Content and Rhetorical Status Selection in Instructional Texts

Leila Kosseim
kosseim@iro.umontreal.ca

Guy Lapalme
lapalme@iro.umontreal.ca

Département d'informatique et de recherche opérationnelle
Université de Montréal
PB 6128, Succ. Centre Ville
Montréal, Québec, Canada
H3C 3J7

Abstract

This paper discusses an approach to planning the content of instructional texts. The research is based on a corpus study of 15 French procedural texts ranging from step-by-step device manuals to general artistic procedures. The approach taken starts from an AI task planner building a task representation, from which *semantic carriers* are selected. The most appropriate RST relations to communicate these carriers are then chosen according to heuristics developed during the corpus analysis.

1 Introduction

A standard problem in text generation is to determine what to include in the text and how to structure it. The goal of this research is to study how the content of instructional texts and their rhetorical structure can be selected automatically. The approach taken starts from a task representation developed by an AI planner, from which a set of *semantic carriers*, specifying the content of the text, is selected. Then the rhetorical relations that best communicate these semantic carriers are selected. The approach is based on a corpus analysis that determined:

- What semantic carriers are found in instructional texts, where they can be found in the task representation and when they are included in the text.

- What rhetorical relations are used to present the semantic carriers and when one is preferred over another.

If these points are not dealt with, an instructional text generator may choose to say everything available in the task representation, and may communicate it using always the same rhetorical strategy. For example, the task of using the one touch record (OTR) feature of a VCR can be represented as in figure 1[1]. From this task representation, the following unacceptable text may be produced:

```
To use the OTR feature, set the speed selector to "SP",
"SP" will light up; select channel 4; specify the
recording time; and press the TIMER button within 9
seconds, the TIMER indicator will light up.
To set the speed selector to "SP", press the SP/EP
button. The speed will change.
To set channel 4, press the channel button. The channel
will change.
To specify the recording time, press the OTR button 3
times.
To press the OTR button 3 times, press it once, PM
10:35; press it a second time, PM 11:05; press it a
third time, PM 11:35.
```

A more natural text would be[2]:

```
The OTR feature:
Set the tape speed selector to "SP".  "SP" will       (1)
light up.
Select channel 4, by pressing the channel button.     (2)
Press the OTR button 3 times to specify the           (3)
recording time.
When the button is pressed once, PM 10:35.            (4)
When it is pressed twice, PM 11:05.                   (5)
When it is pressed 3 times, PM 11:35.                 (6)
Press the TIMER button within 9 seconds.  The         (7)
TIMER indicator will light up.
```

Not all the knowledge of the task representation is introduced in the text. For example, many parent–child relations between operations are left unsaid. In addition, the same type of knowledge is not always presented by the same rhetorical relation. For example the parent-child relation of the operations in sentence (2) is expressed by an **enablement** relation rather than a **purpose** as in sentence (3).

The paper first reviews previous work in instructional text planning. The results of the corpus analysis leading

[1] In figure 1, operations are indicated on the top lines and post-conditions are preceded by an "⇒".

[2] The French version of this text was automatically produced by the prototype implementation, see section 4.

