How to Solve the Conflict of Structure–Preserving Translation and Fluent Text Production*

Karin Harbusch

Univ. Koblenz–Landau, Computer Science Department, Rheinau 1, 56075 Koblenz/Germany
harbusch@informatik.uni-koblenz.de

Abstract. Compared with a 'conventional' natural-language generation system, in Machine Translation (MT), the decisions in a what-to-say component, i.e. the selection of the content of an utterance and an adequate speech act, are made by the speaker. Although the speaker realizes the how-to-say task in the source language, i.e. does the linguistic shaping, a how-to-say component in the target language is required in an MT system. Especially, decisions in this component should be guided by the syntactic realization in the source language in order preserve the structure.
Here, we describe a flexible how-to-say component for MT. On the one hand, it accepts underspecified input with respect to content descriptions not explicitly mentioned by the speaker (e.g. rhetorical relations). On the other hand, its internal decision-making regards syntactic input in order to guide the target-language constructions according to the source utterance. In the following, we define first an input language *DRS-2* on the basis of Discourse Representation Theory. Afterwards, a generator is designed which runs *DRS-2* structures. The component is able to compare internal and external linguistic specifications in order to allow its decision-making to overrule syntactic input. The aim is a more fluent text concurrent with a more structure preserving translation.

1 Motivation

Compared with the tasks of a — so to speak 'conventional'— *natural-language generation* system, in *Machine Translation (MT)*, the decisions in a *what-to-say* component, i.e. the determination of the content of an utterance and the selection of an adequate speech act, are made by the speaker. It depends on the paradigm of the translation system whether a similar conceptual representation is modelled (cf. *interlingual* MT as, e.g. realized in KBMT [Nirenburg 89]). Although the speaker also realizes the *how-to-say* task in the source-language, i.e. does the linguistic shaping of the utterance, a how-to-say component deciding on the linguistic shaping of the utterance in the target language is required in MT. In the following, such a component which builds the target utterance out of the interlingua- or transfer-based representation is called *GENiMT* or simply generation if the context is unambiguous.

The freedom of decision-making in a GENiMT component can be reduced by linguistic specifications provided as input. In a generation system, these specifications arise from the previously uttered text. In MT, also the previously uttered translated text adds syntactic constraints to the decisions in the how-to-say component. Here additionally, syntactic constraints stated by the source utterance must be considered to realise a structure preserving translation. For instance, *structural transfer* (e.g. [Kaplan et al. 89]) determines most decisions in the how-to-say component. In other words, the GENiMT component tests whether the transfer result is syntactically realizable[2]. In contrast to this translation mode, an interlingual MT system provides a conceptual representation for which lexical choice, sentence structure and syntactic realization are determined in the GENiMT component. In between these two extreme cases, a wide range of hybrid systems exists. For instance, deeper analysis and accordingly *semantic* and *pragmatic transfer* is done only *on demand* in VERBMOBIL [Wahlster 93]. So, the input

* **Acknowledgement:** Parts of this work were funded by the German Ministry of Research and Technology (BMBF), Grant Number 01 IV 101 R at the German Resarch Center for AI (DFKI).

[2] The necessity of such a test arises from the so called *generation gap* (cf. [Meteer 90]).

for the generator differs in granularity from a conceptual representation, e.g. for prepositions, to a complete syntactic specification, e.g. for idioms.

In this paper, we describe a *flexible* GENiMT component which can deal with input provided by systems based more on transfer or interlingua. On the one hand, it accepts *underspecified* input with respect to content descriptions not explicitly mentioned by the speaker (e.g. the rhetorical relation). On the other hand, its internal decision–making regards *syntactic input* in order to guide the target–language constructions according to the source utterance.

