Overview of AlethGen

José Coch
ERLI
1, place des Marseillais
F-94227 Charenton-le-pont Cedex
FRANCE
jose.coch@erli.fr

1. Introduction

AlethGen is ERLI's automatic multi-paragraph text-generation toolbox. It was first specified in 1992 and the French version developed in 1993-1994. The English version has been under development since 1995. The Spanish version is planned for 1997.

AlethGen has already been used for generating texts in several applications, notably for producing correspondence for a leading French mail-order company (see [Coch & David 94], [Coch, David & Magnoler 95]).

AlethGen is much more than a sentence generator. Its main characteristics are:

- the high quality of the multi-paragraph texts generated, in terms of fluidity, understandability, and personalisation, and
- the data-driven planning approach, which allows applications to produce an extensive set of different text structures.

AlethGen is an industrial toolbox that uses several techniques in a hybrid way, i.e. it has several modules which can be integrated and used in different ways to meet different applications' requirements. The modules are described in Chapter 2, and the architectures in Chapter 3.

Given that AlethGen is used in commercial and industrial projects, it is important to describe the characteristics of the systems it has been used to build, in terms of quality criteria and performance (Chapter 4).

Finally, the existing projects using AlethGen are described in Chapter 5.

2. Modules

2.1. Overview

The three main modules of AlethGen are the Direct generator, the Text planner, and the Linguistic realisation module.

2.2. Direct Generator (AlethGen/GD)

The main functions of the Direct generator are:

- to plan the structure of the text in a direct mode (top-down), thanks to a conditional script language using a traditional (algorithmic) approach,
- to generate more or less fixed expressions or non-linguistic texts (i.e. tables, addresses, lists, etc.) by manipulating character strings, also using a traditional conditional approach.

The Direct generator can be used without the other modules to generate texts in an automatic but non-linguistic way. Reiter [Reiter 95] calls this technique "the template approach".

The content of the knowledge bases and scripts used by the Direct generator depends on the application.

2.3. Text Planner (AlethGen/Pla)

The function of the Text planner is to plan the text in a data-driven mode. The input of this module is structured data from the application. The content and format of the input thus depend on the application.

The Text planner uses declarative knowledge bases containing rules written in a logical formalism.

The output is an ordered list of Events, with rhetoric features and rhetoric operators (the « surface structure » of the text). This module is divided in two sub-modules: the Conceptual planner and the Rhetorical planner.

Thus, the content-determination and rhetorical planning functions are not integrated in AlethGen, but separated in two different sub-modules. On one hand, AlethGen's Rhetorical planner produces surface rhetorical representation (not intermediate, as for example in RST [Mann & Thompson 88]). On the need for surface rhetorical representation, see [Coch & David 94]. On the other hand, Rhetorical planning depends on the language, whereas Conceptual planning does not. In this way, separation between these sub-modules is useful for multilingual applications, in which several Rhetorical planners (one per language) use the output of a unique Conceptual planner (as in the MultiMeteo project: see below).

Conceptual Planner

The Conceptual planner sub-module performs the "from data to concepts" step. The output of the Conceptual planner is the deep structure of the text, where the events to be realised are selected, linked by conceptual relations, but not yet definitively ordered. The Conceptual planner uses

conceptual rules which depend on the application (but not on the language).

For an overview of the Conceptual planner, see [Coch & David 94].

Rhetorical Planner

The sub-module that calculates the surface order between the events is called the Rhetorical planner. This sub-module chooses concrete surface operators (such as "because", "thus", "if", "then", "and", etc.), modalities ("can", "must", counterfactuals, etc.) and order, according to rhetorical rules. Choices depend on certain attributes, e.g. whether or not the addressee is aware of an event, whether or not an event is in the addressee's favour, etc. The Rhetorical planner uses rhetorical rules which depend on the language and style of the texts to be generated.

2.4. Linguistic Realisation (AlethGen/GL)

The function of the Linguistic realisation module is to produce the output text from its surface structure. This module can be divided into two sub-modules: the planning of noun phrases and anaphora, and the sentence-by-sentence linguistic realisation proper.

Planning of noun phrases

The output of the previous stage (the surface structure of the text) may contain repetitions of objects. It would, of course, be unacceptable to repeat noun phrases referring to the same object without any control.

Introducing an object in a text may also require a definite description or simply a definite article. These problems are solved by the noun-phrase planning sub-module. For a description of this submodule, see [Coch & Wonsever 95].

Sentence-by-sentence Linguistic Realisation

This sub-module is inspired mainly by the Meaning-Text Theory (as developed for example in [Mel'čuk 88]).

The AlethGen Generation Grammar is composed of several sets of rules, defining the transition between the different levels of representation: Events -> Semantic -> Deep Syntactic -> Surface Syntactic -> Morphology. Intermediate representations and transition rules are written in a very general formalism, such as feature structures.