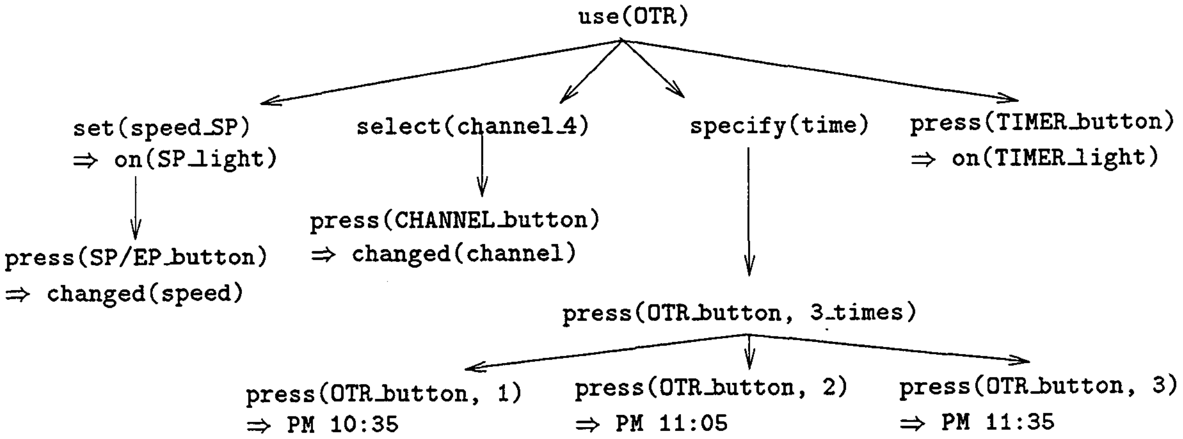

Figure 1: Task representation for using the OTR feature

to the development of the planning approach are then discussed. Finally, preliminary results of a prototype system are presented.

2 Planning Instructional Texts

The goal of an instructional text is to describe the actions to be performed to achieve a particular goal. For the reader to understand/perform the procedure correctly, the instructional text must communicate the plan of the procedure into the text. In AI, planning techniques have been developed to construct such plans automatically. It then becomes natural to consider an AI planner as a preliminary component to an instructional text generator. The output of an AI planner provides a fairly appropriate source for generating instructional texts, as only specific types of information are found in these texts, and most of them can be found or derived from a task representation.

Another important characteristic of instructional texts is that their rhetorical structure is rather stereotyped and use a small set of relations [12, 15]. In addition, these relations correspond very well to those defined in Mann and Thompson's RST [7]. This makes RST an attractive tool for studying this genre.

From these two remarks, the planning of instructional texts is often seen as a two-stage process [8, 1]: a task planning stage, where the plan of the procedure is developed, followed by a text planning stage, where the content of the text is selected from the task representation[3], and the rhetorical status of this knowledge is selected.

When a task representation is used as the source of text planning, the resulting text is very much dependent on the representation's structure. However, a task representation is not universal; several factors can influence its development. The lexical capacity of the language being used may influence the conceptual representation of the task. Also, independently of the language being used, the same task can be represented in various ways (whether with more or less detail, or with a different structure). However, whether we use one representation or another as the basis for generation, the goal of our research is to generate a text similar to "natural" ones so that readers can interpret and possibly execute the procedure correctly. Even when human writers *w1* and *w2* write "natural" instructions for readers *r1* and *r2*, they may base their writing on different task representations and may choose to transmit different information. When reading the texts, *r1* and *r2* may build different task representations from one another and from *w1* and *w2*. But if both texts are adequate, the readers will interpret the prescribed task correctly. No single ideal task representation exists for a procedure and no single ideal text describing it exists. Our goal is not to construct and use the one task representation that allows the generation system to reproduce the source text word for word, but to produce "possibly natural" texts from a "possibly natural" task representation.

2.1 AI Planning

An AI planner, or task planner, attempts to find a set of operations to achieve some task, or goal [13]. It tries to transform the current state of the world, where the goal

[3] A task representation provides most knowledge found in instructional texts, but a model of the reader and a domain knowledge base are also required.

is not satisfied to a final state where the goal is true. The task planner takes as input a library of *operation schemas* and selects and orders a subset of these by constructing some *task representation*. Typically, AI planners are hierarchical and develop non-linear plans. That is, they define the plan by successive refinement, decomposing the current plan to a lower, more detailed level of abstraction until *primitive operations* (which do not require further refinement) are reached. The resulting structure is a *hierarchy of plans*. Non-linearity involves defining a partial order on the operations.

In the context of instruction generation, operation schemas can be built, as in traditional AI planning, out of: (1) the operation name (eg. `use(OTR)`), (2) its body: how the operation can be sub-divided into simpler operations (eg. `set(speed_SP)` $\wedge$ `select(channel_4)` $\wedge \ldots$), (3) its preconditions: states that need to be true in order to apply the operation (eg. `in_vcr(cassette)`); if the task planner cannot solve a precondition, the condition is passed on to the text planner to be included in the text and solved by the agent, (4) and its postconditions: states that become true or false after the operation is executed. Postconditions can be divided into: postconditions of success (if the operation is performed correctly, eg. `on(timer_light)`) and postconditions of failure (if the operation is not performed correctly, eg. `blink(error_light)`).

