Suregen-2: a shell system for the generation of clinical documents

Dirk Hüske-Kraus
Department of Hospital Informatics
Heart and Diabetes Center
Dirk.Hueske-Kraus@suregen.de

Abstract

Suregen-2 applications are intended for use as add-on modules for clinical information systems, generating text from GUI widgets or database content. Currently, Suregen-2 permits refinement of the predefined medical ontology, specification of text plans and description knowledge for objects of the ontology. It has built-in constructs for referential expressions, aggregation, enumeration and recurrent semantic constellations. A first application built with Suregen-2, which currently supports German only, is in routine use in our hospital.

1 Introduction

In 2000 we were faced with the following situation: our existing systems for generating medical findings had a very high user acceptance but, being programmed conventionally, had become almost unmaintainable. We needed a system linking to existing clinical information systems and generating appropriate (German) text for findings, procedure reports and referral letters. The text should be as close as possible to conventionally dictated text with regard to used lexemes, common formulations, conciseness and even the stylistic preferences of individual physicians. The system should be flexible enough to keep pace with the rapid progress of medical technology in a university hospital. The ability to mix "canned text" for normal findings and standard phraseologisms, simple templates and de novo generated text was highly desirable.

We performed a text-linguistic analysis of medical documents (outlined in (Hüske-Kraus, 2003)) with respect to generation, identifying additional requirements for our system:

> It should incorporate an extensive/extendible ontology of clinical medicine, i.e. not only signs, symptoms, diseases and bodily locations but also rather elaborate process structures of clinical actions, and even seemingly trivial items such as syringes, drugs or persons (medical personnel, patients' relatives).
>
> It should be able to handle referential expressions, especially anaphora and deicticals, aggregation of various kinds and ellipsis, but also quite frequent "enumerative expressions".
>
> Due to the vital importance of the generated text, the system designer should have maximum control over lexical choice, sentence formation and suprasentential structures.
>
> In order to give a user instant feedback of the resulting text, the generation of a 100 word paragraph should be done in under one second.

With these criteria in mind and, of course, the necessity of producing results in

German, we undertook an extensive review of the literature (see (Hüske-Kraus, 2003)) but, not surprisingly, such a system was nowhere to be found. So we decided to build one in Allegro CommonLisp, using insights and, wherever possible, techniques from the NLG domain and in so doing deliberately valuing practicability over algorithmic elegance or scientific generality.

2 The Suregen-2 architecture

Since Suregen-2 applications are to be used in conjunction with clinical information systems (host) there is an interface component which

receives administrative patient data via a standard interface(Heitmann et al., 1999),

provides a mapping of host GUI widgets/database fields to Suregen-2's object attributes, and

exchanges field values and generation results with the host (DDE or ODBC).

The Suregen-2 kernel consists of several modules:

- the (augmentable) ontology, the classes of which carry description knowledge,
- the generation algorithm,
- a rudimentary dictionary for German, especially medical terms. A simple lexical semantic for the treatment of synonyms, antonyms and selection of "preferred medical terms",
- morphosyntactical routines for the inflection of lexemes, the composition of complex phrases and clauses as well as certain suprasentential structures,
- a module handling several types of aggregation,
- generation of anaphora and other referential expressions,
- utilities for the quite frequent enumerative constructs,
- a component which allows definition and use of "semantical functions".

3 The ontology

Suregen-2 already has a base ontology of medicine. Concepts like patient, body part/organ, symptom, disease entity, pathological condition or even iatrogenic state are defined and may be augmented with concepts of the respective application domain.

One of the key concepts behind Suregen-2 is the idea of defining description knowledge (via `ToDescribe`) at the class level in the ontology. In an ideal situation it would therefore be sufficient to define the concepts of the domain as subclasses of the Suregen-2 base concepts in order to build a new application. In practise, however, it will no doubt be necessary to tailor some of the descriptions to specific application requirements.

As the description clauses for a class may contain references to other objects' descriptions (`DescribeS`), complex texts may be built up from the description of a single object. The text plan, for example, could consist of a single (complex) object's description.