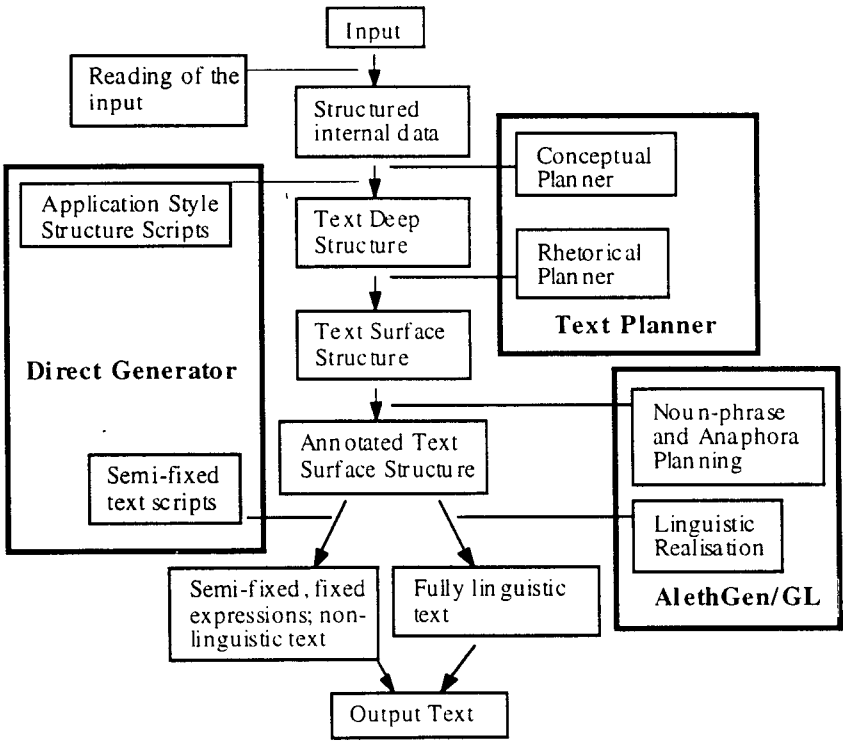
The introduction of the Events level, which does not exist in the Meaning-Text Theory, was suggested by other projects, and is required for making a true distinction between the representation resulting from the application and linguistic semantics, thus ensuring the tool's portability. This distinction is also desirable for multilingual processing.

There is a general version of the Grammar, but it needs to be adapted to each new application.

3. Architectures

3.1. Full-Hybrid Configuration

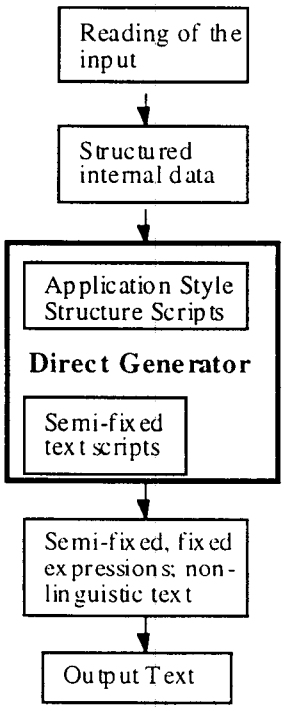
The following is a (simplified) view of the integration of all AlethGen modules in the standard generation process.



Thus, according to [Reiter 95] this architecture can be defined as "hybrid", because it uses both linguistic and template techniques. However, this " full-hybrid " architecture of AlethGen modules seems to be more powerful than those studied by Reiter, because here it is possible to work with both high-level conceptual and direct planning, and with both linguistic and template realisation, depending on the type of text (or part of text) to be generated.

This architecture is used by La Redoute's pilot mail-generation system (see below).

3.2. "Template" Configuration



The advantage of this architecture is that it seems to be easier, cheaper, and quicker when developing a generation system. On the other hand, its main drawbacks (as pointed out by [Reiter 95]) are in its adaptability, upgradability and maintainability when the possible realisations of the sentences vary greatly from a linguistic point of view. For these reasons, the Template architecture is useful for building « one-shot » prototypes.

This particular AlethGen architecture was used to develop a prototype weather-forecast generator in French for Météo France (see below).

4. Characteristics of AlethGen

4.1. Quality

Obviously, the quality of the texts produced using AlethGen does not depend only on the characteristics of AlethGen, but also on the way in which the tool is used for building an application, and, above all, on how precisely quality criteria and methods of evaluating them are defined.

A good example of this are the quality results of the mail-generation system built for La Redoute (see below).

A set of formal and user-oriented quality tests were planned and quality criteria defined during the first phase of the project. Examples of quality criteria are correct spelling, good grammar, understandability, fluidity, appropriateness of tone, personalisation, absence of repetition, precision of terminology used, etc (for details, see [Coch 96]).

