



## Annotating the Facets of Procedural Texts

Patrick Saint Dizier, Estelle Delpech, Asanee Kawtrakul, Mugda Suktacharan,  
Patcharee Varasai

### ► To cite this version:

Patrick Saint Dizier, Estelle Delpech, Asanee Kawtrakul, Mugda Suktacharan, Patcharee Varasai.  
Annotating the Facets of Procedural Texts. SNLP'07, Dec 2007, Pattaya, Thailand. hal-02927059

**HAL Id: hal-02927059**

**<https://hal.science/hal-02927059>**

Submitted on 1 Sep 2020

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

See discussions, stats, and author profiles for this publication at: <https://www.researchgate.net/publication/228944696>

# Annotating the Facets of Procedural Texts

## Article

CITATIONS

0

READS

44

5 authors, including:



**Patrick Saint-Dizier**

Institut de Recherche en Informatique de Toulouse

166 PUBLICATIONS 574 CITATIONS

[SEE PROFILE](#)



**Estelle Delpech**

Airbus

45 PUBLICATIONS 92 CITATIONS

[SEE PROFILE](#)

Some of the authors of this publication are also working on these related projects:



ATC speech recognition [View project](#)



Comparable Corpora & Computer-aided Translation [View project](#)

# Annotating the Facets of Procedural Texts

Patrick Saint-Dizier<sup>1</sup>   Estelle Delpech<sup>2</sup>   Asanee Kawtrakul<sup>3</sup>  
Mukda Suktarachan<sup>4</sup>   Patcharee Varasai<sup>5</sup>

<sup>1,2</sup> Institut de Recherche en Informatique de Toulouse, Université Paul Sabatier,  
118, Route de Narbonne - F-31062 Toulouse cedex 9, France

<sup>3,4,5</sup> Department of Computer Engineering, Faculty of Engineering, Kasetsart University,  
Lat Yao, Jatujak, 10900, Bangkok, Thailand

Tel. +66-2-942-8555 Ext.1438, Fax.: +66-2-579-0358

e-mail : [patrick\\_saintdizier@yahoo.fr](mailto:patrick_saintdizier@yahoo.fr), [delpech@irit.fr](mailto:delpech@irit.fr),  
 [{asanee\\_naist,naist\\_da\\_da,patcha\\_matsu}@yahoo.com](mailto:{asanee_naist,naist_da_da,patcha_matsu}@yahoo.com)

## Abstract

This paper presents ongoing work dedicated to automatically annotating procedural texts, with the goal to answer procedural questions. We present annotation schemas for the pair goal-instructions, but also for other relevant structures: prerequisites, warnings, explanations and arguments.

## 1 Introduction

Annotating, manually or automatically, a procedural text means developing a kind of discourse analysis. This is not very common yet due to the complex intertwining of lexical, syntactic, semantic and pragmatic factors required to get a correct analysis (see e.g. functional discourse grammars and systemic grammars). Discourse analysis has basically a top-down organization, they take discourse acts as their basic units, instead of just words, they account for the structure and the interactions between these acts and require a relatively elaborated conceptual representation as output. Such a grammar must capture the discourse cohesion, and the discourse organization.

Procedural texts explain how to execute procedures (Yin, 2004), (Delin et alii. 1994).

In our perspective, procedural texts range from apparently simple cooking recipes to large maintenance manuals. They also include documents as diverse as teaching texts, medical notices, social behavior recommendations, directions for use, assembly notices, do-it-yourself notices, itinerary guides, advice texts, savoir-faire guides etc. Even if procedural texts adhere more or less to a number of structural criteria, we observed a very large variety of realizations, which may depend on the author's writing abilities and on traditions associated with a given domain, and which makes parsing such texts quite challenging.