In the following, we define an input structure called *DRS-2* on the basis of *Discourse Representation Theory* (*DRT*) (see, e.g. [Kamp&Reyle 93]). Basically, *DRS-2* descriptions are more conceptually oriented. For instance 'meeting(x)' and 'x meets y' or 'meet(x, y)', respectively, are both represented by the same *DRS-2* predicate 'meet' in order to allow linking with the conceptual representation and furthermore, to allow homogeneous processing in the GENiMT component. Additionally, predicates describing the *speech act*, the *rhetorical structure* and *syntactic constraints* are added in a *DRS-2*. Especially, in a *DRS-2* underspecifications are allowed so that interlingua– and transfer–based systems can be run our GENiMT component.

The *flexible GENiMT* component proposed in the following can basically deal with various underspecifications. The component has knowledge to produce an utterance only from conceptual input, e.g. provided by an interlingual system. More fine–grained information — e.g. produced during transfer — states *preferences* in selecting internal rules of the component.

In the next section, we compare the tasks of a how–to–say component in conventional generation and a GENiMT component. In section 3, the input language for our GENiMT component called *DRS-2* is defined. Then, we get into the individual tasks of the generator (see sections 4 and 5). Finally, future work is addressed.

2 The Task of Generation in MT

In Machine Translation, similar tasks can be observed to those of conventional natural–language generation (cf. *shake–and–bake* Machine Translation — see, e.g. [Whitelock 92]). However, generation in MT is not as free to decide on the content and on the linguistic shaping of an utterance as conventional generation is. The output of a generator should sound *natural* and must be *adequate in the situation* and *understandable* — however, it is difficult to measure these criteria. The output of a Machine Translation system should above all be *correct*. Furthermore, a good translation should be *structure preserving*.

In a conventional generation system, basically two tasks are defined. A what–to–say component determines the content of the utterance without knowing its linguistic realization and a how–to–say component shapes it linguistically. In Machine Translation, the speaker performs all these tasks. The speaker's output passes speech–analysis, syntax, semantics and/or pragmatics routines in order to be *'understood'* by the MT system. The meaning of 'understanding' the speaker, i.e. the system's input, depends basically on the translation paradigm.

In an *interlingua–based* approach, the utterance is mapped onto a *conceptual representation* (e.g. the frame–based language in [Nirenburg 89]) abstracting from the linguistic representation. Therefore — loosely speaking — the output an interlingua–based MT system provides looks like the output of a what–to–say component. In both cases an abstract conceptual representation of the utterance is handed over. Actually, in the next section we address slight differences.

In a *transfer–based* MT system, abstract descriptions of source– and target–language expressions are linked. The relation can be stated on the different levels of analysis. For instance, *pragmatic* transfer relates conventionalised speech acts (e.g. salutations) in the source and in the target language where the literal meaning is not of interest but the right tenor of politeness and honour in the target language must be realized. As another example, the German word *'basteln'* must be explained in English, e.g. by *'to do handicraft'* or *'do it yourself'*. A *semantic* transfer associates the English explanation 'DO(AGENT = HUMAN, OBJ = handicraft)' with the German technical term. Further syntactic transfer rules relate the construction of a verbal or a nominal complex in the two languages as, e.g. by (source-real = N, target-real(participle(sem)) and (source-real = V, target-real = V). Finally, *syntactic* or *structural* transfer relates syntactic realizations such as the German adverb *'gerne'* with the English verbal complex *'like to'*. The translation of *idioms* such as *'Es regnet Bindfäden'* in German into *'It rains cats and dogs'* in English determines completely lexical choice and syntactic structure. The examples demonstrate that transfer performs several tasks of a how-to-say component, i.e. provides information that normally is not handed over to a how-to-say component in conventional generation.

Accordingly, we define a *general input language DRS-2* for a GENiMT component — allowing for underspecification as well as for stating syntactic constraints, i.e. the language is independent from the paradigm of the input providing MT system. Afterwards, a flexible GENiMT component is presented which can interprete the all these kinds of descriptions.