Please note that the ontology is not merely a means of organizing (explicit) description knowledge but also of conveying implicit information used for generation. Thus, it suffices to declare the class `enlargement` as a subclass of `pathology` to enable generation of an appropriate description of, say, an enlargement of the left pyelon because there is a standard construct handling the "pathology at a given location" (see below).

4 The generation algorithm

In Suregen-2 a large portion of the generation logic is handled in a pipeline architecture (Reiter and Dale, 1997) by the

specific functions described below. Thus, the algorithm essentially only

- recursively substitutes occurrences of `DescribeS` in the text plan with the corresponding description clauses,
- inserts pointers for nominal references, and
- evaluates the text plan (ignoring references to `Aggregate`),
- in the next step calls to `Aggregate` are evaluated, referential pointers are resolved and then
- word and phrase order is modified.

In successive modifications to the text plan a constituent tree is built up where each node may hold information regarding syntactical form, syntactical role and base lexemes of the subtree.

5 Description templates / morphosyntacticals

The function `ToDescribe` is used to define description templates for Suregen classes. There may be several templates for a given class, varying with regard to the (syntactical) form of generation result, the style to which the description belongs and the facet of the object to be described. Consider for example[1] the class `CAD` inheriting from its superclass `SuregenDiseaseEntity` the attributes `degree` and `affectedBodyStructures`:

```
(ToDescribe :a CAD :as :NOUN
   :use (LET-Case CAD_grade
           (my :degree)
           (:unspecified "CAD")
           (1 "single vessel
disease")
           (2 "two vessel disease")
           (3 "three vessel
disease")
           (OTHERWISE :fail)))
```

[1] Please note the examples are translated into English, Suregen-2 currently only generates German text. This accounts for some inconsistencies as, for instance, in the example below "single vessel disease" is NOT a noun. The corresponding expression in German, however, is a noun: "Eingefäßerkrankung".

Wherever a description of an instance of CAD as a noun is now referred to the above clause (after :use) is inserted. Consider:

```
(ToDescribe :a CAD :as :NP
   :use (Noun-phrase
      :noun (DescribeS it :as :NOUN)
      :adjective "coronary"
      :attribute
(Parenthesized
   (DescribeS
(affectedBodyStructures it)
      :its :stenoses :as :NP)))
```

This illustrates how to describe CAD instances as noun phrases making use of the previous description (as a noun) and the description of the `:stenoses`-facet of the affected body structures. The `:use`-clause may contain an arbitrary mix of "canned text" strings, calls to Suregen-2 utilities, descriptions of other objects and calls to morphosyntactic functions. Designed to be configurable by trained end users, Suregen-2 offers grammatical constructs which could be described as an elementary grammar with defaults:

```
(Main-Clause :subject "heart"
   :predicate "to beat")
```

gives "heart beats" since :singular, 3 (:person) :present (:tempus), :active and :indicative are defaults. Suregen-2 offers inflection of words (noun-form, verb-form…), complex phrases (noun-phrase, adverbial-phrase…), clauses (main-clause-simple, relative-clause…) and supraclausal and suprasentential (hypotaxis, parataxis…) constructs. Here Suregen-2 does nothing new, at least if one disregards the medical peculiarity of building Latin and Greek plural forms for some lexemes.

6 Semantic functions

In the building of the first tentative applications with Suregen-2 it became obvious that there are certain constellations which tend to reoccur frequently in the intended domains. For instance, the quantification of a medical parameter is very common and so is the description of a certain pathological condition at a certain body location. To avoid defining

structurally identical patterns for descriptions of body temperature, heart rate or LVEDD, Suregen-2 offers the possibility of defining the constellation "parameter with value, dimension and unit" as a semantical function which can be regarded as a "macro"-construct. The semantic function sf-quantification may then generate "(a cut of) approx. 2 cm length", "(patient may only climb) three floors" or "(fever of) 39.1 °C".[2] Unlike other approaches (for instance TECHDOC (Stede, 1999)) these functions do not operate on relations specified declaratively in a formalism such as LOOM (MacGregor and Bates, 1987) but operate on instances' attributes. This decision surely trades flexibility and expressivity of a standard formalism for simplicity and performance.