Procedural texts explain how to realize a certain goal by means of actions which are at least partially temporally organized. Procedural texts can indeed be a simple, ordered list of instructions to reach a goal, but they can also be less linear, outlining different ways to realize something, with arguments, advices, conditions, hypothesis, and preferences. They also often contain a number of recommendations, warnings, and comments of various sorts. The organization of a procedural text is in general made visible by means of linguistic and typographic marks. Another feature is that procedural texts tend to minimize the distance between language and action. Plans to realize a goal are made as explicit as necessary, the objective being to reduce the

inferences that the user will have to make before acting. Texts are thus oriented towards action, they therefore combine instructions with icons, images, graphics, summaries, preventions, advices, etc.

We propose here annotation schemas that account for the structure of these texts. We also present linguistic cues that contribute to automatically identifying and tagging procedural structures.

With respect to the TEI, our view is oriented towards content analysis and also towards annotating linguistic structures. We need to tag dedicated structures as well as more common ones (e.g. verb classes) for the purpose of question-answering.

Our goal is to develop a question answering system that responds to How-to? questions. This involves unifying the question with a goal, and providing the user with the instructions associated with that goal. This is a very challenging task in general (Hovy et al., 2002), (Moldovan et al., 2000), (Maybury, 2004).

## 2 The structure of procedural texts

The answer to a procedural question is a well-formed fragment of a text, it includes in general a sequence of instructions linked by various markers (e.g., coordinators, temporal marks) or typographical marks (e.g. comma, dot, new line).

Some web pages are in structured or semi-structured format but some are in unstructured format. The main parts of procedural structured or semi-structured text can be recognized by html tags. The most difficult task is for unstructured text where the instruction is hidden in the content without marks like title, task, etc.

In this section we develop an analysis of the segment title- instruction compounds. This analysis is global and somewhat semantically formal, but we feel this is an important and necessary step for a correct understanding of such structures.

We view the instructional structure of procedural texts composed of the following items:

- titles, hierarchically organized, which express a goal to reach, realized by the actions that follow,
- instructions, associated with titles. However, instructions are not just lists of actions to perform. They often form a complex structure, presented hereafter, where there are main and subordinate instructions, comments, etc. We will therefore be here essentially concerned with an extended view of instructions, compounds of instructions, that we call **instructional compound**,
- lists of prerequisites and warnings, besides those included into instructional compounds.

Let us essentially, in this contribution, focus on the instructional compound, which is, by far, the most complex element. It has a relatively well organized discourse structure, composed of several layers. These are composed of the following items:

- The **justification and explanation structure**, which has wider scope over the remainder of the compound, indicates motivations for doing actions that follow in the compound (e.g. *in your bedroom, you must clean regularly the curtains...*, which here motivates actions to undertake, see example below).
- The **instruction kernel structure**, which contains the main instructions. These can be organized temporally or just be sets of actions (as, for example, in social behavior texts, where instructions are unordered lists of advices). Actions are identified via the presence of action verbs (in relation to the domain) in the imperative form, or in the infinitive form introduced by a modal.

- The **deontic and illocutionary force structures**: consist of marks that operate over instructions, outlining different parameters:
  - **deontic**: obligatory, optional, forbidden or impossible, alternates (or),
  - **illocutionary and related aspects**: stresses on actions: necessary, advised, recommended, to be avoided, etc.

The **conditional structure**: introduces conditions over instructions in the compound or even over the whole instructional compound. The **rhetorical structure** whose goal is to enrich the kernel structure by means of a number of subordinated aspects among which, most notably: causality, enablement, motivation, circumstance, elaboration, instrument, precaution, manner. The rhetorical structure is in general composed of instructions (satellites) related to the instructions in the kernel.

Instructional compounds are separated by means of two main devices:

- punctuation or typo-disposition marks (in .html, for example): new paragraphs, different elements in an enumeration,
- linguistic marks that indicate a strong break. Among these we have temporal marks that introduce a new phase (glosses from French): next, after 2 hours; aspectual verbs: begin by, resume, etc.; fixed forms: this being done, you can now proceed to, etc.