Note that we consider operations to be events whose actor is the agent performing the procedure, and could occur now or in the near future. Events whose actor is the device being manipulated, some undefined actor, or which have occurred previously are considered states and may be represented as postconditions or preconditions. For example, the actor of `turn_on(timer_light)` is not the agent of the procedure, but the device being manipulated. This is thus not represented as an operation in the schemas but as a postcondition state of an agent operation.

2.2 Previous Work in Planning Instructions

Previous work on planning the contents of instructional texts include Dale's EPICURE [1] and Mellish and Evans's system [9], which both start from the output of an AI task planner to select the text's content. However the first system seems only applicable to a particular domain; while the other seems to lack linguistic motivation.

In [1], all primitive operations of the task representation are included in the text as a sequence of actions. Furthermore, only these primitive operations are included in the text. In EPICURE's domain (cooking recipes), 90 to 100% of the content are action sequences [15]. Since the other knowledge available in the task representation (non-primitive operations, parent-child relations, ...) are usually not presented as action sequences, the strategy seems appropriate in this domain. However, it cannot be applied directly in most other domains where the proportion of action sequences is lower. On the other hand, to explain a plan, [9] includes in the text all operations and hierarchical relations available in the task representation. No result, condition, negative imperative [3], or other information found in naturally occurring texts are included. Moreover, the rhetorical strategies used do not seem linguistically motivated.

Our efforts are aimed at developing heuristics guiding the selection of content and rhetorical status to produce "natural" instructions. We have emphasized two types of tasks. According to the classification of [5], these are: operator tasks, ie. procedures on a system or device to accomplish a goal external to that system/device (eg. mowing the lawn); and maintenance/repair tasks, that is, specific operations on a system/device (eg. repairing a tape recorder).

3 The Corpus Analysis

In order to generate "natural" texts, we first analyzed a corpus of 15 French instructional texts ($\approx$ 13,300 words) from different writers, domains, and text types.

Our view of instructional texts is rather larger than in other NLP research. For example, in [15, 12, 2] texts are restricted to execution-oriented instructions, where the reader is assumed to be the agent of the procedure. However, many texts that indicate how to perform a task are explanation-oriented and thus do not assume that the reader will immediately or even ever execute the procedure. Our corpus thus ranges from step-by-step procedures (also called procedural directives [4] and linear explanations [5]), like device manuals to explanations-oriented texts like arts and crafts books.

We performed an RST analysis only at the bottom level of the textual structure, that is at the inter- and intra-clausal levels. As reported in Vander Linden [15] and Rösner and Stede [12], instructional texts use only a subset of RST relations. The most common are **temporal sequence**, **circumstance/condition** (c-condition[4]), **result** (volitional and non volitional), **purpose**, **enablement** and other non-procedural relations (attributes of objects, motivation ...).

[4] A c-condition combines RST's relations of circumstance and condition. It is what [15] and [12] call "precondition", but we prefer to use this term in its AI planning definition.

3.1 A Semantic Level

With the idea of using an AI task planner, and later rhetorical relations, we first set out to see how the knowledge available from a task representation (call it *task knowledge*) could directly determine rhetorical relations. In most cases, this is successful. For example, operation nodes in the task representation are mostly presented by action sequences, parent operation nodes by purpose relations, postconditions by results ...However in many cases, the same type of task knowledge is communicated through different relations.

For example, parent-child relations can be presented in the text by a purpose related to an action. In this case the relation is seen bottom–up and explains why the child operation should be performed, as in:

(1) Revissez l'écrou-capuchon sur la lyre *pour ne pas le perdre.*
(Screw the screw-cap on the lamp shade holder *so that you do not lose it.*[5])

A parent-child relation can also be presented by an enablement related to an action. In that case, the relation is seen top–down and explains how the parent operation should be performed, as in:

(2) Régler la ceinture *en la tirant par la languette.*
(Adjust the belt *by pulling it by the flap.*)

Another example involves preconditions which can be presented by a purpose relation or a c-condition, as in:

(3) a. *Pour vous aider*, poussez fermement le flanc du pneu avec votre pied.
(*To help you*, firmly press the side of the tire with your foot.)

b. *Si la victime est debout*, placez-vous derrière elle.
(*If the victim is standing*, place yourself behind him.)

