Representing Conceptual and Linguistic Knowledge for Multi-Lingual Generation in a Technical Domain

Stefan Svenberg
Department of Information and Computer Science
Linköping University, S-581 83 Linköping, Sweden
e-mail: ssv@ida.liu.se

Abstract

We report on a head-driven way to generate a language-specific representation for a language-independent conceptual structure. With a grammar oriented towards conceptual rather than phrasal structure, the approach shows some advantages over previous works in head-driven generation. It is particularly suited for multi-lingual generation systems where language-independent representations and processes should be maintained to a maximum extent. We briefly sketch the architecture of our Genie system based on some results of an analysis of a technical manual for a gearbox.

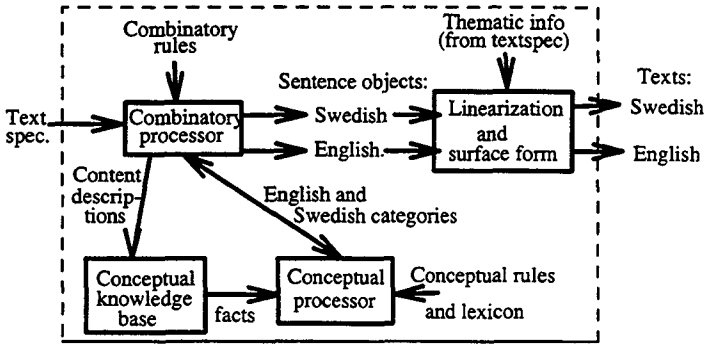

Figure 1: The architecture of Genie

1 Introduction

The Genie system explores a way to rationalize multi-lingual production of technical documentation. The system is semi-automatic in that the user designs an inter-lingual text specification describing content and form for a document. Genie constructs the document in the desired languages as modelled by the specification, matching contents to a knowledge base, constructing categories, and forming sentences according to combinatory rules.

The paper focusses on generation of language-specific categories from language independent conceptual structures.

2 The Document Analysis

We have chosen a 110-page manual, English ([3]) and Swedish ([8]), of the truck gearbox R1000 to analyse. The manual is for expert servicemen and shows the design, function, and service instructions.

The manual communicates some different kinds of domain information. We choose here to concentrate on the following two:

- Static information (i.e what something is). Examples:

  (1) The R1000 is a gearbox. (2) The gearbox has nine forward gears. (3) The gearbox is mechanically operated.

  (1) R1000 är en växellåda. (2) Växellådan har nio växlar framåt. (3) Växellådan manövreras mekaniskt

- Processive information (i.e what something does). Examples:

  (4) The purpose of the inhibitor valve is to prevent inadvertant shifting of the range gear when a gear in the basic box is in mesh. (5) The inhibitor cylinder prevents inadvertant shifting in the basic box when range shifts are being carried out.

  (4) Spärrventilen har till uppgift att förhindra växling av rangeväxeln när någon av växlarna i baslådan ligger i ingrepp. (5) Spärrcylindern förhindrar växling i baslådan när växling med rangen sker.

The text can be broken down into approximately sentence-sized units, each one communicating a piece of information considered true in the domain. We observe a tight correspondence between the kind of information and its textual realization. The carefully defined terminology not only determines words, but their combinations as well.

The text structure follows from conventions of language use for efficient communication about the domain.

These findings are in line with the issue of domain communication knowledge (Kittredge [7]). Rösner and Stede ([9]) distinguish similarly between the macro and micro structure of texts. The architecture of Genie is built up around the division of sentence and text structure; the user incorporates the conventions in the specification while Genie provides the terminological definitions.

The English and Swedish versions of the manual align at sentence level. Genie can cope with semantically non-equivalent sentence pairs, but not the very rare ones differing in content. Nevertheless, the documents correspond nicely compared to the difficulties Bateman reports ([1]) on a study of medical didactic texts. Grote and Rösner ([5]) have studied car manuals for the TECHDOC system, and they observe a close correspondence.

We have employed Functional Grammar (FG) (c.f [6]) as a principal analysis tool to developing representations for domain and language.

3 Domain Representation

Domain representation is based on conceptual structures (Sowa [11]) and the transitivity structure of FG. Concept nodes are typed in an inheritance network. We follow Sowa's definition and notation of conceptual graphs.

Next, we sketch how static and processive information are represented as facts, called aspects and transitions, respectively, in the knowledge base.