It is not easy, on the basis of linguistic marks, to make a distinction between marks that introduce a clear separation between instructional compounds from those which structure internally an instructional compound (outlining internal articulations, introducing forms of continuity, etc.). Those latter marks are weak separators or they

convey an idea of continuity between instructions. Weak marks are, for example: then, finally, now, when, etc.

Let us give two illustrative examples (translated from French). Here is a text extracted from the 'Home' domain. *In the bedroom, it is necessary to clean curtains. These are cleaned first with a vacuum-cleaner to remove dust, then, if they are in cotton, they can be washed in the washing machine at 60 degrees. If they are white, it is even recommended to add some bleach so that they look whiter. With some 'amidon', they can be easily ironed.*

The sequence: *In the bedroom, it is necessary to clean curtains* is analyzed as a justification of the actions to undertake. The next portion: *These are cleaned first with a vacuum-cleaner to remove dust, then, if they are in cotton, they can be washed in the washing machine at 60 degrees. If they are white, it is even recommended to add some bleach so that they look whiter* is the instruction kernel, where the last two instructions are associated with conditions. Finally, *With some 'amidon', they can be easily ironed.* is an advice.

The second example introduces a subordinate instruction: *A window appears to allow you to define the properties of the server, a password is asked, choose it with at least 6 digits. It is better to define it now since it will be asked to the user.* The first proposition describes a circumstance, it is followed by the kernel: *a password is asked*, note that this is a kind of indirect action (please provide a password), and then followed by a subordinate action of type 'precaution': *choose it with at least 6 digits*. The text ends by a motivation or justification.

### 3 Introducing Procedural Tags

#### 3.1 The Task Structure

The top level structure in a procedural text is the task structure. It may be composed of a set of subtasks (sequence of tasks). A task has a title and contains a set of instruction compounds on the one hand and possibly a number of additional structures like prerequisites and warnings on the other hand. In procedural texts, titles express goals. To account for the hierarchy of tasks and subtasks, identified by titles (goals), we use an attribute that indicates the task level. We propose the following tags for tasks:

task: `<task level="n"> .... </task>`

sequence of tasks: `<seqtask>`.

Instructions proper to a task are within the scope of a task tag.

Besides tasks, titles may introduce tasks as well as prerequisites or warnings. Their tag thus contain their type:

`<title type="X"> .... </title>`

where X = goal, prerequisite or warning.

#### 3.2 Annotation of Instruction Compounds

The tag is `<instruction compound>` includes `< instruction >` tags. Each such a tag has an attribute that describes its 'illocutionary' force. To encode deontic information, we limit ourselves to basic cases:

`<instruction type="X"> .... </instruction>`

where X = imperative, optional, or recommended. At this stage, we will not go into the instruction compound structure (which may contain a variety of data) because it is not relevant for answering procedural questions, our main aim.

Instructions are connected by means of a variety of connectors, the most noticeable being temporal connectors. Instructions are organized sequentially, but also in parallel or as alternative options (like case

structures). There is quite of a lot of literature on tagging temporal connectors (e.g. TIMEML), so we won't detail these here. We postulate that temporal connectors just connect two instructions. They may appear before two instructions or between them. It is quite frequent to have conditions associated with sequences of instructions, like a case structure: if condition1 then sequence1, if condition2 then sequence2, etc... where conditions are often mutually exclusive.

