participant contains two main fields, the private field and the shared field. The shared field contains information which has been explicitly established during the conversation and is assumed to be shared by both participants. The process of entering information into the shared field is known as “grounding” and usually involves explicit or implicit acknowledgement. In monological text generation there is, of course, no feedback and thus no grounding process. However, it seems reasonable that a writer would assume that whatever text the reader has previously read has been understood. In other words, the writer may be said to assume that the text is immediately grounded as soon as it is read. The distinction between private and shared information is thus still potentially useful in text generation. Since whatever is in the shared field is assumed to be known by the reader, the system can e.g. avoid unnecessarily repeating itself by checking the shared field before generating a certain sentence.

Adding the notion of a plan to the Göteborg theory of information states used in [Cooper *et al.*(1999)], we represent information states of a dialogue participant as records of the following type:

$$(1) \left[\begin{array}{l} \text{PRIVATE} \\ \text{SHARED} \end{array} : \left[\begin{array}{ll} \text{PLAN} & : \text{LIST}(\text{ACTION}) \\ \text{AGENDA} & : \text{STACK}(\text{ACTION}) \\ \text{BEL} & : \text{SET}(\text{PROP}) \\ \text{QUD} & : \text{STACK}(\text{QUESTION}) \end{array} \right] \right]$$

The PLAN field contains a dialogue plan (here: a text generation plan), i.e. is a list of dialogue actions that the agent wishes to carry out. The AGENDA field, on the other hand, contains the short term goals that the agent has, i.e. what the agent is going to do next.

Transitions between information states are defined in terms of a restricted set of operations. The operations we have used in this coding are:

Stack: push, pop
Set: add an element

In order to make clear what changes are taking place in the transition from one information state to another we represent the operations resulting from the preceding contribution in the annotation before the resulting information state.

When a participant in dialogue raises a question and that question is acknowledged, it is entered onto the stack of Questions Under Discussion (QUD). The other participant typically puts on his agenda the action to respond to that question. Texts are typically structured according to topics and issues, and one way of viewing topics and issues is as questions under discussion. Thus, in text generation we assume that certain pieces of text (e.g. section headers) raise questions; whenever that text is generated, the corresponding question is automatically raised. When the a dialogue system executes an *raise* move, it will expect the user to respond to the question. In text generation, however, the system will have to answer its own questions. As in dialogue, the system will put on the agenda to try to respond to the raised question, which prompts it to search for a plan for responding. Plans for answering questions contain dialogue moves such as *inform* and *raise* (for raising questions). The plan may generate text which in turn raises questions, which will trigger further plans. In this way, plans for answering questions give rise to a hierarchical structure of a text. The topmost structure of a text may be provided by a table of contents, where each topic and subtopic raises a question.

The shared field also contains a set of shared beliefs, which are propositions that the text generation system assumes as background knowledge which can be used to determine the text output. This can be useful e.g. for the generation of custom-made manuals appropriate for different reading styles. By adopting different strategies for setting the contents of the shared field at the beginning of each section, a single database can be used to generate manuals for different reading styles. For example, to produce a manual for a user who wants to go immediately to a specific subsection, the strategy would be to not assume knowledge of previous sections. On the other hand, in the case where the user reads the whole book from cover to cover one would assume knowledge of previous sections.

4.3.3 Text structure and information state annotation

As an illustration we have annotated an excerpt from a Xerox maintenance manual with information state updates, supplemented with move-type.

- (2) 7. Managing Paper
7 When to Change Paper Rolls?
When a paper roll is finished, an 'End of Paper' alarm will be generated and the print engine will stop automatically. A new roll should be mounted. In some cases you might want to change rolls before the mounted roll is finished (change of paper grade).
As a premature roll change results in some extra paper loss and non-productive time, it is advisable to group the printing of jobs as much as possible on their paper needs.
7 Roll checklist before printing

From the annotation a hierarchical structure of the text emerges, as can be seen in (3)¹. Text fragment [1] raises a question, and the plan for answering this question is to raise two further questions, corresponding to [2] and [7]. To answer the question corresponding to [2], fragments [3] and [4] are produced. However, [4] raises a further question, which triggers a plan containing [5] and [6].