3 Definition of *DRS-2*

Discourse Representation Theory[3] (DRT) is a widely applied formalism for representing semantics in natural-language analysis. Basically, we have chosen DRT here because the formalism allows to uniformly express syntactic and semantic knowledge. Furthermore it easily allows for *underspecification*. Certainly, extensions arise from the specific requirements in generation. A second group of new predicates (syntactic predicates) realizes the fine-grained linguistic specification according to the realization in the source utterance.

In the following, a *DRS-2* is not defined formally. Instead, we construct first *DRS-2* representations by adapting the construction principles according to [Kamp&Reyle 93]. Solely to illustrate the differences between a DRS and a *DRS-2*, transition rules are presented which transform DRSs into *DRS-2*s. In parallel, new predicates are introduced to specify the associated syntactic realization. Afterwards, the further new constructions are defined.

The two sentences S_1: *'Every farmer who owns a donkey beats it.'* and S_2: *'If a farmer owns a donkey he beats it.'* express the same meaning, but they have slightly different DRSs (see Figure 1) because the anaphora-resolution process and the deconstruction principle of relative clauses bear different numbers of variables. In $\mathbf{K}_1$, *'it'* produces a variable u. The relative

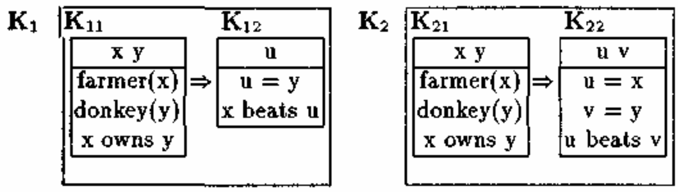

Figure 1. Differences in the DRSs for S_1 (cf. $\mathbf{K}_1$) and S_2 (cf. $\mathbf{K}_2$)

pronoun *'who'* copies the variable x from $\mathbf{K}_{11}$ to $\mathbf{K}_{12}$. In $\mathbf{K}_2$, the two personal pronouns *'he'* and *'it'* result in the two variables u and v which must be resolved.

In a *DRS-2*, all equations are eliminated and the respective variables are replaced by the defining value in all predicates, i.e. $\mathbf{K}_{12}$ and $\mathbf{K}_{22}$ both become 'x beats y' and $\mathbf{K}_1 = \mathbf{K}_2$. Additionally, by the new syntactic predicate 'anaphora(e, x)' a pronominalisation in the target language — as performed in the source language — can be suggested. For instance, $\mathbf{K}_{12}$ would look like 'e_2: x beats y, anaphora(e_2, y)'. The new predicate 'rel-sent(e_1)' where e_1: 'x owns y' specifies a relative-clause construction. The interpretation of the new predicate 'participle(e_1)'

[3] In this paper we rely on the definition of *Discourse Representation structures (DRSs)* defined in the two volumes of [Kamp&Reyle 93] in order to have a common terminological basis with the reader.

would produce *'the donkey owning farmer'*. As in a interlingua–based system, the syntactic realisation can be missing. So, it is left to the GENiMT component to choose one possibility.

Accordingly, the differences of the resulting DRS in a verbal and a nominal realization as in S_3: *'Bill's talk takes place on Sunday.'* and S_4: *'Bill gives a talk on Sunday.'* bearing different predicates as outlined in Figure 2 (skipping *'on Sunday'*) are eliminated as well. This means, we argue for a representation which is in a sense 'conceptually influenced' because the advantages of having a more homogeneous representation prevail — especially from the angle of generation and even more intensively, from the viewpoint of an interlingua-based MT system. Here, concepts are the only available knowledge source.