The skeleton structure is then:

```
<task>
<title type="goal" level="0"> cooking a paella
</title>
<seqtask> <task> .... </task>
<task> <title type="goal" level="1"> preparation of
the rice </title>
<instruction compound>
<instruction type="imperative"> put 2 spoons of oil
in your pan </instruction> <connector> then
</connector>
<instruction type="imperative"> add onions carefully
</instruction>...
<instruction type="option"> add spices if you wish
so. </instruction> </instruction compound>
.... </seqtask> </task>
```

The example of skeleton structure in Thai is:

```
<task>
<title type="goal" level="0">กระถางดินจิ๋ว/ Miniature
Terracotta Pots</title>
<seqtask><title type="goal" level="1">วิธีทำกระถางดินจิ๋ว/
How To Crete the Miniature Terracotta Pots
</title>
<warning type="advice">กระถางดินขนาดเล็กที่ใช้สำหรับกล้วยไม้
จิ๋วสามารถเปลี่ยนเป็นของขวัญขนาดเล็กได้/ Small terracotta
pots that are used for miniature orchids can be
turned into the cutest gift </warning>
<instruction compound>
<instruction type="imperative">บุกระถางด้วยกระดาษแก้วชิ้น
เล็กๆ เพื่อป้องกันน้ำซึมออกมา/ Line the pot with a little
piece of cellophane to stop water leaking
out.</instruction>
<connector>หลังจากนั้น/after
that<connector><instruction type="
imperative">ตัดก้อนโอเอซิลที่ยังไม่ชุ่มน้ำให้เป็นก้อนขนาด
พอดีกับกระถาง/ Cut a small pre-soaked oasis block
large enough for it to fit perfectly into the pot.
</instruction>
```

<instruction type="imperative">ตัดก้านก□หลายจั่วให้เหลือ  
ยาวประมาณหนึ่งนิ้ว/ Cut the miniature rose leaving  
about an inch of stem </instruction>  
<connector>แล้วจึง/and then<connector><instruction  
type="recommended">เริ่มจัดดอกไม้โดยเริ่มปักจากตรง  
กลางก่อน/ nd start arranging then flowers from  
the middle.</instruction>  
<warning type="prevention">ขณะจัดควรปักอย่างระมัดระวัง  
เนื่องจากก้านของดอกกุหลาบจะบอบบางมาก/ Take care  
when arranging as the stems of the miniature  
roses are quite delicate.</warning>  
<instruction type="imperative">ปักดอกกุหลาบบนโอเอซิสจน  
เต็ม/ Fill the whole oasis block with  
flowers.</instruction>  
<connector>แล้ว<connector><instruction type="imperative">ผูกริบบิ้นเส้นเล็กๆ รอบกระถาง / Tie the  
pot up with a little ribbon <connector>  
และ/and</connector>คุณก็จะได้ของขวัญชิ้นน้อยๆ มาไว้□  
there is your little gift </instruction>  
</instruction compound>  
</seqtask></task>

From the example above, the structure of a procedural text is composed of:

- 1) title which expresses goal
- 2) instruction compounds which express the actions to perform. Actions are separated by 2 means:
  - 2.1 typographic and punctuation marks, for example; bullet, new paragraph etc.
  - 2.2 linguistic marks which express temporality; แล้วจึง [laew jung], ขณะ[kana]
- 3) lists of warnings included in instruction compounds

### 3.3 Warnings and Prerequisites

Procedural texts also contain prerequisites, in general given before the instructions. Prerequisites contain lists of equipments to have, but also a few instructions for the preparation of the task. Warnings are quite different. They contain a variety of types of information from advices and rewards to preventions and threats. Warnings are often associated with arguments that explain why something needs to be done in a certain way or should be avoided (Anonymous 2006XXXXXX). We introduce the following tags:

<prerequisite>

and <warning type="X"> where X = advice, reward, prevention, threat.

Advice is the neutral type, used by default. In procedural texts, warnings can be adjoined to almost everything.

## 4 How to tag a Procedural Text?

Let us now briefly review the linguistic and typographical cues useful to automatically tag procedural texts. These are now being implemented by means of a finite state automaton, in order to be able to automatically parse the discourse structure of procedural texts.

It is clear that the criteria required to define the above structures largely varies over application domains, textual genres and the targeted audiences. We developed a small number of automata, each being able to recognize the procedural structure of a group of structurally close domains.