As shown from these examples, the task knowledge does not uniquely determine the rhetorical relation used. The opposite is also true. In both examples (1) and (3a), a purpose relation is used, but in (1), it indicates a hierarchical relation and in (3a) it indicates a precondition on the "press" operation.

In order to map the task knowledge to the appropriate rhetorical structure, we have introduced an intermediate semantic level. This level classifies task knowledge into *semantic carriers* according to functional criteria (the mandatory/optional nature of operations, the execution time, the influence of an operation on the interpretation of the procedure ...). Semantic carriers help determine what task knowledge is introduced in the text and what rhetorical relation should be used.

For example, in (1), the parent-child relation carries a sense *causality* because it indicates that the "screwing" operation will cause the agent not to lose the screw-cap. The parent operation (not losing the screw-cap) does not influence how the child operation should be executed but rather justifies it. For this kind of semantic carrier, a purpose relation is most frequently used. In (2) however, the execution of the "pull" operation is influenced by the "adjust" goal; it carries a sense of *guidance* on how the execution should be performed. In this particular case, an enablement relation was selected. In (3a), the precondition indicates an *option* that the agent will probably select, this explains why a purpose relation is used. Finally, in (3b), the precondition indicates a *material condition*; this semantic carrier is mostly presented by a c-condition relation.

3.2 Some Results

Figure 2 shows the correspondence between the task knowledge, the semantic carriers they can bring about, and the rhetorical relations used to present them. These semantic carriers are by no means the only way to interpret the information communicated in instructional texts (see for example [10, 3]), and only account for procedural type information. They are based on our interpretation of our corpus.

The heuristics to introduce certain semantic carriers rely heavily on the notion of *basic-level operations* introduced by Rosch [11] and Pollack [10]. Basic-level operations are those operations that people seem to remember and are able to represent mentally most easily. In the texts, they turn out to be detailed enough to be descriptive, but general enough to be useful. In her work, [11] found considerable agreement among people on the kinds of units of events that are remembered. For example, when asked to recall events that occurred in the morning, subjects remembered operations like brushing their teeth, taking a shower, but no one mentioned smaller units like squeezing the toothpaste tube ...or larger units like "doing the morning chores". It was hypothesized then that people have a more accessible memory representation for basic-level operations, than for any other type of event. This hypothesis seems appropriate in instructional texts: basic-level operations are included in the text as the writers have a memory representation for them and promote the reader's recall, for they can easily build a memory representation of the procedure. Then, depending on the level of knowledge of the reader, more or less detailed operations are given. Basic-level operations are a rather subjective notion and depend heavily on factors like the communicative goal, the discourse domain ...

Only the most common semantic carriers (making up more than 80% of the texts) are discussed here. These

[5] All English translations are ours.

