International Workshop on Parsing Technologies
Symposium of Pacific Association
for Computational Linguistics

Fine-grained Utterance Delimitation and Organization in Incremental Explanation Generation

INUI Kentaro, SAKANIWA Katsuyuki, ISHIZAWA Hiroaki,
TOKUNAGA Takenobu, and TANAKA Hozumi
Graduate School of Information Science and Engineering
Tokyo Institute of Technology
2-12-1 Ôokayama Meguro Tokyo 152 Japan
{inui,sakaniwa,ishizawa,take,tanaka}@cs.titech.ac.jp

Abstract

In dialogue, the overall information that is to be conveyed is usually decomposed into smaller chunks of information and conveyed individually by a sequence of fine-grained utterance units. An utterance unit may be a sentence, or alternatively a phrase or even a word. This aspect of dialogue has received little attention for existing explanatory dialogue systems. In this paper, we first review the state-of-the-art text generation model, and then discuss how to enhance it in order to implement a fine-grained incremental utterance generator, presenting a new three-layered model that consists of the content planner, utterance planner, and utterance realizer. In this model, (a) the content planner and utterance planner interact through fine-grained utterance goals, and (b) the utterance planner performs such tasks as utterance delimitation and organization through unification-based aggregation of fine-grained utterance goals.

1 Introduction

As many researchers have pointed out, task-oriented dialogues such as explanatory dialogues should be incremental and interactive (e.g. [4,17]). These two requirements come from an important aspect of communication: the interlocutor can never be certain of the correct model of her counterpart's beliefs or knowledge. Therefore, the interlocutor needs to negotiate as to what relevant information is necessary and to what extent detailed information is required. The interlocutor does this by giving or requesting incremental development of the ongoing information exchange. The dialogue systems variously designed by Cawsey [4], Moore [17], and Carletta [3] are good examples of systems that address this issue.

Dialogue has another important facet: the grain size of primitive utterances (which we term *utterance units* in this paper). In dialogue, the overall information that is to be conveyed is usually decomposed into smaller chunks of information and conveyed individually by a sequence of utterance units. An utterance unit may be a sentence, or alternatively a phrase or even a word. This aspect has received little attention for existing explanatory dialogue systems. Most existing systems are designed under the assumption that the grain size of an utterance unit is no smaller than a clause, with phrase-sized utterance units interpreted as elliptical expressions (e.g. the three systems cited above). This is mainly because the linguistic theories underlying existing plan-based dialogue systems, such as Mann and Thompson's RST and Austin's speech act theory, assume that a speech act is realized by a single utterance unit, which is defined to be the size of a clause.

However, we have several good reasons to reconsider this conventional assumption. In explanatory dialogue, the information recipient gets the chance to give feedback, such as a back channel cue. between the information provider's utterance units. In fact, it has been observed that the information provider usually utilizes particular prosodic patterns to let the the recipient know that the current utterance unit has finished, thereby prompting the recipient's feedback (e.g. [16]). This implies that the provider can to some extent control the timing and frequency of the recipient's feedback. The smaller the grain size of an utterance unit, the more often the recipient can give feedback. Allowing the recipient to give frequent feedback would improve the efficiency of communication in settings of the following types:

- the information that is to be conveyed is complicated,
- the information provider has great difficulty in modelling what the recipient already knows,

- the channel is noisy,
- the recipient is required to understand the conveyed information correctly (e.g. in order to accomplish a given task).

Furthermore, it is reported that in Japanese dialogue, back channel cues tend to be given much more frequently than, for example, in English dialogue [5]. Thus, in designing Japanese dialogue systems, realization of fine-grained utterance units is further significant.

Motivated by these facts, we are trying to construct a computational model of an explanatory dialogue system that can interact with the user on a more fine-grained level than existing systems. Among the issues we need to consider to realize such an interactive system, our focus has been so far on the task of generating fine-grained utterance units. To realize this task, one would need to take the following requirements into account.

First, in fine-grained interactive dialogues, a single phrase, or possibly a word, can constitute an utterance unit, and even such a fine-grained utterance unit can be considered to be produced founded on one or more independent goal(s). What should be noted here is that such goals can be more fine-grained than those handled in existing systems. For example, the goal of an utterance unit may simply be the shift of the discourse focus or reference to an object. Given the fact that, in response to this type of fine-grained utterance unit, the recipient can give feedback such as acceptance, a follow-up question, etc., we claim that the system should be able to handle fine-grained utterance goals corresponding to such fine-grained utterances so that the system can infer which goals have been attained and which goals have not.