Each type of structure we have to tag requires a different type of analysis, because the identification criteria are very different. We briefly present them below.

### 4.1 Dealing with Instructions

Instructions are relatively straightforwardly recognized by a finite state automaton, because they have a rather typical structure. The main marks for instructions are, for French:

1. typographic and punctuation marks, and temporal marks, used for delimiting instructions, these are quite standard tags found in the literature.
2. verbs, together with their morphology (basically modal + infinitive and imperative forms are rather frequent, but depend on the domain), and their semantic class (action verbs and subclasses) since semantic class may be used to identify different types of instructions; deverbals and predicative nouns are also relevant. For that

purpose, we need to tag verbs, their tense and semantic class.

3. various classes of marks: modal marks ('you must do'), reminders ('do not forget to'), performance marks ('care about doing'), optionality or advice marks ('it is preferable to'), injunctive forms, and adverbs of manner. Most of these marks need a specific tagging that we have developed. They basically introduce either deontic or modal aspects on instructions or subordinated instructions.

These marks have a relatively universal character. However, in languages such as Spanish, it seems that verbs in instructions have a larger variety of inflected forms. Besides these elements used to identify instructions, these may contain a number of elements such as: low level goals, pictures, local warnings, hyperlinks, etc.

Thai is an isolating language. It has neither inflection nor derivation and it is also word segmentation. Thus Thai morphological processing is mainly used to recognize word boundary instead of recognizing lexicon form like English. The main marks for instruction are a slightly different.

1. Typographic especially, new paragraph delimit titles, instruction zones, prerequisites and warnings
2. Verb in imperative form is utilized as in English. However Thai have no voice form as active or passive voice as in English.
3. various classes of marks : modal marker ต้อง [tong] (must), advice marker ควร [kuan] (should) and temporal marker such as และ [lae] (and), แล้ว [laew] (then), แล้วก็ [laew ko] (and then), แล้วจึง [laew jung] (and.... then) , ต่อมา [tor ma] (after that), ถัดมา [tat ma] (from then on), ต่อจากนั้น [tor jak nan] (next), หลังจาก [lung jak] (after), หลังจากนั้น [lung jak nun] (after that). Besides some elements such as number, bullet, pictures, etc. are used....

## 4.2 Recognizing Titles

Recognizing titles is much more challenging. These have in general the form of an instruction (e.g. mounting your computer), but with a different layout. Recognizing titles is crucial for answering questions.

Titles are first identified by the typography: bold font, possibly underlined, or via the use of dedicated html marks (h1, etc.). When this is not possible, elements such as the number of words and the proximity to an instructional zone of a certain density are good heuristics. In general, titles are much shorter than instructions. Finally, we can also rely on the level of generality of the verbs used in titles, which are more generic than those found in instructions (we use the Volem verb base for that purpose). In fact, titles can be viewed as 'super-instructions', this distinction being highly domain dependent.

Prerequisites as well as warnings may also have titles. However, these two latter objects have a different typology (although they may also contain instructions), which allows us to make the distinction among types of titles, and to isolate those effectively governing instructions, to be interpreted as denoting goals.

A second problem is to identify the hierarchy of titles, which occurs in most texts of a certain length. Identifying such a hierarchy allows us to associate more precisely sequences of instructions to a goal. Since dedicated html tags are not so frequent to discriminate titles, we must rely on other factors, among which:

- presence of capital letters, or size in number of words (quite frequently 4 to 5 words, with no pronominal references),
- level of the verbs in titles: higher titles contain more generic verbs,
- identification of islands of instructions which share a quite large number of



common words (entailing a certain thematic cohesion of instructions below a title, as in Centering Theory).

- identification of summaries or introductions below titles which contain words present in subtitles.

However, these criteria largely vary from one domain and author to another, and results are somewhat inconsistent. So far, we can identify 2 levels, and it may be sufficient to answer procedural questions from texts which are not too large.