K_3

x e t
Bill's talk(x)
e: x takes place
n≺t, e⊆t

K_4

x y e t
Bill(x)
talk(y)
e: x gives y,n≺t, e⊆t

Figure 2. Different DRSs for S_3 (cf. K_3) and S_4 (cf. K_4)

The construction of such a general concept as 'talk' should illustrate the differences between a DRS and a *DRS-2*. The so called *DRS-4-lex* contains pairs of patterns (u ⇒ v) where u is a DRS pattern and v a *DRS-2* pattern. For instance, ([y's talk (x) ∧ e: x takes place ⇒ [pred–y(x) ∧ e: talk(x)]) and ([pred–z(x) ∧ talk(y) ∧ e: x gives y ⇒ [pred–z(x) ∧ e: talk(x)]). Additionally, syntactic predicates specifying 'cat(x) = N/V/...', 'number(x) = s/p', 'specifier(x) = NIL/def/indef/...', 'modifier(x) = y', 'person(x) = 1/2/3', 'gender(x) = m/f/n', 'lex(x) = perspron/prop_name/lex_entry', 'subject = x', 'obj_i = x' (i = 1, 2), '$pp\text{-}obj_i$ = x, $prep_i$ = y' (i ∈ $\mathbb{N}$), etc. can be added if desired[4].

For idiomatic expressions, DRS–4–lex maps onto the literal meaning (e.g. ([es(x) ∧ Bindfäden(y) ∧ x regnet y] ⇒ [rain(z) ∧ heavy(u) ∧ modifier(z, u)]. Depending on the actual syntactic transfer rules, the idiom either in the source language or the more abstract representation, can be associated with the syntactic specification 'subject = x, lex(x) = perspron, person(x) = 3, number(x) = s, gender(x) = n, lex(z) = rain, verb = z, voice(z) = act, tense(z) = pres, obj_1 = u, u = $conj_1(u_1, u_2)$, $conjunctor_1$ = and, lex(u_1) = cat, number(u_1) = p, specifier(u_1) = NIL, lex(u_2) = dog, number(u_2) = p, specifier(u_2) = NIL' which determines the idiomatic realization.

In [Kamp&Reyle 93], the genitive–construction principles (cf. CR.α's β, CR.α's) build predicates such as *'Bill's hat'* or *'z's hat'*. For all realizations of the owner–owned relation — in a very general meaning — we anticipate a common predicate 'owner–of(var_{owned}, var_{owner})'. In order to construct this representation, DRS–4–lex contains rules as, e.g. ([x's y] ⇒ [owner–of(y, x)]) and ([y of x] ⇒ [owner–of(y, x)]). Furthermore, a rule ([x owns y] ⇒ [owner–of(y, x)]) covers constructions as *'Bill is the owner of ...'* and *'Bill owns ...'*. By the syntactic predicates 'genitive(x)' or '$pp\text{-}obj_1$ = x, $prep_1$ = of' an individual realization can be addressed if desired.

Beside these cases, the syntax–independent constructions defined in DRT according to [Kamp&Reyle 93] are identically realized in *DRS-2*. For instance, the construction principles for disjunctive and conjunctive propositions. Here, elisions are not explicitly marked in the DRS. For instance, (x beats y)∧(v beats w) is produced by *'Bill beats a cat (and/,) Frank (beats/NIL) a donkey.'* where the alternatives in brackets (separated by '/') are optional. The new *DRS-2* predicate 'elision(x)' can mark a preferred realization. Furthermore, the temporal representation in volume two of [Kamp&Reyle 93] is adopted. Here, the new predicate 'tense(x) = pres/...' can specify a desired realization. So, a transfer rule can replace the English progressive form by the present tense in German and add the adverb *'gerade'* (currently). Beside these cases we now add important knowledge for generation which is not represented at all in a DRS.

[4] At least, the information of number and specifier which was directly encoded in a DRS must be provided. Otherwise, you need some kind of default handling (cf. [Harbusch et al. 94].

As an example, relations between sentences are discussed here. *'Bill will not come.* **The reason is that** *he is on holiday.'* vs. *'Bill will not come* **because** *he is on holiday.'* vs. **'Since** *Bill is on holiday he will not come.'* are some exemplified realizations of the same content. In terms of *Rhetorical Structure Theory (RST)*, the content of these sentences is represented as [NUCLEUS:= inform(intend(come(bill))), SATELLITE:= (**reason**, inform(on-holiday (bill))] — abstracting from details here. RST is a widely applied representation formalism in the generation community (see, e.g. [Moore&Paris 92]) allowing the specification of the content of an utterance, the intended speech acts and the relations between individual propositions. A main advantage of this representation is its abstract level of description.