Second, by way of distributing the information to be conveyed over a set of fine-grained utterance goals, the system might need to aggregate some of those goals again in order to reduce the redundancy of utterance. The question is then in what condition fine-grained utterance goals can be aggregated and realized as a single utterance unit. This can be considered as a matter of utterance delimitation and organization. We call this generation subtask *utterance planning*. As a minimum requirement for controlling this aggregation task, i.e. utterance planning, we need to take the following factors into account:

- *Feedback prompting:* The information provider suspends an utterance, attaching a feedback prompt at its end, if she needs to confirm the attainment of the intended goals.
- *Efficiency:* The provider tends to try to include as much content as possible in a single utterance unless she needs the recipient's feedback.
- *Realizability:* Each delimited content must be linguistically realizable (i.e. there must be some way of realizing each content).
- *Conciseness:* If the original content is too complex to realize as a single utterance unit, it should be subdivided.
- *Cohesiveness:* Neighboring fine-grained utterance units should be locally cohesive.

Of the above five factors, the former two constraints are obviously related to the intentional structure of dialogue, whereas the latter three constraints additionally associate the task with linguistic constraints. Thus, the main issue in utterance planning is what architecture would facilitate the process of effectively compromising between these constraints.

Although several attempts of incremental generation have already been reported (e.g. [7–9, 22]), these do not directly address design issues for fine-grained incremental utterance generation in interactive settings. de Smedt and Kempen [7, 8] and Reithinger [22] both focus their attention on monologic text generation, and thus do not consider, for example, the factors such as feedback prompting and cohesion between fine-grained linguistic units. Dohsaka's generation system [9], on the other hand, explicitly represent utterance goals (plans) that are more fine-grained than conventional speech act goals (plans) in order to realize incremental generation. However, Dohsaka's system incremantally generates utterances not for the purpose of prompting user feedbacks, but due to the time constraint such that the speaker should not keep silent for a long time. However, unless simulation of human behavior is concerned, the time constraint may not be the primary factor for utterance delimitation since the computational cost strongly depends on the machine, algorithm, etc. Furthermore, Dohsaka's system has no mechanism for aggregation of fragmental plans, which is supposed to be important in fine-grained incremental generation.

In this paper, we present a computational model that can incrementally generate a sequence of fine-grained explanatory utterances. Starting with the state-of-the-art three-layered text generation model, we discuss how to enhance it in order to implement a fine-grained incremental utterance generator that takes all the above requirements into account. In what follows, we first discuss the issues that need to be considered in fine-grained incremental utterance generation based on our corpus analysis in Section 2. We next review previous three-layered text generation models as the basis of our model, and present the overview of our model in Section 3. We then elaborate our model in Section 4 and 5. Throughout this paper, we restrict our discussion to explanatory dia-

P1: *de soko-kara mata*	P1: and, from there, again
R1: *hai*	R1: yeah
P2: *hidari-ni massugu*	P2: to the left, straight
R2: *suihê-de în-desu-ka?*	R2: is that horizontal?
P3: *hai suihê-ni gorira-no atari-made it-tekudasai*	P3: yeah, move horizontally to the gorilla
R3: *a gorira imasen*	R3: um, I can't find the gorilla
P4: *a gorira imasen*	P4: you can't find it
R4: *hai*	R4: no
P5: *â sô desu-ka êto soredewa ...*	P5: O.K., then, ...

Figure 1: An example of a typical interaction appearing in the map task dialogue

logue, assuming the system and user to be an explanation provider and explanation recipient, respectively.

2 Fine-grained Utterances

To explore how human interlocutors produce fine-grained utterances, we analyzed several dialogues (roughly two hours long in total) from the Chiba University map task dialogue corpus[1] and the NTT-ISL route navigation dialogue corpus [18], both of which are collections of Japanese dialogues.