- (3) [1] Managing Paper
[2] When to Change Paper Rolls?
[3] When a paper roll is finished, an 'End of Paper' alarm will be generated and the print engine will stop automatically. A new roll should be mounted.
[4] In some cases you might want to change rolls before the mounted roll is finished
[5] (change of paper grade).
[6] As a premature roll change results in some extra paper loss and non-productive time, it is advisable to group the printing of jobs as much as possible on their paper needs.
[7] Roll checklist before printing

¹The section numbers in the original text has been omitted here for reasons of clarity.

4.3.4 The annotated text

[1] Managing Paper

DATABASE: THIS SECTION TITLE RAISES A QUESTION

pushRec(SHARED.QUD, 'How is paper managed?')

$$\left[\begin{array}{l} \text{PRIVATE} \\ \text{SHARED} \end{array} = \left[\begin{array}{l} \text{AGENDA} = \langle \rangle \\ \text{PLAN} = \langle \rangle \\ \text{QUD} = \langle \text{'How is paper managed?'} \rangle \\ \text{BEL} = \{ \} \end{array} \right] \right]$$

MOVE: FIND PLAN (TO RESPOND TO QUESTION)

pushRec(PRIVATE.PLAN, **raise**('What should I do before printing?'))

pushRec(PRIVATE.PLAN, **raise**('When should paper rolls be changed?'))

$$\left[\begin{array}{l} \text{PRIVATE} \\ \text{SHARED} \end{array} = \left[\begin{array}{l} \text{AGENDA} = \langle \rangle \\ \text{PLAN} = \langle \text{raise('When should paper rolls be changed?')} \\ \text{raise('What should I do before printing')} \rangle \\ \text{QUD} = \langle \text{'How is paper managed?'} \rangle \\ \text{BEL} = \{ \} \end{array} \right] \right]$$

MOVE: REFILL AGENDA

pushRec(PRIVATE.AGENDA, **raise**('When should paper rolls be changed?'))

popRec(PRIVATE.PLAN)

$$\left[\begin{array}{l} \text{PRIVATE} \\ \text{SHARED} \end{array} = \left[\begin{array}{l} \text{AGENDA} = \langle \text{raise('When should paper rolls be changed?')} \rangle \\ \text{PLAN} = \langle \text{raise('What should I do before printing')} \rangle \\ \text{QUD} = \langle \text{'How is paper managed?'} \rangle \\ \text{BEL} = \{ \} \end{array} \right] \right]$$

[2] When to Change Paper Rolls?

MOVE: RAISE

pushRec(SHARED.QUD, 'When should paper rolls be changed?')

popRec(PRIVATE.AGENDA)

pushRec(PRIVATE.AGENDA, **respond**('When should paper rolls be changed?'))

$$\left[\begin{array}{l} \text{PRIVATE} \\ \text{SHARED} \end{array} = \left[\begin{array}{l} \text{AGENDA} = \langle \text{respond('When should paper rolls be changed?')} \rangle \\ \text{PLAN} = \langle \text{raise('What should I do before printing')} \rangle \\ \text{QUD} = \langle \text{'When should paper rolls be changed?'} \rangle \\ \text{BEL} = \{ \} \end{array} \right] \right]$$

MOVE: FIND PLAN IN DATABASE (TO RESPOND TO QUESTION)

popRec(PRIVATE.AGENDA)

pushRec(PRIVATE.PLAN, **inform**('In some cases you might want to change rolls before...'))

pushRec(PRIVATE.PLAN, **inform**('When a paper roll is finished, an 'End of Paper' alarm will be...'))