3.1 Aspects

An aspect contains a simple conceptual graph where an object has an attributive relation to a value. We define the *is-a* link as attributive and the type becomes the value. Sentence (1) and (2) are:

$[r1000] \rightarrow (isa) \rightarrow [gearbox]$
$[r1000] \rightarrow (f\text{-}gears) \rightarrow [f\text{-}gear{:}\text{coll}\{f1, f2, ..., f9\}@9]$

Both aspects happen to be close to their linguistic realizations, which is not necessarily always the case.

3.2 Transitions

A transition is a concept *trans* with three relations, *pre*, *means*, and *post*. *means* has an event as value. *pre* and *post* hold circumstances that obtain before and after the event has occurred.

An event carries mandatory, e.g *actor*, *goal*, and peripheral role relations, e.g *instr* to other objects. We can differentiate roles into subtypes, e.g *i-instr* inhibits the event.

A circumstance can be: (i) a state characterized as a setting of some variable parameter. An example is in the aspect for sentence (4):

$[trans]$ -
$(pre) \rightarrow [basic\text{-}box\text{-}gears{:}\text{disj}\{*\}]$ -
  $(in\text{-}mesh) \rightarrow [+]$
$(means) \rightarrow [range\text{-}shifting]$ -
  $(i\text{-}instr) \rightarrow [inh\text{-}valve]$

(ii) As an event, exhibited by sentence (5):

$[trans]$ -
$(gen\text{-}dur\text{-}pre) \rightarrow [trans]$ -
  $(means) \rightarrow [range\text{-}shifting]$
$(means) \rightarrow [basic\text{-}box\text{-}shifting]$ -
  $(i\text{-}instr) \rightarrow [inh\text{-}cyl]$

Sub-events have their own transitions as value for *pre* and *post*, which allows us to link events together. *gen-dur-pre* is a version of *pre* used to give a meaning to "... being carried out".

Transitions are more powerful than what has been outlined here. Much of their internal temporal constituency, complex parameters, lambda-abstractions, and different kinds of constraints have been left out for clarity.

4 Linguistic Representation

This section describes how Genie derives categories for a fact, as part of generation. We first describe English categories briefly.

4.1 Categories

Categories are expressed in a language of typed feature structures. We define how categories can be formed, their different types and content.

Construction of categories are inspired by modern Categorial Grammars (CG), such as UCG (c.f [12]), but differ in some respects. The set of categories $\mathcal{C}$ is defined recursively, (i) Basic categories $\in \mathcal{C}$. (ii) If A and B $\in \mathcal{C}$, then the complex category A|B $\in \mathcal{C}$.

The differences from CG are (i) the association of categories to facts and concepts, and (ii) complex categories are non-directed.

Categories compose using the *reduction rule* to unify:

$$A|B,\ B \Rightarrow A$$

Categories are expressed as typed feature structures (tfs) (c.f Carpenter [2]). $a(name)$ denotes the set of attributes the type *name* carries, and $s(name)$ the immediate subtypes. *cat* is the root with $a(cat) = \{\}$, $s(cat) = \{xcat, bcat\}$. *xcat* is the | operator. *bcat* are the basic categories, $a(bcat) = \{\texttt{fb}, \texttt{st}\}$, $s(bcat) = \{lcat, pcat\}$. *lcat* and *pcat* are the lexical and phrasal categories. The attribute *fb* holds some feature bundle, rooted at *fb* and named appropriately, e.g *np-fb*, *n-fb*, *agr-fb*. *st* has a FG

mood-structure to hold subcategories. A *pcat* has a certain tfs under the type *st* to encode the structure, while a *lcat* has a pointer into a surface lexicon. *s-st* is the structure for clauses. Elements are coded as attributes, e.g `subj`, `fin`, `compl` etc.

4.2 Conceptual Grammar

Facts are associated to categories composed of those obtained from the conceptual constituents. The grammar rules state that a particular domain type corresponds to a category with certain combinatorial properties. If violated, the rule cannot derive an adequate category for the fact. Concept nodes are associated to a number of categories as defined by lexical rules.

We call this a conceptual grammar, since it is tied to conceptual rather than phrase structures. The rules are language independent as the linguistic material is effectively hidden within the basic categories. Rules have the following notation:

<*head*> **when** <*body*>.

<*head*> carries an association of the general form *cs* $\Longrightarrow$ *cat*, where *cs* is a conceptual structure, and *cat* is the category. <*head*> holds whenever all constraints in <*body*> hold[1]. Help associations (arrow with a symbol on top) support $\Longrightarrow$ with extra material. We describe rules for atoms, objects, aspects and transitions.