The dialogue in Figure 1 is an example of a typical interaction appearing in the Chiba University map task corpus. Here, the explanation provider P is describing the route specified on her map. The recipient R also has her own map but has no information about the route. The required task is to exchange information so that R can draw that route provided to P on her own map. The interaction can be complicated by the fact that the two maps do not necessarily share all the same landmarks. In P1 and P2 in the example dialogue, P describes the starting point, direction and form of a certain part of the route. Following this, R asks a follow-up question in R2, and P gives an answer followed by extra information about the destination in P3. However, since R does not understand the reference point "*gorira* (gorilla)", a remedial subdialogue is entered into.

The dialogue in Figure 2 is an example of a typical interaction appearing in the NTT-ISL route navigation dialogue corpus. Here, the explanation provider P explains to the explanation recipient R the route from a certain place to his/her house. The corpus is a collection of dialogues classified according to five types of different dialogue settings, from which we chose only those dialogues where the interlocutors exchange information without any map, eye contact, or body gestures. As compared to the setting for the map task dialogue are that, in the NTT-ISL corpus, the interlocutors do not have any map, and they talk about routes existing in the real world.

Although the dialogue settings are slightly different as mentioned above, the dialogues in the above two corpora seem to share most essential features; in particular, they are quite similar in terms of the patterns of utterance delimitation and organization.

First, in fine-grained interactive dialogues, a single phrase, or possibly a word, can constitute an utterance unit, and even such a fine-grained utterance unit seems to be intended to achieve one or more independent goal(s). For example, utterance P1 in Figure 1 can be considered to have been produced in order to shift the discourse focus to the next subsegment of the overall route, and, at the same time, to convey the information that that route segment starts from the place referred to as "*soko* (there)". Utterance R2 shows that the recipient understood the fine-grained utterance goals of P1, by way of giving an acceptance feedback. Similarly, utterance P2 can be considered as an informing action in itself, which conveys the direction and shape of the current route segment. The same applies with the dialogue in Figure 2. For example, utterance P1 can be considered to have been produced with a focus shift goal, while utterance P6 was produced only to convey a part of the attributes of the current route segment. These utterance goals are clearly more fine-grained than any conventional speech act goal such as informing the hearer of a proposition, requesting the hearer to execute an action, etc. In order to realize fine-grained interactions, the system should be able to handle such fine-grained utterance goals, and use them to infer which goals

[1] We analyzed a sort of "beta version" of the corpus (http://cogsci,l.chiba-u.ac.jp/MapTask/), of which only several dialogues are available to the public. The example presented here is from that open transcription.

P1: *mazu moyorieki-wa*	P1: first, the closest station
R1: *hai*	R1: yeah
P2: *Keiôsen-no Hatsudai-toiu eki na-n-desu-ga*	P2: is called Hatsudai, st the Keio line
R2: *Hatsudai*	R2: Hatsudai
P3: *hai Shinjuku-kara mittsume-no eki desu*	P3: yeah, [it] is the third station from Shinjuku
R3: *hai*	R3: yeah
P4: *de Hatsudai-no kitaguchi-o deru-to*	P4: and when you go out the north exit
R4: *hai*	R4: yeah
P5: *sugu mae-ni dôro-ga aru-node*	P5: you'll find a road directly ahead
R5: *hai*	R5: yeah
P6: *soko-o migi-no hô-ni*	P6: which [you turn] to the right onto
R6: *migi*	R6: the right
P7: *hai migi-no hô-ni hyaku-mêtoru-hodo ikimasu*	P7: yeah, you go right for about 100 meters
R7: *hai*	R7: yeah

Figure 2: An example of a typical interaction appearing in the route navigation dialogue

have been achieved and which goals have not on a pre-determined fine-grained level.

Second, it should also be noted that a single unit utterance is usually intended to attain more than one fine-grained utterance goal simultaneously. As mentioned above, utterance P1 in Figure 1, for example, can be considered to have been produced founded on a focus shift goal, a reference goal, and possibly a confirmation goal. This implies that introducing fine-grained utterance goals would require some mechanism for aggregation of such fragmental goals of the system.

Third, most of the current dialogue systems utter a clause as a unit of utterance and use cohesive devices such as connectives and meta comments to make salience the relationship connecting those clauses. However, in fine-grained interaction, cohesive devices used to connect fine-grained utterances tend to be sometimes rather implicit. For example, one can easily infer the relation between utterances P1 and P2 in Figure 2 since these phrases can be seen as constituents of the same identifying clause, the relation having been indicated by the grammatical case marker "*wa* (TOP/NOM)" in P1. The relation between P2 and P3, on the other hand, can be considered to be an elaboration relation, which has been indicated by the particular ending pattern of P2 "*-ga* (CONTINUE)". The system would also need to take such kinds of local cohesion between fine-grained utterances.