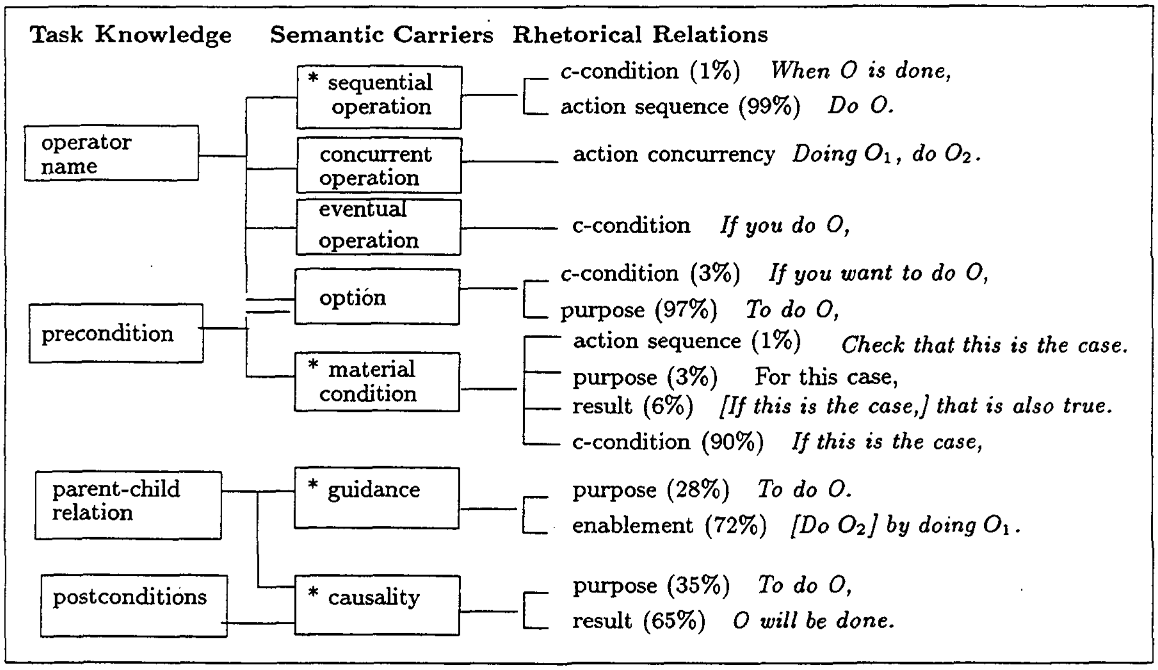
Task Knowledge
Semantic Carriers
Rhetorical Relations
operator name
precondition
parent-child relation
postconditions
* sequential operation
concurrent operation
eventual operation
option
* material condition
* guidance
* causality
c-condition (1%) When O is done,
action sequence (99%) Do O.
action concurrency Doing O_1, do O_2.
c-condition If you do O,
c-condition (3%) If you want to do O,
purpose (97%) To do O,
action sequence (1%) Check that this is the case.
purpose (3%) For this case,
result (6%) [If this is the case,] that is also true.
c-condition (90%) If this is the case,
purpose (28%) To do O.
enablement (72%) [Do O_2] by doing O_1.
purpose (35%) To do O,
result (65%) O will be done.

Figure 2: Correspondence between the task knowledge, the semantic carriers and the rhetorical relations

include sequential operations, material conditions, guidances and causalities[6].

Sequential operations indicate an operation whose execution is mandatory, immediate and not concurrent with another. This semantic carrier is the most frequent in all types of instructions. In our corpus, it counts 748 occurrences and make up 80% (in step-by-step instructions) to 30% (in explanations-oriented instructions) of the text.

Sequential operations are found as operation names in the task representation. According to our analysis, three types of operations should be included in the text:

• All basic-level operations;

• All children operations of basic-level operation, that have different postconditions from their siblings, and whose postconditions will be included in the text as a causality (see below).

• All children operations of basic-level operation that the reader does not know how to perform. In this case, a sub-procedure is introduced.

Sequential operations are sometimes presented by c-condition[7] (1% of the time), but almost always by a sequence of action clauses (99%).

o A c-condition is used if a result, a negative imperative [3] or an action sequence will follow the operation in the text, and the operation is durative or follows case 2 above (see sentences 4, 5 and 6 of the VCR text).

o Otherwise, the operation is presented by a temporal sequence of actions.

Material Conditions are preconditions on the state of the environment that the task planner is not able to verify.

• All such preconditions are included in the text to let the agent decide for himself whether the next line of operations should be performed.

Out of 158 material conditions of the corpus, only 1% are presented by an action sequence, 3% by a purpose, 6% by a result and 90% by a c-condition.

To determine the rhetorical status of material conditions we believe that:

o As [15] found, material conditions that specify ensurative actions (that the agent can make sure are true or do something so that they become true) are presented by action sequences. For example,

[6] For easy reference, they are preceded by a * in figure 2. In the text, content selection heuristics are preceded by a • and rhetorical selection heuristics by a o.

[7] What Vander Linden calls rhetoric demotion [16].

(4) Introduire la cassette (*vérifier que la languette de la vidéocasette n'a pas été enlevée.*)
Insert the cassette (*check that the tab of the video cassette has not been removed.*)

o Material conditions that pertain to the type of device/system are presented as often by a purpose relation or by a c-condition.