In a natural way, the following two representations for rhetorical relations in Figure 3[5] (where all temporal descriptions are suppressed) fit into the current DRT framework. The first alternative (cf. $\mathbf{K}_5$) follows the line of defining new DRS predicates. The second one (cf. $\mathbf{K}_6$) relates two DRSs similar to the operator '⇒' in Figure 1. The new syntactic predicates 'subs = x', 'subintro = y', 'supers = x' and 'lin = $x_1...x_n/\epsilon$' can determine the sentence structure if desired.

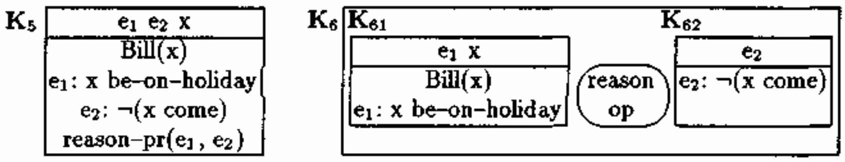

Figure 3. Different DRS:= *DRS-2* constructions for the relation 'reason_for'

If two sentences without an explicit relation are uttered (*'Bill will not come. He is on holiday.'*), in [Kamp&Reyle 93] both are integrated into one DRS which looks like $\mathbf{K}_5$ without 'reason–pr(e_1, e_2)'. In order to get a homogeneous representation in a *DRS-2* for all syntactic realizations, we propose to relate the two events by a new predicate 'sequence–pr' or an operation 'sequence–op', respectively. Accordingly, an entry in DRS–4–lex such as ([event–x ∧ event–y] ⇒ [x sequence–op y]) allows for underspecification in order to prevent the introduction of all possible relations — e.g., if an inference process in an interlingual system remains ambiguous.

Another knowledge source not represented in a DRS is the intended *speech act* (e.g., inform ↔ declarative sentence). In a DRS according to [Kamp&Reyle 93], the sentences *'Bill meets John.'*, *'Does Bill meet John?'* and *'Meet John, Bill!'* all have the same representation. For generation, the intended speech act is an unrenounceable information. Therefore, the predicates 'inform(e)', 'request(e)' and 'command(e)' are defined.

Obviously, the list of new syntactic predicates remains incomplete here because they depend on the granularity of the linguistic specification used in analysis and transfer. In a fairly natural way, new syntactic predicates can be specified in a *DRS-2*.

4 Generating from Conceptual Information

Here, we demonstrate how a *DRS-2* guides the decisions in a how–to–say component in a Machine Translation system. As outlined before, the input can range from a pure conceptual description to a complete syntactic specification of the utterance. So, on the one hand, the how–to–say component serves as a linguistic generator for interlingua–based concepts being completely free in applying its rules. On the other hand, it realizes a test for the utterance the transfer has completely determined.

As outlined in [Levelt 89], the task of how–to–say can be differentiated in the following way in order to separate different linguistic knowledge sources. *Micro planning (MIP)* realizes the further shaping of each speech act to bring it into the format required by the preverbal

[5] In our implementation, we prefer operations for reasons of modularity although it complicates reasoning on a DRS.

message. *Sentence formulation (SF)* accesses lemmas, inspects grammatical relations, and maps these onto inflectional and phrasal structure.

In terms of this specification, a DRS provides the input description for the micro planner. Within other tasks, MIP determines the sentence structure (e.g. a reason is realized as noun — presupposing a nominal lexical entry exists — introduced by *'because of'* in the main clause). Furthermore, MIP selects the sentence type — depending on the current context (e.g. perspective of the last sentence) and on pragmatic information (e.g. level of politeness).