7 Aggregation

As has already been observed by other researchers (Shaw, 1998a), the sublanguage of clinical medicine uses much aggregation. Suregen-2 therefore currently supports two types of aggregation, conjunction reduction ("segregatory coordination" (Shaw, 1998b) or "syntactic aggregation" (Reape and Mellish, 1999)) and conceptual aggregation (Wilkinson, 1995). The former is mainly used to aggregate noun phrases with identical (or coreferential) components. Conceptual aggregation can in our case be paraphrased as "finding a single term for a collection of terms" and is necessarily connected with (typically hierarchical) relations holding in the domain. The most common case of conceptual aggregation is that from subparts of the body to the encompassing body structure, but it may apply to nosological or microbiological entities as well. Unfortunately it is not always possible to perform aggregation along relations in the ontology since not all relations are transitive along the part-of relation (Bernauer, 1996). Consider for instance an inflammation of the left and the right lung, which might be described as "an inflammation of both lungs". On the other hand, a malignant neoplasy of the left lung and the same of the right lung might be a primary tumor and its metastatic filial tumor or two metastases of quite another tumor but never "a tumor of both lungs".

Moreover, speech habits in medicine permit aggregation even where it is unjustified by part-whole reasoning. Thus, one may speak of "global enlargement of the heart" when both ventricles and atria of the heart are enlarged, irrespective of several other structures which may not (or even cannot) be enlarged.

To account for this, Suregen-2 offers the possibility of defining "aggregator objects", in the above example an object which performs the possible aggregations given the presence or absence of a certain condition for the left and right atria and ventricles. It must be noted that this approach is viable only where there are clear cut and accepted entities in a hierarchy. To generate expressions like "a third of all red blood cells are malformed" would require quite another mechanism.

Although the necessity of defining ad-hoc aggregator objects is somewhat unsatisfactory, there seems to be no solution to this problem other than to define "artificial compound objects" in the ontology.

We were confronted with the fact that the distinction between conjunction reduction, regarded as purely syntactical, and conceptual aggregation is by no means a sharp boundary. Consider a pathological condition at various body sites. When described with noun phrases these might not be easy to aggregate because there may be different prepositions used for the prepositional object, as in "edema of the ankle", "edema in the sacral region" etc. Moreover, it might be necessary (or at least preferrable) to rephrase the latter expression as "sacral edema". We solved this problem in a very ad hoc manner by defining another aggegator object for "conditions in locations".

[2] In fact, this function is now Suregen-2 standard and permits some more parameters such as normal values etc.

8 Lexical choice

The problem of lexical choice, for which there are plenty of approaches (Cahill, 1999; Edmonds, 1997; Elhadad, 1991; Elhadad, 1996; Nogier and Zock, 1992), has more been circumnavigated than solved in Suregen-2.

Firstly, there are many proper names in the Suregen-2 target application domain, such as the nomina anatomica, diseases or drug substances, standardized e.g. in the UMLS (McCray and Nelson, 1995; Nelson et al., 2001). There is also a large body of names and acronyms common to a medical sub-specialty. Together with the lexemes which are "hard coded" in the `DescribeS`-clauses all these have one-to-one associations of concepts to lexemes.

Additionally, using a nanofunctionality of lexical semantics, Suregen-2 is able to deal with synonyms, antonyms and hyperonyms. This allows the aggregation of two adjacent noun phrases such as "Einbringen des Kameratrokars" ("insertion of the camera trocar") and "Einführen der Optik" ("introduction of the optical system") with different but synonymous heads.

The process for the production of referential expressions is based on a quite simple algorithm which is nevertheless sufficient for clinical documents. Since there happens to be a rather clear-cut thematic-rhematic structure, simple pronouns and an occasional reference using a hyperonym for the object in focus is appropriate.