(5) *Pour un commutateur ordinaire* [...], touchez la vis de la borne de cuivre avec la pince du vérificateur.
For an ordinary switch [...], touch the screw of the copper terminal with the pliers of the checker.

o Material conditions that are difficult to evaluate, are presented by a result; and an equivalent condition, easier to test, is given and presented by a c-condition; as in:

(6) S'ils [les vis] portent la marque "L", *ils ont le filetage à gauche*, et vous devez les dévisser [...]
If they [the screws] have an "L" mark, *they have a left winding*, and you must unscrew them [...]

o In all other cases, material conditions are presented by a c-condition.

(7) *Si elle est endomagée*, il faut remplacer la douille.
If it is damaged, the socket must be replaced.

Guidances indicate how or why an operation should be performed and, at the same time, influence or guide its execution. This information is found in the task representation in the hierarchical relation between operations.

Previous work on deciding whether or not to include hierarchical relation prescribed the inclusion of all [8], or no relation [1]. According to our analysis, a guidance is generally introduced when:

- The execution of a basic-level operation depends on the execution of its parent operation (eg. for a stopping condition, a method to follow ...).

(8) Vous devez les dévissez *en tournant dans le sens des aiguilles d'une montre.*
You must unscrew them *by turning clockwise.*

- A basic-level operation requires precisions on how to execute it (eg, the reader does not know all steps, hesitate between 2 methods ...). In that case, the most important and discriminating sub-operation(s) is given. For example,

(9) Introduisez un crayon dans la conduite en provenance de la pompe à essence *afin d'éviter tout écoulement.*
Insert a pencil in the pipe from the gas pump *in order to prevent any leakage.*

Of the 120 guidances in our corpus, 72% are presented by an enablement and 28% by a purpose.

o A purpose relation is always used if more than one sub-operation is given; if there is only one, both an enablement and a purpose may be used.

o If the sub-operation specifies the use of a particular instrument or a particular way of doing an operation, an enablement is generally used (see example (8)).

o Otherwise, a purpose is generally used as in example (9).

Causalities specify what the execution of an operation causes to the current state of the world. That is, what becomes true and what is no longer true in the world. Causalities are found in an operation's post-conditions and in its parent operation. Indeed, the effect of any well executed operation is the achievement of its goal (the parent operation) and its postcondition.

- A causality is included in the text if the reader is not aware of the causal link between an operation and a postcondition or
- does not understand why a basic-level operation should be executed.

Causalities are always brought about by an agent operation and can specify an operation from the device (some reaction) or from the agent. Of the 136 causalities of the corpus, 35% are presented by a purpose relation and 65% by a result.

o Causalities specifying a device's reaction are always communicated through a result relation, as in:

(10) Presser "4", et *le canal 4 sera selectionné dans les 2 secondes.*
Press "4", and *channel 4 will be selected within 2 seconds.*

Causalities specifying an agent operation can be communicated through a purpose or a result relation. In this case, the causality justifies why a series of operations that may seem strange should be performed. They are used to satisfy the reader's curiosity and, unlike guidances do not influence the performance of the operations. Compare, for example, sentence (8) above with:

(11) *Pour protéger les bornes contre la tension*, nouez les extrémités séparées du cordon.
To protect the terminals against electric tension, tie the extremities of the wire away from each other.

o If the causality specifies an operation that the agent wishes to perform, the causality is presented by a purpose.

o If the causality indicates an operation that the agent does not know needs to be perform, the causality is presented by a result relation.

4 The SPIN System

The results of the corpus analysis have been implemented in the SPIN[8] prototype. SPIN is involved in all levels of text generation: strategic, tactical, and motor [17] levels. Generation is performed linearly with the emphasis put

[8] Système de Planification d'INstructions

on the strategic stage. SPIN builds a task representation from a top level goal and an initial description of the world using a hierarchical non–linear planning technique. The resulting hierarchy of plans is traversed breath-first by the text planner to select the semantic carriers. Then, the most appropriate local rhetorical relations are chosen. At the linguistic realization level, the actual grammatical form and position of the relations are selected based on the results of [15] adapted to French[9].