A *DRS-2* as stated in the previous section could look as illustrated in Figure 4. Basically, in $\mathbf{K}_7$, $\mathbf{K}_6$ is extended by temporal descriptions and new DRS predicates for the individual speech acts. Here, no syntactic information occurs because it is easier to explain the construction principles from scratch. How to regard syntactic constraints is described in the next section.

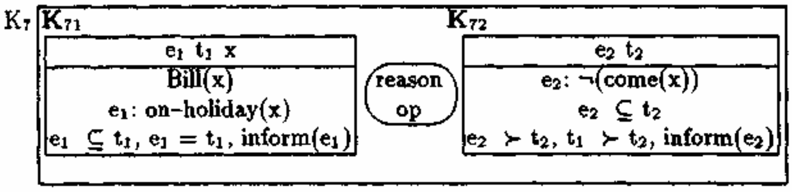

Figure 4. Example of a conceptual input DRS for a micro planner

The micro planner inspects a given *DRS-2 top-down*. First, ($\mathbf{K}_{71}$ reason–op $\mathbf{K}_{72}$) is visited. An inspected linking operation triggers the selection of a — possibly complex — sentence structure. For instance, for [a reason–op b] the corresponding rule specifies: [SUBCONJ = since, SUBS = a, SUPERS = b, LIN = ab], [SUBCONJ = because, SUBS = a, SUPERS = b, LIN = ba], [SEQUENCE(SUPERS = b, SUPERS = a), LIN = ϵ], ...

Each rule is associated with two lists of more or less general application conditions — the list of ***strong*** and the list of ***weak application conditions***. Both sets are evaluated if no syntactic predicate is specified in the input. The union of the two sets characterizes the most adequate situation to apply the corresponding alternative (e.g. (SEQ.(b, a), LIN = ba) is strongly associated with a question–answer situation as in '... *A: Will Bill join us? B: 'Oh-no. Bill will not come. He is on holiday.'*). Another strong condition here is that a verbal realization is chosen for e_1 and e_2 (preventing *'Bill will not come. Holiday.'*).

Continuing the top-down inspection, each predicate in $\mathbf{K}_{71}$ activates associated rule sets. The predicate specifying the speech act — in the example 'inform(x)' — triggers a specific set of grammatical rules. For instance, a declarative realization is activated. Lexical choice, as for 'be–on–holiday' provides a verbal construction as well as a nominalisation. Furthermore, the temporal description is inspected and results in selecting the present tense here. All activated rule alternatives for individual predicates negotiate on the actual sentence realization (see, e.g. [Kay et al. 94] for a model of ***negotiation***).

Accordingly, $\mathbf{K}_{72}$ is evaluated and forms a declarative sentence in the future tense. Since syntactic shaping heavily depends on the grammatical formalism, we skip the actual realization here but go into more detail for the case of revisions in the next section.

5 Generating under Syntactic Constraints

If a transfer component decides on lexical choice and syntactic constraints the input can look as in Figure 5. Actually, people will insist that this output would not be provided by a sophisticated transfer component. We are aware that a transfer rule which selects ***'be on vacation'*** should be blocked if a modifier — as ***'well-deserved'*** – comes along with ***'vacation'***. Instead, ***'holiday'*** should be selected. However, the example should only demonstrate the capability to revise decisions stated in the input. As described before, the micro planner inspects the input top-down. Therefore, at first, the decisions of the sentence structure are scanned. The information is stored in a list *BEST-STRUC* which collects information of the sentence structure, i.e. the syntactic predicates 'subs', 'supers', 'lin', 'subintro' etc. Next, ($\mathbf{K}_{s1}$ reason–op $\mathbf{K}_{s2}$) is read. Now, the above described process of deciding on the sentence structure — triggered by 'reason–op' — regards the constraints stated in the input in the following way. All lexical

subs = $\mathbf{K}_{s1}$, subconj = since, supers = $\mathbf{K}_{s2}$, lin = $\mathbf{K}_{s1}\mathbf{K}_{s2}$