$$\left[\begin{array}{l} \text{PRIVATE} \\ \text{SHARED} \end{array} = \left[\begin{array}{l} \text{AGENDA} = \langle \text{inform('When a paper roll is finished, an 'End of Paper' alarm will be...')} \rangle \\ \text{PLAN} = \langle \text{inform('In some cases you might want to change rolls before...')} \\ \text{raise('What should I do before printing')} \rangle \\ \text{QUD} = \langle \text{'When should paper rolls be changed?'} \rangle \\ \text{BEL} = \{ \} \end{array} \right] \right]$$

MOVE: REFILL AGENDA

pushRec(PRIVATE.AGENDA, **inform**('When a paper roll is finished, an 'End of Paper' alarm will be...'))

popRec(PRIVATE.PLAN)

$$\left[\begin{array}{l} \text{PRIVATE} \\ \text{SHARED} \end{array} = \left[\begin{array}{l} \text{AGENDA} = \langle \text{inform('When a paper roll is finished, an 'End of Paper' alarm will be...')} \rangle \\ \text{PLAN} = \langle \text{inform('In some cases you might want to change rolls before...')} \\ \text{raise('What should I do before printing')} \rangle \\ \text{QUD} = \langle \text{'When should paper rolls be changed?'} \rangle \\ \text{BEL} = \{ \} \end{array} \right] \right]$$

[3] When a paper roll is finished, an 'End of Paper' alarm will be generated and the print engine will stop automatically. A new roll should be mounted.

MOVE: INFORM

addRec(SHARED.BEL, 'When a paper roll is finished, an 'End of Paper' alarm will be...')

popRec(PRIVATE.AGENDA)

$$\left[\begin{array}{l} \text{PRIVATE} \\ \text{SHARED} \end{array} = \left[\begin{array}{l} \text{AGENDA} = \langle \text{inform('In some cases you might want to change rolls before...')} \\ \text{raise('What should I do before printing')} \rangle \\ \text{QUD} = \langle \text{'When should paper rolls be changed?'} \rangle \\ \text{BEL} = \{ \text{'When a paper roll is finished, an 'End of Paper' alarm will be...'} \} \end{array} \right] \right]$$

MOVE: REFILL AGENDA

pushRec(PRIVATE.AGENDA, **inform**('In some cases you might want to change rolls before...'))
 popRec(PRIVATE.PLAN)

$$\left[\begin{array}{l} \text{PRIVATE} \\ \text{SHARED} \end{array} = \left[\begin{array}{l} \text{AGENDA} = \left\langle \mathbf{inform}('In\ some\ cases\ you\ might\ want\ to\ change\ rolls\ before...') \right\rangle \\ \text{PLAN} = \left\langle \mathbf{raise}('What\ should\ I\ do\ before\ printing') \right\rangle \\ \text{QUD} = \left\langle \begin{array}{l} 'When\ should\ paper\ rolls\ be\ changed?' \\ 'How\ is\ paper\ managed?' \end{array} \right\rangle \\ \text{BEL} = \left\{ 'When\ a\ paper\ roll\ is\ finished,\ an\ 'End\ of\ Paper'\ alarm\ will\ be...' \right\} \end{array} \right] \right]$$

[4] In some cases you might want to change rolls before the mounted roll is finished

MOVE: INFORM, RAISE

addRec(SHARED.BEL, 'In some cases you might want to change rolls before...')
 popRec(PRIVATE.AGENDA)

$$\left[\begin{array}{l} \text{PRIVATE} \\ \text{SHARED} \end{array} = \left[\begin{array}{l} \text{AGENDA} = \langle \rangle \\ \text{PLAN} = \left\langle \mathbf{raise}('What\ should\ I\ do\ before\ printing') \right\rangle \\ \text{QUD} = \left\langle \begin{array}{l} 'When\ should\ paper\ rolls\ be\ changed?' \\ 'How\ is\ paper\ managed?' \end{array} \right\rangle \\ \text{BEL} = \left\{ \begin{array}{l} 'In\ some\ cases\ you\ might\ want\ to\ change\ rolls\ before...' \\ 'When\ a\ paper\ roll\ is\ finished,\ an\ 'End\ of\ Paper'\ alarm\ will\ be...' \end{array} \right\} \end{array} \right] \right]$$