Figure 3 shows an output of SPIN. It indicates how to use the one touch recording (OTR) feature of a VCR (its English translation was given in section 1). Let us sketch the planning of this text.

From a library of operation schemas, SPIN develops the task representation of figure 1. Note that figure 1 only includes the operation names and the postconditions, whereas the actual plan representation includes the entire operation schemas.

In this task, basic-level operations are considered to be: `set` any speed, `select` any channel, and `press` any button.

The text planner initially selects the top–level goal as the title of the text. Let us deal with postconditions first: from the heuristics of section 3.2, recall that causalities from postconditions are included if the reader does not expect them. After consulting the model of the reader[10], SPIN rules out the postconditions `changed(speed)` and `changed(channel)` already expected by the reader, and decides to include all other postconditions as causalities, using a result relation.

Because `set(speed_SP)` is a basic–level operation, SPIN selects it as a sequential operation. Because it does not satisfy the special case for using c-conditions, it is presented by an action sequence (sentence 2). The reader knows how to perform the operation so its child operation is not given. `qSet(channel_4)` is a basic–level operation, but the reader is assumed to know two methods of executing it. A guidance with the sub–operation `press(channel_button)` is therefore included. Because the operation specifies a particular instrument to be used, SPIN chooses an enablement to communicate this guidance (sentence 3). `Specify(time)` is not considered a basic–level operation, but `press(OTR_button, 3_times)` is. The reader may wonder why this last operation should be done (according to his model, he does not know that this operation is done to specify the time), so a causality with `specify(time)` is included in the text. The reader knows this parent operation should be performed, therefore the causality is presented by a purpose relation (sentence 4). The 3 children of `press(OTR_button, 3_times)` are also included in the text as operation sequences because they have different postconditions included in the text. For this reason, they are presented by c-conditions in sentences 5, 6 and 7. Finally, `press(timer_button)` is a basic–level operation, it is thus included as a sequential operation and communicated through an action sequence.

```
La touche OTR                                     (1)
Réglez le sélecteur de vitesse de bande           (2)
sur ''SP''.
Sélectionnez le canal 4, en appuyant la           (3)
touche de canal.
Appuyez sur la touche OTR 3 fois pour             (4)
spécifier l'heure d'enregistrement.
Lorsque la touche est enfoncée 1 fois, PM         (5)
10:35.
Lorsqu'elle est enfoncée 2 fois, PM 11:05.        (6)
Lorsqu'elle est enfoncée 3 fois, PM 11:35.        (7)
Appuyez sur la touche TIMER dans un délai         (8)
de 9 secondes.
```

Figure 3: SPIN output: the VCR text

SPIN puts an emphasis on the planning stage, thus many aspects of the linguistic realization are left unconsidered. However, to avoid generating heavy and unnatural descriptions, SPIN can generate some anaphora (see for example the referring expression *la touche* of sentence 5 and the pronouns *elle* of sentences 6 and 7). This feature is especially useful in system/device-oriented instructions where the same objects are often used. This was done by implementing a subset of Tutin's study [14, 6].

5 Conclusion

This research is aimed at planning instructional texts from the output of an AI planner. The approach is based on a corpus study of a wide range of operator and repair/maintenance domains. It is based on a two stage process: a task planning stage, and a text planning stage. Text planning is not performed constructively through RST schemas. Rather, from the task representation a set of semantic carriers are selected, then from these, appropriate RST relations are selected.

Several aspects of instructional texts have been left aside. Repetitions, for example, do not occur frequently

[9] The positions of the rhetorical relations in both languages seem fairly similar, but the grammatical realization often differs. In addition, because we consider explanation–oriented instructions, the variety of grammatical forms is larger.