$\mathbf{K}_{s1}$			$\mathbf{K}_{s2}$	
	$e_1\ t_1\ x$			$e_2\ t_2$
	Bill(x) e_1: on–holiday(x), $e_1 \subseteq t_1$, $e_1 = t_1$, inform(e_1), verb(e_1) = be voice(e_1) = act, tense(e_1) = pres, modifier(e_1, u), lex(u) = current, subject(x), lex(y) = vacation, num(y) = s, pp–obj$_1$ = y, prep$_1$ = on, spec(y) = poss(x), modifier(y) = well–deserved	reason op		e_2: $\neg$(come(x)) $e_2 \subseteq t_2$ $e_2 \succ t_2$ $t_1 \succ t_2$ inform(e_2),

Figure 5. Input *DRS-2* ($\mathbf{K}_s$) specifying linguistic constraints

entries for [a reason–op b] are intersected with *BEST-STRUC*. In our example, [SUBCONJ = since, SUBS = a, SUPERS = b, LIN = ba] is the only valid result of the intersection. As noted before, each rule is associated with a list of application conditions. In order to measure the *quality of the syntactic constraints* provided as input, the list of application conditions is separated into two lists: the lists of weak and strong application conditions. The current decision–making is revised if a strong condition does not hold. For instance, the question–answer–situation as stressed before can belong to the set of strong application conditions which allows overruling of the linear ordering decided for in the input. Accordingly, as a result of the revision a sequence of two sentences is preferred. The list of weak conditions is checked although *BEST-STRUC* is filled if there remain ambiguities, i.e. the syntactic constraints do not determine the utterance completely. The weak conditions select an adequate realization from the set of remaining alternatives.

In the list *BEST-REAL*, constraints specifying the syntactic realization of the sentence, i.e. syntactic predicates such as 'subject' or 'lex' and associated morphological information, are gathered. If conceptual predicates are provided as in $\mathbf{K}_{s1}$, they are mapped onto syntactic trees according to the ordinary micro planner rules. Actually, many details depend on the underlying grammar formalism. In our implementation, we apply *Tree Adjoining Grammars (TAGs)* (cf.[Harbusch et al. 91], [Harbusch 94]). For instance, TAGs provide an easy construction principle for idioms which build complete trees containing markers to allow or prevent modifications. In our example, in the tree for *'be on vacation'*, the noun phrase *'vacation'* is marked with *Null Adjoining*, i.e. a modification of this construction is blocked. If no concepts are specified all syntactic trees are activated. The syntactic constraints in *BEST-REAL* are intersected with *abstract descriptions* in terms of *DRS-2* predicates.

Here, strong and weak conditions work as previously specified. For instance, the voice in the previous sentence should influence the voice in the current sentence (as psycholinguistic experiments illustrate [Levelt 89]). Presupposing the previous sentence could only be expressed in the passive, the current sentence is also realized in the passive — caused by a weak condition.

In the example, we presuppose that no strong application condition causes a revision. Moreover, no weak application condition is activated because the intermediate result is presupposed to be unique. Actually, the idiomatic tree for *'be on vacation'* and the modifier tree for *'well-deserved'* are activated. In this situation, the combination operation of these two trees fails.

To overcome this specific type of failure is not trivial because it is difficult to figure out why the combination process failed. Specific strategies depending on the amount of information available for the GENiMT component can help here. As simple strategy, the system can switch into the mode of conceptual generation if concepts are provided as input. Sophisticated backtracking methods allow for *reusing* information previously computed, e.g. an already built complex tree as for *'well-deserved by quantities of unpaid overtime'* not concerned by the revision. A strategy if only syntactic information is available, activates rules which describe alternative realizations — again in terms of abstract predicates (e.g. [(voice = active, subj = x, obj_1 = y) $\Longleftrightarrow$ (voice = passive, subj = y, facultative(pp-obj_1 = x, $prep_1$ = by))]).

To complete our example, the idiomatic tree is replaced on the basis of a lexical choice process providing *'be on holiday'*. Now, the combination operation of the recent trees is successful, i.e. as a second element in the sequence *'Bill is currently on his well-deserved holiday.'* is produced.