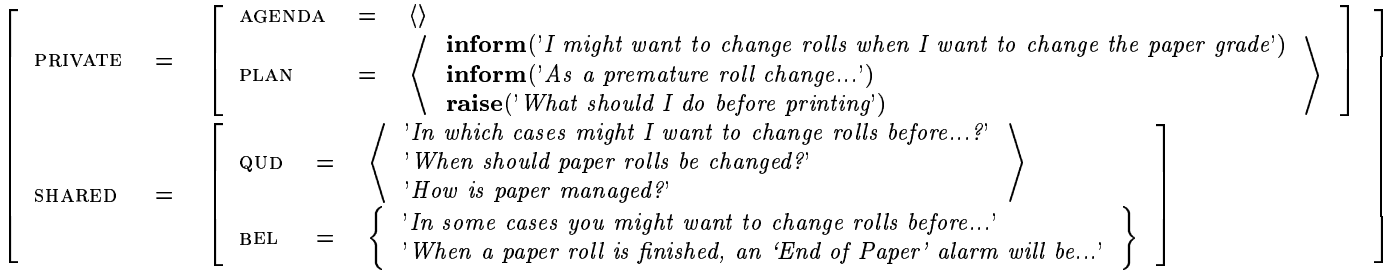
DATABASE: THE ABOVE MOVE ALSO RAISES A QUESTION:

pushRec(SHARED.QUD, 'In which cases might I want to change rolls before...?')
 pushRec(PRIVATE.AGENDA, **respond**('In which cases might I want to change rolls before...?'))

$$\left[\begin{array}{l} \text{PRIVATE} \\ \text{SHARED} \end{array} = \left[\begin{array}{l} \text{AGENDA} = \left\langle \mathbf{respond}('In\ which\ cases\ might\ I\ want\ to\ change\ rolls\ before...?') \right\rangle \\ \text{PLAN} = \left\langle \mathbf{raise}('What\ should\ I\ do\ before\ printing') \right\rangle \\ \text{QUD} = \left\langle \begin{array}{l} 'In\ which\ cases\ might\ I\ want\ to\ change\ rolls\ before...?' \\ 'When\ should\ paper\ rolls\ be\ changed?' \\ 'How\ is\ paper\ managed?' \end{array} \right\rangle \\ \text{BEL} = \left\{ \begin{array}{l} 'In\ some\ cases\ you\ might\ want\ to\ change\ rolls\ before...' \\ 'When\ a\ paper\ roll\ is\ finished,\ an\ 'End\ of\ Paper'\ alarm\ will\ be...' \end{array} \right\} \end{array} \right] \right]$$

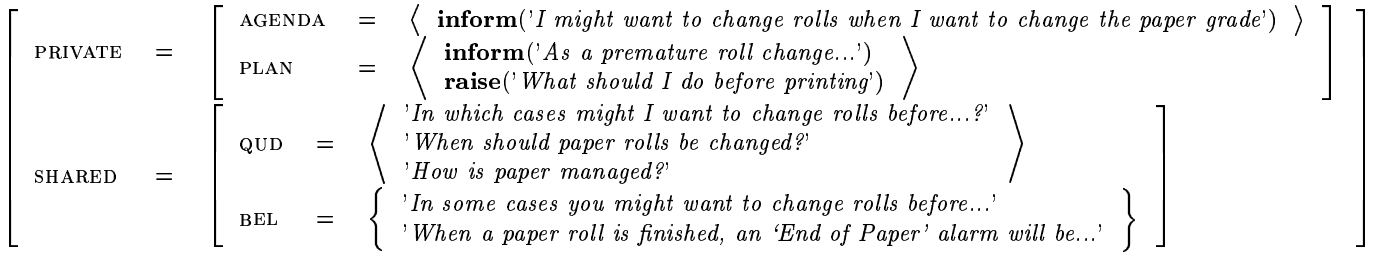
MOVE: FIND PLAN IN DATABASE (TO RESPOND TO QUESTION)

pushRec(PRIVATE.PLAN, **inform**('As a premature roll change...'))
 pushRec(PRIVATE.PLAN, **inform**('I might want to change rolls when I want to change the paper grade'))
 popRec(PRIVATE.AGENDA)