3 The Basic Architecture

Starting with what is called sentence planning (e.g. [14, 20, 21]) is one promising approach to utterance planning, since the utterance planning task inherits most of its features from sentence planning. In general, sentence planning involves such processes as sentence delimitation, local discourse organization, phrase ordering, and choice of referring expressions, playing the role of bridging the gap between the content planning and sentence realization tasks.

Such three-layered generation models have several advantages, among which the most significant is in the modularity of each layer. Content planning tends to be domain-dependent; for example, the plan library may be replaced depending on the domain, leading to the ontology of the content representation also being domain-dependent. In contrast, the sentence realizer can be expected to be designed as a domain-independent general module similarly to such existing sentence generators as Penman and its multilingual extension KPML [2]. By introducing the sentence planner as the bridge between these two modules, one can maintain the domain-independency of the sentence realizer.

However, since previous three-layered models were originally designed for monologic text generation, we need to enhance them in the following three respects:

- *Feedback prompting*: As mentioned in Section 1, the system may suspend an utterance, attaching a feedback prompt at its end, in order to confirm the attainment of the intended goals. Obviously, this factor strongly influences utterance delimitation. However, in previous models, consideration of this factor is lacking, and it needs to be newly taken into account.

- *Fine-grained utterance goals*: Our model deals with utterance goals that are more fine-grained than any conventional speech act goal.
- *Aggregation-based utterance delimitation and organization*: Since the information to be conveyed is distributed over a set of fine-grained utterance goals, in utterance delimitation and organization, the aggregation of such information pieces is necessarily the centered task. Our model facilitates this process by employing both a well-founded grammatical theory and a well-founded unification-based formalism.

In our enhanced model, we call the intermediate and linguistic modules the *utterance planner* and *utterance realizer*, respectively, to distinguish our model from previous models. Our model is illustrated in Figure 3. The content planner infers what to say, producing a set of *fine-grained utterance goals* (FUGs). Before content planning completes, the utterance planner starts to consume the FUGs, generating a sequence of *utterance plans* (UPs), which are then handed to the utterance realizer[2]. Thus, these three modules theoretically work in parallel as in, for example, De Smedt and Kempen's incremental generation model [7] and Reithinger's POPEL [22]. The utterance planner is responsible for utterance delimitation, lexical choice, phrase ordering and attachment of cohesive devices. To perform these tasks, the utterance planner first generates fragmental utterance plans (what we call *atomic utterance plans* AUPs). The knowledge for mapping from FUGs to AUPs is described as a set of production rules (*AUP generation rules*). The utterance planner then aggregates AUPs to produce fully specified UPs, which possibly include some delimitation constraints. Finally, receiving an UP from the utterance planner, the utterance realizer first generates the corresponding surface structure, and then articulates its subconstituents, delimited by the delimitation constraints. If the user gives acceptance or no feedback to that utterance, the utterance realizer continues to generate the subsequent subconstituents; otherwise, the utterance planner requires the content planner to produce a revised set of FUGs.

The utterance planner and utterance realizer have been implemented for the domain of the NTT-ISL route navigation corpus. As we mention below, we represent (fine-grained) utterance plans as typed feature structures, and realize the tasks of utterance planning and utterance realization by means of unification-based operations. To implement such a unification-based generation system, we used the CUF typed unification system [10]. The CUF system is also directly used to implement the lexicogrammatical knowledge, which we describe and maintain in the systemic fashion, in a manner analogous to that proposed by Henschel [13]. The resources are empirically proven to be sufficient to generate several tens of patterns of utterance units, covering the dialogues we analyzed.

In the following sections, we first discuss design issues for FUGs, and then describe the process of generating fine-grained utterances, focusing on the utterance planning process. Design issues for content planning are beyond the scope of this paper. We simply assume that the utterance planner is inputted with a set of FUGs generated from the content planner, whether plan-based or schema-based, at each stage of the dialogue.