Obviously, the content of the sets of strong and weak application conditions influences directly how literal the translation is. The set of strong application conditions, especially, should be designed very carefully because they overrule input specifications.

6 Final Discussion

Let us end with some general considerations. In the paper, basically interlingua- and transfer-based Machine Translation systems were addressed. Another main research topic in MT are *corpus-based* methods. The question arises if this approach can be combined with a GENiMT component. We propose to relate the two components on a *sub-sentence* level, i.e. some phrases are translated by subsymbolic methods, others are provided, e.g. by transfer. The GENiMT component must *integrate the pieces into one utterance*. Such a hybrid system would fit into an *anytime sytem* (cf. [Dean&Boddy 88]) which activates components only on demand in order to run in *real time*. Here, a corpus-based translation component on the sub-sentence level could act as a first-level translator providing preliminary results enriched or revised by a transfer component which can give its control to an interlingual system if necessary.

References

[Dean&Boddy 88] Th. Dean, M. Boddy. 1988. An Analysis of Time-Dependent Planning. In *Proceedings of the 7th AAAI*, Saint Paul, Minnesota/USA, pp. 49-54.

[Harbusch et al. 91] K. Harbusch, W. Finkler, A. Schauder. 1991. Incremental Syntax Generation with TAGs, In *Proceedings of the 4th International GI-Congress*, Munich/Germany, 1991, pp. 363-374.

[Harbusch et al. 94] K. Harbusch, G.-I. Kikui, A. Kilger, 1994. In *Proceedings of the 15th International Conference on Computational Lingustics*, Kyoto/Japan, pp. 356-362.

[Harbusch 94] K. Harbusch. 1994. Toward an Integrated Generation Approach with Tree Adjoining Grammars. *Computational Intelligence*, 10, pp. 579-590.

[Kamp&Reyle 93] H. Kamp, U. Reyle. 1993. *From Discourse to Logic: Introduction to Modeltheoretic Semantics of Natural Language, Formal Logic and Discourse Representation Theory.* Part I and Part II, Kluwer Academic Publishers, Dordrecht/The Netherlands.

[Kaplan et al. 89] R.M. Kaplan, K. Netter, J. Wedekind, A. Zaenen. 1989. Translation by Structural Correspondences. In *Proceedings of the 4th Conference of the European Chapter of the Association for Computational Linguistics*, Manchester/U.K., pp. 272-281.

[Kay et al. 94] M. Kay, J.M. Gawron, P. Norwig. 1994. *Verbmobil: A Translation System for Face-to-Face Dialog.* CSLI, Stanford, CA/USA.

[Levelt 89] W.J.M. Levelt. 1989. *Speaking.* The MIT Press, Cambridge, MA/USA.

[Mann&Thompson 88] W.C. Mann, S.A. Thompson. 1988. Rhetorical Structure Theory: Towards a functional theory of text organisation. *Text*, 8, pp. 243-281.

[Meteer 90] M.W. Meteer. 1990. The 'Generation Gap': The Problem of Expressibility in Text Planning. Technical Report 7347, BBN Systems and Technology Corporation, MA/USA.

[Moore&Paris 92] J.D. Moore, C.L. Paris. 1992. Planning text for advisory dialogues: Capturing intentional and rhetorical information. Technical Report 92-22, University of Pittsburgh, PA/USA.

[Nirenburg 89] S. Nirenburg. 1989. Knowledge-based MT. *Machine Translation* 4, pp. 5-24.

[Wahlster 93] W. Wahlster. 1993. VERBMOBIL: Translations of Face-to-Face Dialogs. In *Proceedings of the MT Summit IV*, Kobe/Japan, pp. 127-135.

[Whitelock 92] P. Whitelock. 1992. Shake-and-Bake Translation. In *Proceedings of the 14th Conference on Computational Linguistics*, Nantes/France, pp. 610-616.