The last point where lexical choice is involved is in the generation of "preferred terms". Only here are declarative statements used to allow for the selection of common terms, as in:

```
(AssertThat :a "inflammation"
   :which (Is-Located o-appendix)
   :is-called "appendicitis")
```

9 Evaluation

For a shell system such as Suregen-2, evaluation has two different, but not entirely unconnected meanings. The first would address the question of how easy it is to build an NLG application using Suregen-2. The second would focus on a prototypical Suregen-2 application considering text quality (and, of course, stability performance, etc.). We did not perform any formal evaluation in either direction, but a few remarks can nevertheless be made.

9.1 Suregen-2 as an NLG shell

Although development of Suregen-2 (which is still ongoing) and the building of the cardiology application were intricately connected, it is possible to estimate the efforts for designing an application in, say, gastroenterology. The project time for the configuration of a typical examination like a gastroscopy should amount to two to three person weeks. With an estimated total of 10 to 20 diagnostic and therapeutic procedures, the field of gastroenterology should be covered within one person year. Of course, each new application will add functionality to the generic body of Suregen-2, for instance new classes such as "ulcer" or "endoscopy" or even new semantic functions, such that subsequent applications may benefit from earlier work. The wish for individual text for each physician may, in the reverse extreme, multiply the efforts required.

It can definitely be said that, for an author who has conventionally programmed several systems with text generation functionality in routine use, Suregen-2 means less effort and better text quality. It is, however, still an open question whether this benefit will be accessible to other developers as well.

9.2 Evaluation of the cardiology application

Integration of the cardiology application into our hospital information system was seamless, robust and performant. With regard to text quality only a few options seem feasible:

A sort of turing test where physicians have to decide which texts are automatically generated.

A modified CLOZE procedure (Taylor, 1953) with gaps of varying granularity. Utilizing the CLOZE test, where persons are asked to fill in artificial gaps with missing morphemes, lexemes or even entire phrases, it would be possible to measure the accuracy of, and time used for this process, giving estimations of correctness for lexical choice and inflection, as well as aggregate measures of the coherence/cohesion of the generated text.

Another approach geared towards evaluation of text quality would draw on the fact that Suregen-2 is ontology-based. A possible test setup would consist of a physician who, having read a Suregen-2 generated text, would re-enter the information into the system which had generated the text in the first place. The corresponding states of the ontologies' instances, O_1 and O_2, could then be compared, giving a measure of how accurately the output text reflected O_1. Moreover, the time required by the physician to re-enter the information would hint at the "communicative efficiency" of the output.

We did neither of these, relying for the time being on the rather informal method of displaying generated text as a response to each physicians' mouse click in the structured data entry dialog. Considering that the findings text is manually signed by the physician who is personally responsible for the correctness of the findings' content (which may be - literally - of vital importance to the patient!), and considering further that this very text is reused in the referral letters which are, in turn, signed by the senior physician who again scrutinizes the text before countersigning it, we are positively assured that the text quality reached by the cardiology module is sufficient.

The following sample generated by the cardiology application (a demo of which my be downloaded from (Hüske-Kraus, 2002)) may illustrate this:

"Seit 1990 arbeitet Frau Test als IT-Leiterin, seit 2/2002 ist sie arbeitsunfähig. Aktuell gibt die Patientin gelegentliche Übelkeit an. Sie klagt über gürtelförmige, retrosternale Schmerzen, die in den linken Arm ausstrahlen, formal CCS III. Diese Beschwerden treten unter Ruhebedingungen und bei Belastungen auf, darüber hinaus postprandial sowie bei Wetterwechsel. Nitropräparate helfen nach ca. fünf Minuten. Dyspnoe NYHA II, Treppensteigen ist bis zu zwei Etagen möglich. Gelegentlich sowohl prätibiale als auch Knöchel- und Sakralödeme überwiegend links, keine Rhythmusbeschwerden oder Nykturie."

The text in this example is syntactically correct, uses appropriate terms like "CCS", referential expressions ("die Patientin", "Sie", "diese Beschwerden") and ellipsis. It contains an enumerative expression (after "gelegentlich") in turn containing two "preferred terms" ("Knöchelödem" and "Sakralödem"), generated by the lexical choice nano-module, sorted in a way that enables the first of the two to be hyphenated in the correct manner for compound nouns in German. In short, it does not in any way look "computer-generated", which our hospital would regard as unacceptable.