4 Fine-grained utterance goals

As mentioned above, we consider the representation of FUGs as the domain-dependent interface between content planning and utterance planning. Thus, the types of FUGs one needs to prepare, and the appropriate grain-size of an FUG is assumed to be domain-dependent. However, one may be able to consider certain general principles for designing an FUG type set as follows.

First, our motivation in introducing FUGs immediately leads us to stipulate that FUGs should be fine-grained enough that the system can represent which FUGs have been attained by each single fine-grained, and thus fragmental, utterance.

Second, it is important to analyze the intention of each fine-grained utterance from the interpersonal and presentational perspectives, and not only the informational perspective. Let us look at the dialogues in Figure 1, and 2 again. According to our corpus analysis, each utterance unit can be considered to have been produced to attain one or more FUG(s), with the range of utterance goal types diverse even in such a restricted domain as route navigation. For example, utterance P2 in Figure 1 can be considered to have been produced with an informational utterance goal to convey attributes of the current route segment. On the other hand, utterance P1 in Figure 1 can be seen as the realization of a focus shift goal, which is related to the presentational aspect of dialogue. Furthermore, P1 can also be considered to be associated with a goal to confirm whether that fo-

[2] There also should be a fourth module, the utterance articulator, below the utterance realizer. Despite noticing the importance of utterance articulation, we do not discuss it in this paper.

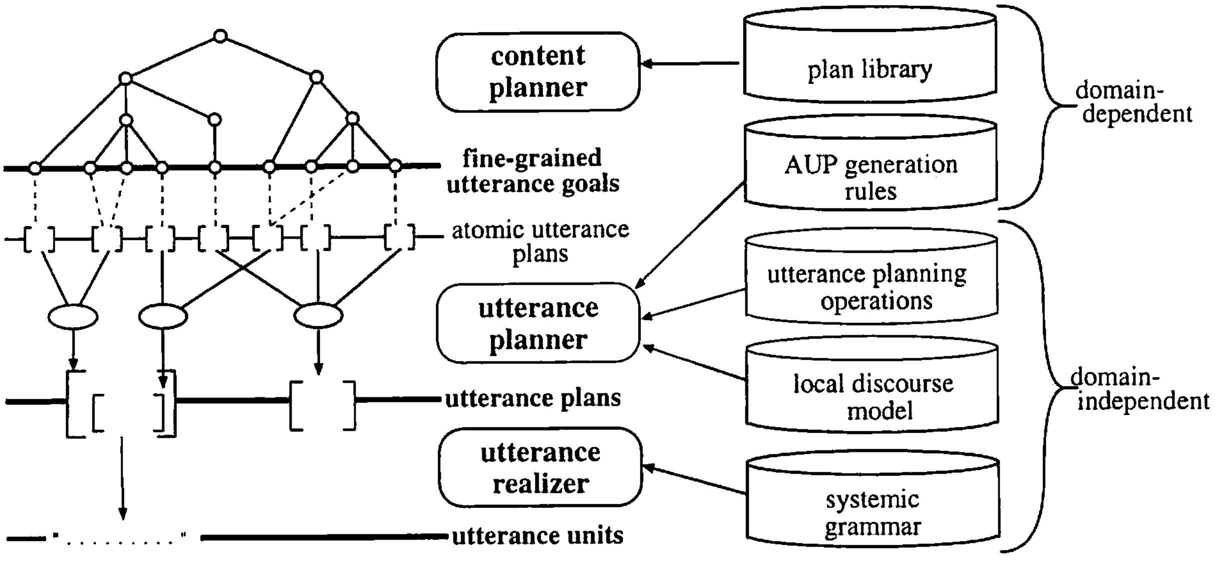

Figure 3: Three-layered incremental utterance generation model

cus shift goal has been successfully attained or not, prompting the user's feedback. This goal is of the intentional type. These three aspects, i.e. the informational, interpersonal, and presentational aspects, of dialogue correspond to what Halliday calls the metafunctions of language: ideational, interpersonal, textual. When we try to decompose the intention of each utterance unit into FUGs, these three different perspectives are expected to render helpful guidelines.