[10] The model of the reader is a library of operation schemas representing the reader's knowledge. This library is allowed to be inconsistent with the task planner's and is corrected and updated dynamically as the text is produced.

and so have not been fully considered. Although iterative operations in a task representation should be explicit on their stopping condition and their scope, pragmatic knowledge allows "natural texts" to be less specific. Consider, for example, instructions for using a shampoo: *Wet hair, lather, rinse and repeat.*

The current area of research involves analyzing how the communicative goal of the instructional text influences the content selection heuristics. For the moment, we are specifically looking at texts with different degrees of execution incentive. Texts designed for the immediate execution of the procedure seem to use different heuristics for introducing semantic carriers than explanation-oriented instructions. For example, only external causalities are included in step-by-step instructions; while both external and internal causalities are included in explanation instruction texts. We presume that the execution incentive does not influence the choice of RST relations, although a full investigation should be performed.

Acknowledgments

Many thanks to all members of the SCRIPTUM research team, and especially to Richard Kittredge and Massimo Fasciano. This work was supported by a NSERC scholarship.

References

[1] R. Dale. *Generating Referring Expressions: Constructing Descriptions in a Domain of Objects and Processes.* The MIT Press, 1992.

[2] J. Delin, D. Scott, and T. Hartley. Knowledge, Intention, Rhetoric: Levels of Variation in Multilingual Instructions. In O. Rambow, editor, *Proceedings of the ACL Workshop on Intentionality and Structure in Discourse Relations*, pages 7–10, Ohio State University, June 1993.

[3] B. Di Eugenio. *Understanding Natural Language Instructions: A Computational Approach to Purpose Clauses.* PhD thesis, University of Pennsylvania, 1993.

[4] P. Dixon, J. Faries, and G. Gabrys. The Role of Explicit Action Statements in Understanding and Using Written Directions. *Journal of Memory and Language*, 1988.

[5] P. Konoske and J. Ellis. Cognitive Factors in Learning and Retention of Procedural Tasks. In R. Dillon and J. Pellegrino, editors, *Instruction: Theoretical and Applied Perspectives*, chapter 3. Praeger, New York, 1991.

[6] L. Kosseim, A. Tutin, R. Kittredge, and G. Lapalme. Generating Anaphora in Assembly Instruction Texts. In *Proceedings of the Fourth European Workshop on Natural Language Generation*, Pisa, April 1993.

[7] W. Mann and S. Thompson. Rhetorical Structure Theory: towards a functional theory of text organization. *TEXT*, 8(3):243–281, 1988.

[8] C. Mellish. Natural Language Generation from Plans. In M. Zock and G. Sabah, editors, *Advances in Natural Language Generation*, Communications in Artificial Intelligence Series, chapter 7. Pinter Publishers, London, 1988.

[9] C. Mellish and R. Evans. Natural Language Generation From Plans. *Computational Linguistics*, 15(4):233–249, 1989.

[10] M. Pollack. *Inferring Domain Plans in Question-Answering.* PhD thesis, University of Pennsylvania, 1986.

[11] E. Rosch. Principles of Categorization. In E. Rosch and B. Lloyd, editors, *Cognition and Categorization*, pages 27–48. Lawrence Erlbaum, Hillsdale, NJ, 1978.

[12] D. Rösner and M. Stede. Customizing RST for the Automatic Production of Technical Manuals. In R. Dale et al., editors, *Aspects of Automated Natural Language Generation*, Lecture Notes in Artificial Intelligence, pages 199–214. Springler–Verlag, 1992.

[13] E. D. Sacerdoti. *A Structure for Plans and Behavior.* Elsevier, New York, 1977.

[14] A. Tutin. Lexical choice in context: generating procedural texts. In *Proceedings of COLING-92*, pages 763–769, Nantes, 1992.

[15] K. Vander Linden. *Speaking of Actions: Choosing Rhetorical Status and Grammatical From in Instructional Text Generation.* PhD thesis, University of Colorado, 1993.

[16] K. Vander Linden, S. Cumming, and J. Martin. Using System Networks to Build Rhetorical Structures. In R. Dale et al., editors, *Aspects of Automated Natural Language Generation*, Lecture Notes in Artificial Intelligence, pages 183–198. Springler–Verlag, 1992.

[17] Michael Zock. La génération interactive de langage: comment visualiser le passage de l'idée à la phrase. In J. Anis and J-L. Lebrave, editors, *Texte et Ordinateur, Les Mutations du Lire-Écrire*, pages 201–220. Éditions de l'Espace Européen, 1990.