MOVE: REFILL AGENDA

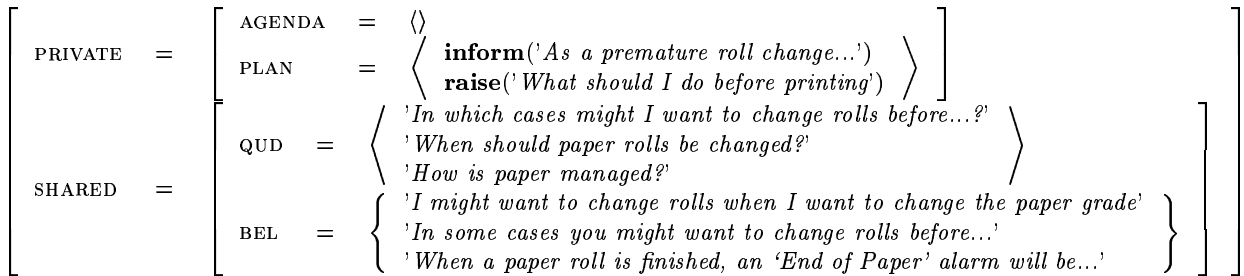
pushRec(PRIVATE.AGENDA, **inform**('I might want to change rolls when I want to change the paper grade'))
popRec(PRIVATE.PLAN)



[5] (change of paper grade).

MOVE: INFORM

addRec(SHARED.BEL, 'I might want to change rolls when I want to change the paper grade')
popRec(PRIVATE.AGENDA)



MOVE: REFILL AGENDA

pushRec(PRIVATE.AGENDA, **inform**('As a premature roll change...'))
popRec(PRIVATE.PLAN)

$$\left[\begin{array}{l} \text{PRIVATE} \\ \text{SHARED} \end{array} = \left[\begin{array}{l} \text{AGENDA} = \langle \text{inform}('As a premature roll change...') \rangle \\ \text{PLAN} = \langle \text{raise}('What should I do before printing') \rangle \\ \text{QUD} = \langle \begin{array}{l} 'In which cases might I want to change rolls before...?' \\ 'When should paper rolls be changed?' \\ 'How is paper managed?' \end{array} \rangle \\ \text{BEL} = \left\{ \begin{array}{l} 'I might want to change rolls when I want to change the paper grade' \\ 'In some cases you might want to change rolls before...' \\ 'When a paper roll is finished, an 'End of Paper' alarm will be...' \end{array} \right\} \end{array} \right]$$

[6] As a premature roll change results in some extra paper loss and non-productive time, it is advisable to group the printing of jobs as much as possible on their paper needs.

MOVE: INFORM

popRec(PRIVATE.AGENDA)

addRec(SHARED.BEL, 'As a premature roll change...')

$$\left[\begin{array}{l} \text{PRIVATE} \\ \text{SHARED} \end{array} = \left[\begin{array}{l} \text{AGENDA} = \langle \rangle \\ \text{PLAN} = \langle \text{raise}('What should I do before printing') \rangle \\ \text{QUD} = \langle \begin{array}{l} 'In which cases might I want to change rolls before...?' \\ 'When should paper rolls be changed?' \\ 'How is paper managed?' \end{array} \rangle \\ \text{BEL} = \left\{ \begin{array}{l} 'As a premature roll change...' \\ 'I might want to change rolls when I want to change the paper grade' \\ 'In some cases you might want to change rolls before...' \\ 'When a paper roll is finished, an 'End of Paper' alarm will be...' \end{array} \right\} \end{array} \right]$$