The evaluation was carried out by an independent jury (representative of end users), which studied the quality of the various types of letter, including:

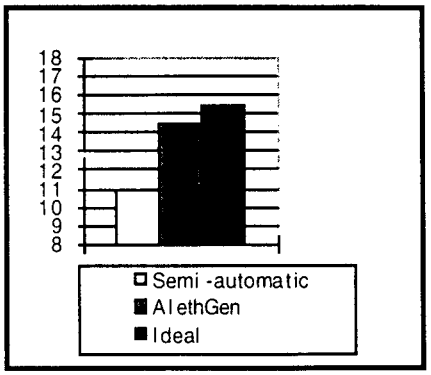
- those written by a semi-automatic fill-in-the-blanks system ("SA"), currently in use,
- those generated automatically by the pilot system based on AlethGen, and
- those written manually in an "ideal" way, by an excellent writer, without time constraints.

A report was drawn up on each letter, with values for assessment on quality criteria defined by La Redoute.

The results of the validation test show that:

- the « Ideal » letters are the best (this is not surprising!). However, the difference between « Ideal » human letters and AlethGen's letters is not that great;
- the quality of the letters generated by the pilot system using AlethGen is greater than that of the semi-automatic system, for all quality criteria (and especially for personalisation, absence of repetition, and precision of the terminology used).

These results are illustrated in the following graph (marks out of 20):



4.2. Performances

The systems built with AlethGen generate a whole text on à complex problem (15-20 sentences) in less than 2 seconds. Sometimes (as in the mail-generation project for La Redoute: see below) one or two minutes are needed for user-interface dialog.

As regards productivity gains, performance levels are to be compared with more than 5 minutes for the other approaches, and sometimes several tens of minutes for human writing.

4.3. Technical characteristics

The system was written in C++ under Unix and runs on Unix stations. The Direct generator module also runs on PC/Windows.

5. Applications

5.1. Mail generation for La Redoute

La Redoute is the leading mail-order firm in France. It receives several thousand request letters, faxes, or telephone calls each day. La Redoute and ERLI developed a real-situation pilot system for automatically replying to these requests. This system (for details on this project, see [Coch, David & Magnoler 1995]) builds a text (i.e. a letter) from data entered by the human operator processing the request, a customer database, and knowledge bases. The overall system is composed of two main modules: the Decision module and the Generation module.

The Decision module allows the writer (reading the request letter) to identify the author and subject of the request letter, ask him/her for relevant information, and suggests a decision. After validation, it communicates the relevant information to the Generation module, which automatically produces the reply letter in an SGML format. This last module was built using AlethGen tools in a full-hybrid architecture.

5.2. Weather-forecast production prototype

ERLI developed a « one-shot » weather-forecast generation prototype in French for Météo France. Weather forecasts are currently generated in a general-public style only, with a geographically and seasonally limited vocabulary. The prototype runs on PC/Windows and is integrated in a text processor (Word 6.0).

5.3. English generation for a translation tool

One of the objectives of the EUREKA GRAAL project is to construct a machine-translation engine. It uses the AlethGen's Linguistic realisation module, which can be used as a module for deep generation or surface generation from machine-translation transfer input.

5.4. MultiMeteo: multilingual generation

The goal of the MultiMeteo project is to build an automatic multilingual generation system to be used by Meteo France, Instituto Nacional de Meteorología and other European Weather Offices, for producing weather forecasts from structured data. This system will allow each European forecaster to produce texts in English, French, German, and Spanish automatically.

MultiMeteo is a 3-year project funded partially by the Language Engineering programme of the European Commission.

In each country, the MultiMeteo software will be installed and tested at 4 or 5 geographical sites, representative of different meteorological characteristics (south, north, plain, mountain, sea, etc.). In each site, 4 or 5 different styles of forecast will be developed (local general-public, regional general-public, mountain sports, sea-side sports, agriculture, aviation, etc.).

REFERENCES

[Coch & David 94]. Coch, J.; David, R.: "Representing knowledge for planning multisentential text", in *Proceedings of the 4th Conference on Applied Natural Language Processing*. Stuttgart, Germany, 1994.

[Coch, David & Magnoler 95]. Coch, J.; David, R.; Magnoler, J.: "Quality test for a mail generation system", in *Proceedings of Linguistic Engineering 95*, Montpellier, France 1995.

[Coch & Wonsever 95]. Coch, J.; Wonsever D.: "Improvement of an Algorithm for Planning and Generating Anaphora", in *Proceedings of Deixis 95*, Nancy, France 1995..

[Coch 96]. Coch, J.: "Evaluating and comparing three text-production techniques", in *Proceedings of the 16th Conference of Computational Linguistics, Coling 96*, Copenhagen, Danmark 1996.

[Mann & Thompson 88]. Mann W. C., Thompson S. A. : "Rhetorical Structure Theory: Towards a functional theory of text organization", in *Text* 8(3), 1988.

[Mel'cuk 88]. Mel'cuk I.: "Dependency Syntax: Theory and Practice", State University of New York Press, Albany, NY, USA 1988.

[Reiter 95]. Reiter, E.: "NLG vs. Templates" in *Proceedings of the 1995 European Natural Language GenerationWorkshop, Holland,* 1995.