Another kind of difficulty is that titles are often elliptic (e.g. the verb is missing). In some situation they may be just absent, therefore, we may need, to answer questions, to be able to reconstruct these.

#### 4.2.1 Dealing with Warnings

Warnings as well as arguments are introduced by a range of specific verbs often in the imperative form or by negative connectors, which is quite easy to identify and to tag in French. Here are a few examples:

- negative connectors: *sous peine de, sinon, car sinon, sans quoi*, etc. (otherwise, under the risk of),
- risk verbs: *risquer, causer, nuire, commettre*, etc.
- prevention verbs: *éviter, prévenir*, etc.
- negative expressions: *de façon a ne pas, pour ne pas, pour que ... ne ...pas*, etc. (in order not to).

In Thai, verb in imperative form and negative connectors are also easy to identify and to tag. For example.....

- negative connector : ไม[mai] ไม่ได้ [mai dai] ไม่ใช่ [mai chai] มิได้ [mi dai] ไร [rai]
- risk verb: เลี่ยง [liang], หลีกเลี่ยง [lig liang], ระวัง [rawang], ระมัดระวัง [ramad rawang]
- prevention verbs : อย่า [ya] ห้าม [ham] ไม่ให้ [mai hai]
- imperative verb: จง [jong] ให้ [hai]

- negative expression: ไม่ว่า...ก็, ไม่...ก็ (*in order not*)

#### 4.2.2 Identifying Discursive marks

Temporal marks are the most frequent marks, they express: precedence, overlap, inclusion, parallelism, etc. They are mainly realized by means of adverbs, prepositions, conjunctions, aspectual verbs and propositions describing the realization of an event. Marks are tagged via the use of the TreeTagger and typed temporally more precisely via a predefined list we have elaborated.

We also have conditionals and alternative marks. These are often introduced by causal marks, prepositions or semantically closely related terms. To identify and interpret them, we use the PrepNet framework ([www.irit.fr/recherches/ILPL/prepnet.html](http://www.irit.fr/recherches/ILPL/prepnet.html)).

### 5 Perspectives

This short paper relates ongoing work on annotating procedural texts and the development of a tagger for French, capable of recognizing the above structures. Tagging is in principle language independent. However, linguistic cues may be quite different from one language to another.

The implementation proposed so far is preliminary and allows us to explore the various types of problems one may encounter when dealing with such types of texts. The corpus considered is of a rather modest size (150 procedural texts), but with quite diverse structures. It is a development corpus, allowing us to better analyse the behavior of the different components we have developed.

For Thai, we just considered 30 procedural texts that are the representative of manual instruction text from website and books. They consisted of three domains: flower

decorating, cooking and how to get these manual.

### **Acknowledgement**

We thank the ANR-RNTL program through the TextCoop project and NECTEC for supporting this research.

### **References**

- Delin, J., Hartley, A., Paris, C., Scott, D., Vander Linden, K., Expressing Procedural Relationships in Multilingual Instructions, Proceedings of the Seventh International Workshop on Natural Language Generation, pp. 61-70, Maine, USA, 1994.
- Hovy, E., Hermjakob, D., Ravichandran, D., A Question/Answer Typology with Surface Text Patterns, Proceedings of the DARPA Human Language Technology Conference (HLT), San Diego, 2002.
- Maybury, M., New Directions in Question Answering, The MIT Press, Menlo Park, 2004.
- Moldovan, D., Harabagiu, S., Pasca, M., Milhacea, R., Goodrum, R., Gorju, R., Rus, V., The Structure and Performance of an Open-Domain Question Answering System, Proceedings of the 38th Meeting of the Association for Computational Linguistics (ACL), Hong Kong, 2000.
- Yin, L., Topic Analysis and Answering Procedural Questions, Information Technology Research Institute Technical Report Series, ITRI-04-14, University of Brighton, UK, 2004.