TOPMOST QUESTION ANSWERED; REMOVE FROM QUD

popRec(SHARED.QUD)

$$\left[\begin{array}{l} \text{PRIVATE} \\ \text{SHARED} \end{array} = \left[\begin{array}{l} \text{AGENDA} = \langle \rangle \\ \text{PLAN} = \langle \text{raise}('What should I do before printing') \rangle \\ \text{QUD} = \langle \begin{array}{l} 'When should paper rolls be changed?' \\ 'How is paper managed?' \end{array} \rangle \\ \text{BEL} = \left\{ \begin{array}{l} 'As a premature roll change...' \\ 'I might want to change rolls when I want to change the paper grade' \\ 'In some cases you might want to change rolls before...' \\ 'When a paper roll is finished, an 'End of Paper' alarm will be...' \end{array} \right\} \end{array} \right]$$

TOPMOST QUESTION ANSWERED; REMOVE FROM QUD

popRec(SHARED.QUD)

$$\left[\begin{array}{l} \text{PRIVATE} \\ \text{SHARED} \end{array} = \left[\begin{array}{l} \text{AGENDA} = \langle \rangle \\ \text{PLAN} = \langle \text{raise('What should I do before printing')} \rangle \\ \text{QUD} = \langle \text{'How is paper managed?'} \rangle \\ \text{BEL} = \left\{ \begin{array}{l} \text{'As a premature roll change...'} \\ \text{'I might want to change rolls when I want to change the paper grade'} \\ \text{'In some cases you might want to change rolls before...'} \\ \text{'When a paper roll is finished, an 'End of Paper' alarm will be...'} \end{array} \right\} \end{array} \right] \right]$$

[7] Roll checklist before printing

MOVE: RAISE

pushRec(SHARED.QUD, 'What should I do before printing')

$$\left[\begin{array}{l} \text{PRIVATE} \\ \text{SHARED} \end{array} = \left[\begin{array}{l} \text{AGENDA} = \langle \rangle \\ \text{PLAN} = \langle \text{raise('What should I do before printing')} \text{ n} \rangle \\ \text{QUD} = \langle \begin{array}{l} \text{'What should I do before printing'} \\ \text{'How is paper managed?'} \end{array} \rangle \\ \text{BEL} = \left\{ \begin{array}{l} \text{'As a premature roll change...'} \\ \text{'I might want to change rolls when I want to change the paper grade'} \\ \text{'In some cases you might want to change rolls before...'} \\ \text{'When a paper roll is finished, an 'End of Paper' alarm will be...'} \end{array} \right\} \end{array} \right] \right]$$

Bibliography

- [Cooper *et al.*(1999)] Cooper, R., Larsson, S., Matheson, C., Poesio, M., and Traum, D. (1999). Coding instructional dialogue for information states. deliverable TRINDI deliverable D1.1, TRINDI.
- [Di Eugenio(1992)] Di Eugenio, B. (1992). Understanding natural language instructions: the case of purpose clauses. In *Proceedings 30th Annual Meeting of the Association for Computational Linguistics*, pages 120–127. Association for Computational Linguistics.
- [Knott and Dale(1993)] Knott, A. and Dale, R. (1993). Choosing a set of rhetorical relations for text generation. In *Proc of the fourth European Workshop on NL Generation*.
- [Lux(1998)] Lux, V. (1998). *Elaboration d'un Franais Rationalis Etendu Modulaire (FREM) pour les Manuels de Maintenance d'Aronefs*. Ph.D. thesis, Universit Paris.
- [Moens and Steedman(1987)] Moens, M. and Steedman, M. (1987). Temporal ontology in natural language. In *Proceedings of the 25th Annual Meeting of the ACL*.
- [Polanyi(1988)] Polanyi, L. (1988). A formal model of discourse structure. *Journal of Pragmatics*, **12**.