4.2.1 Atoms and Objects

Atoms have a rather simple and direct association:

[*mechanical*] $\Longrightarrow$ *a*[**st**:*mechanical*]
[*9*] $\Longrightarrow$ *det*[**fb**:*det-fb*[**agr**:*agr-fb*[**numb**:*pl*]]
st:*n9*]

The type of category depends on how it will be used, but should be basic. The examples are typical.

The object R1000 gives "a gearbox" in:

[*r1000*] $\Longrightarrow$
cnp[**fb**:*np-fb*[**agr**:Agr=*agr-fb*[**numb**:*sg*, **pers**:*3rd*]
spec:*indef*]
st:*np-st*[n:*n*[**fb**:*n-fb*[**agr**:Agr]
st:*gearbox*]]]

There are potentially many alternative associations. Lexical choice is not addressed in this paper, although we recognize its necessity in generation systems.

4.2.2 Aspects

The category for the relation in an aspect is seen as a function of the categories for the two concepts. The grammar rule for aspects fetches and applies the function. A relation *operation*, as in the aspect for sentence (3), has a category *s*|*np*|*a*:

[*operation*] $\Longrightarrow$
s[**st**:*s-st*[**subj**:Subj
fin:*v*[**fb**:*v-fb*[**pass**:+, **agr**:Agr=*agr-fb*]]
pred:*v*[**st**:*operation*]
compl:Compl]] |
Subj=*np*[**fb**:*np-fb*[**agr**:Agr]] |
Compl=*a*[**fb**:*a-fb*[**adv**:+]]

The rule says that one category should fill the `compl` element as an adverbial, and another to become an *np* in the `subj` element. Note the subject-verb agreement.

The aspect rule simply reduces the relation category with the categories obtained from the concepts:

O=[*concept*] $\rightarrow$ R=(*rel*) $\rightarrow$ V=[*concept*] $\Longrightarrow$ A
when
R $\Longrightarrow$ A=*cat*|B=*cat*|C=*cat*, V $\Longrightarrow$ C, O $\Longrightarrow$ B.

An aspect is matched to the right hand side of the head to bind the variables O, R and V. The rule proves the following category for sentence (3):

[*r1000*] $\rightarrow$ (*operation*) $\rightarrow$ [*mechanical*] $\Longrightarrow$
s[**st**:*s-st*[**subj**:*cnp*[**fb**:*np-fb*[**agr**:Agr=*agr-fb*[**numb**:*sg*
pers:*3rd*]
spec:*indef*]
st:*np-st*[n:*n*[**fb**:*n-fb*[**agr**:Agr]
st:*gearbox*]]]
fin:*v*[**fb**:*v-fb*[**pass**:+
agr:Agr=*agr-fb*]]
pred:*v*[**st**:*operation*]
compl:*a*[**fb**:*a-fb*[**adv**:+]
st:*mechanical*]]]

4.2.3 Transitions

Associations for transitions are more complex, but still compositional. The idea is to get a category for the event and reduce it with all roles to obtain a basic category. This is reduced with the transition type category and with those for **pre** and **post** relations and values.

The association for *trans* is defined by the rule:

Trans=[*trans*] -
(*means*) $\rightarrow$ Ev=[*event*]
Pre-R=(*pre*) $\rightarrow$ Pre-C=*concept*
Post-R=(*post*) $\rightarrow$ Post-C=*concept*
$\Longrightarrow$ Res **when**
Trans $\overset{\text{type}}{\Longrightarrow}$ Res1=*cat*|Event=*cat*
Pre-R $\Longrightarrow$ Res2=*cat*|Res1|Pre=*cat*
Pre-C $\Longrightarrow$ Pre
Post-R $\Longrightarrow$ Res|Res2|Post=*cat*
Post-C $\Longrightarrow$ Post, Ev $\Longrightarrow$ Event

[1] Like a Prolog rule.

The transition is matched to bind variables in the head. $\overset{type}{\Longrightarrow}$ retrieves the complex category of one argument for the mandatory event. *pre* and *post* are optional and have their own categories, e.g:

[*gen-dur-pre*]⇒*S*|Pre=*progressive-s*| S=*s*[st:*s-st*[pre:Pre]]

The category constrains the category in the pre to be a *progressive-s*. The rule for events basically looks like:

EV=[*event*] -
  → (*mrel*) → OM_1=[*concept*]
  ...
  → (*mrel*) → OM_n=[*concept*]
  → PR_1=(*prel*) → OP_1=[*concept*]
  ...
  → PR_m=(*prel*) → OP_m=[*concept*]
⟹ RES=*cat* **when**
  EV $\overset{type}{\Longrightarrow}$ $PCAT_0$=*cat*|ARG_n=*cat*|...|ARG_1=*cat*
  for i=1..n do OM_i ⟹ ARG_i
  for j=1..m do
    PR_j ⟹ $PCAT_j$=*cat*|$PCAT_{j-1}$=*cat*|ARG_j=*cat*
    OP_j ⟹ ARG_j
  RES = $PCAT_m$

The event category reduces with the mandatory role values to reveal the innermost result category for the event. It will then reduce with the peripheral roles.