The fact that the cardiology module is in routine use at a leading cardiological center in Germany may lend further plausibility to the claim of acceptable text quality.

10 Conclusions

The main motive for developing Suregen-2 was to arrive at a working, useful system. Concededly this entailed some decisions in favor of simple, but workable solutions rather than those with more elegance or generality. With regard to referential expressions the approach chosen was sufficient; with respect to lexical choice reliance on one-to-one associations of concepts to lexems may prove too rigid. In the case of conceptual aggregation we had to resort partly to ad-hoc aggregation constructs, due to peculiarities of the medical sublanguage.

Suregen-2 is still an ongoing project. The results we have obtained so far - with regard to coverage of linguistic phenomena

in the medical sublanguage, flexibility and performance of the generation component - seem to warrant the statement not only that the architectural design of Suregen-2 is well adapted to the needs of medical document generation, but also that further efforts, especially to refine the ontology and the generic semantic functions, will be extremely worthwhile.

References

Bernauer, J., 1996. Analysis of part-whole relation an subsumption in the medical domain. Data & Knowledge Engineering, 20: 405-415.

Cahill, L., 1999. Lexicalisation in applied NLG systems. ITRI-99-04, Information Technology Research Institute, Brighton.

Edmonds, P., 1997. Choosing the Word Most Typical in Context Using a Lexical Co-occurence Network. In: Estival, Lavelli, Netter and Pianesi (Editors), 35th Annual Meeting of the Association of Computational Linguistics (ACL/EACL 97), Madrid, pp. 507-509.

Elhadad, M., 1991. Generating Adjectives to Express the Speaker's Argumentative Intent, 9th National Conference on Artificial Intelligence (AAAI 91), Anaheim, CA, USA, pp. 98-104.

Elhadad, M., 1996. Lexical Choice for Complex Noun Phrases. Machine Translation, 11: 159-184.

Heitmann, K.U., Blobel, B. and Dudeck, J., 1999. HL7 - Kommunikationsstandard in der Medizin, 1. Alexander Mönch, Köln, 56 pp.

Hüske-Kraus, D., 2002. Suregen-2.

Hüske-Kraus, D., 2003. Text Generation in Clinical Medicine - a Review. Methods of Information in Medicine, 1.

MacGregor, R. and Bates, R., 1987. The LOOM representation language. ISI/RS-87-188, USC Information Sciences Institute.

McCray, A.T. and Nelson, S.J., 1995. The representation of meaning in the UMLS. Methods of Information in Medicine, 34(1-2): 193-201.

Nelson, S.J., Powell, T. and Humphreys, B.L., 2001. The unified Medical Language System (UMLS) Project. US National Library of Medicine, National Institute of Health.

Nogier, J.-F. and Zock, M., 1992. Lexical choice as pattern matching, Knowledge-Based Syst., pp. 200-212.

Reape, M. and Mellish, C., 1999. Just what is aggregation anyway?, 7th European Workshop on Natural Language Generation (EWNLG-99), Toulouse, pp. 20-29.

Reiter, E. and Dale, R., 1997. Building applied natural language generation systems. Journal of Natural Language Engineering, 3(1): 57-87.

Shaw, J., 1998a. Clause Aggregation Using Linguistic Knowledge, 9th International Workshop on Natural Language Generation, Niagara-on-the-Lake, Canada.

Shaw, J., 1998b. Segregatory coordination and Ellipsis in Text Generation, 36th Annual Meeting of the Association for Computational Linguistics and the 17th International Conference on Computational linguistics (COLING-ACL 98), pp. 1220-1226.

Stede, M., 1999. Lexical semantics and knowledge representation in multilingual text generation. Kluwer Academic Publishers, 219 pp.

Taylor, W.L., 1953. Cloze Procedure: A new tool for measuring readability. Journalism quarterly, 30(4): 415-433.

Wilkinson, J., 1995. Aggregation in Natural Language Generation: Another look. M. Sc. Thesis, University of Waterloo, Canada, Waterloo.