An example of an event category carried by $\overset{type}{\Longrightarrow}$

[*lock*] $\overset{type}{\Longrightarrow}$ *s*[st:*s-st*[subj:SUBJ
  fin:*v*[fb:*v-fb*[agr:AGR, pass:-]]
  pred:*v*[st:*lock*]
  compl:OBJ]] |
SUBJ=*np*[fb:*np-fb*[agr:AGR=*agr-fb*]] | OBJ=*np*

4.3 Discussion

The conceptual grammar is a semantic-head grammar, where the semantic head is the top node of the graph a rule analyzes. The grammar processor is a plain Prolog resolution. It behaves as the standard semantic-head driven generator (SHDG) (Shieber et al [10]) does when all nodes are pivots, i.e a purely top-down manner. SHDGs in general are quite different from ours in the way knowledge is organized. They follow the structure of categories in grammars that are more suitable for parsing, i.e allowing content-less words but not word-less contents. Hence, there is an assymetry between compositionality of words and semantics (Dymetman [4]). A content-less word can potentially occur anywhere in the output string and a generator must consider this to terminate gracefully. Problems of ensuring coherence and completeness degrade efficiency further. Our generator resembles a parser to a large extent, having a conceptual structure instead of a string to work on. As such, it is free from the problems and can potentially benefit directly from many research results in parsing technology.

The rules are designed to work on any language, thus lessening the burden when adding more linguistic support. More rules have to be written only when new kinds of facts are added to the knowledge base, to account for their structures. We do not need a reachability relation, as the problem of goal-directedness in generation is achieved by doing clever choices of categories in lexical rules.

The relations between domain types and categories are similar to the semantic type assignments in classic CGs. Our version is more flexible as a consequence of the type system.

Genie is in an experimental state (about 20 aspects and 10 transitions), but has proven feasability of the issues discussed in this paper. It is less competent in lexical choice and the combinatory grammar. Development is continuing in the Life environment.

References

[1] John A. Bateman, Liesbeth Degand, and Elke Teich. Towards multilingual textuality: some experiences from multilingual text generation. In *4th European Workshop on NLG*, pages 5–17, 1993.

[2] Bob Carpenter. *The Logic of Typed Feature Structures.* Cambridge University Press, 1992.

[3] Volvo Truck Corporation. *Service Manual Trucks: Gearbox R1000.* Volvo Truck Corporation, 1988.

[4] Marc Dymetman, Pierre Isabelle, and Francois Perrault. A symmetrical approach to parsing and generation. In *Proc. of Coling-90*, volume 3, pages 90–96, 1990.

[5] Brigitte Grote and Dietmar Rösner. Representation levels in multilingual text generation. In *From Knowledge to Language - Three Papers on Multilingual Text Generation*, FAW-TR-93019. FAW Ulm, Germany, 1993.

[6] M. A. K. Halliday. *An Introduction to Functional Grammar.* Edward Arnold, 1985. ISBN 0-7131-6365-8.

[7] Richard Kittredge, Tanya Korelsky, and Owen Rambow. On the need for domain communication knowledge. *Canadian Computational Intelligence Journal*, 7(4):305–314, 1991.

[8] Volvo Lastvagnar. *Servicehandbok Lastvagnar: Växellåda R1000.* Volvo Lastvagnar, 1988.

[9] Dietmar Rösner and Manfred Stede. Customizing rst for the automatic production of technical manuals. In *Aspects of Automated NLG: 6th International Workshop on NLG*, pages 199–214, 1992.

[10] Stuart M. Shieber, Fernando C. N. Pereira, Gertjan van Noord, and Robert C. Moore. Semantic-head-driven generation. *Computational Linguistics*, 16(1):30–42, March 1990.

[11] J. F. Sowa. *Conceptual Structures.* Addison-Wesley, 1984.

[12] Henk Zeevat, Ewan Klein, and Jonathan Calder. Unification categorial grammar. Technical Report EUCCS/RP-21, Centre for Cognitive Science, University of Edinburgh, Scotland, 1987.