Third, in terms of the informational aspect, one of the major issues is the generation of referring expressions. For example, the explanation provider P in the dialogue in Figure 2 attempts to refer to a particular station by describing its attributes, such as its name, line, etc. As claimed, for example, by Appelt [1], Dale [6], and Heeman [12], which attributes should be used to refer to an instance should strongly correlate to those attributes which can effectively and efficiently discriminate that instance from the other instances in the domain. The user model would also be employed in this decision. Thus, the choice of the content of a referring expression should be entrusted to the content planner, while entrusting the choices of anaphoric expressions to the utterance planner. This division of labor requires that FUGs can fully represent the specification of each referring action.

Fourth, the system is sometimes required to suspend an utterance, attaching a feedback prompt at its end, in order to confirm the attainment of the intended goals. Since the choice of linguistic means of prompting the recipient's feedback requires linguistic knowledge, it should not be a matter for content planning. Thus, it is desirable that the content planner can communicate with the utterance planner, and request the generation of a feedback prompt through an abstract utterance goal such as g_confirm(*fug*) for a certain goal *fug*.

The following are example FUG types, which we implemented for the domain of the NTT-ISL route navigation corpus:

- informational FUGs:
  - refer to an instance:
    g_refer(*instance*)
  - describe the class or an attribute of an instance:
    g_type(*instance*, *class*)
    g_attrib(*instance*, *attribute*, *value*)
  - describe the class or an attribute of an instance to achieve a referring goal (*FUG*):
    g_ref_type(*fug*, *instance*, *class*)
    g_ref_attr(*fug*, *instance*, *attribute*, *value*)
- interpersonal FUGs:
  - confirm the attainment of a FUG:
    g_confirm(*fug*)
  - describe the intentional relation between FUGs:
    g_subgoal(fug_1, *relation-type*, fug_2)
  - express the speaker's attitude to the content described or referred to by an informational fug, e.g.:

g_inform(*informational-fug*)
g_request(*informational-fug*)

- inform whether the speaker assumes that the hearer knows the instance or not:

  g_hearer_knowledge(*instance*, *status*)

  where *status* is, for example, *known*, *unknown*, *inferable*, etc.

- presentational FUGs:
  - attain fug_1 before fug_2:

    g_order(fug_1,fug_2)
  - shift the discourse focus to an instance:

    g_focus(*instance*)
  - open/close a discourse segment:

    g_open_explanation_exchange
    g_move_to_next_transaction

Figure 4 shows an example set of FUGs, from which utterance units P1, P2, and P3 in Figure 2 are generated.

5 Utterance Planning

In this section, we first mention AUP generation rules, and then move to the aggregation process.

5.1 Mapping from FUGs to AUPs

Each AUP generation rule is defined as a production rule of the form:

$$\begin{array}{c}\{G_1\!:\!FUG_1, G_2\!:\!FUG_2, \dots\}\\ (Cond_1, Cond_2, \dots)\end{array} \Rightarrow \begin{array}{c}\{AUP_1, AUP_2, \dots\}\\ (Constr_1, Constr_2, \dots)\end{array}$$

where the left hand side consists of the source FUGs (G_i : FUG_i, where G_i is the identifier of FUG_i) and the applicability conditions ($Cond_j$) of the rule, whereas the right hand side consists of the target AUPs and the additional constraints ($Constr_i$) on those AUPs (*UP constraints*).

We represent AUPs (and UPs) as typed feature structures, and use the unification operation to merge AUPs in generating a whole UP. This unification-based processing facilitates the aggregation operation in the following respects. First, the monotonicity of unification allows us to describe AUP generation rules purely declaratively, which facilitates development and maintenance. Second, as we mention in Section 5.2, since the ontology of AUPs is defined in a systemic fashion, the unifiability of AUPs guarantees the realizability of the resultant UP.

The following is an example of a simple AUP generation rule, which represents the mapping from

```
               % informational FUGs
  ref1: g_refer(id1)
  rtp1: g_ref_type(ref1,id1,identifying)
 rat11: g_ref_attr(ref1,id1,identified,cs1)
 rat12: g_ref_attr(ref1,id1,identifier,st1)
  ref2: g_refer(cs1)
  rtp2: g_ref_type(ref2,cs1,closest_station)
  ref3: g_refer(st1)
  rtp3: g_ref_type(ref3,st1,station)
 rat31: g_ref_attr(ref3,st1,name,'Hatsudai')
 rat32: g_ref_attr(ref3,st1,line,ln1)
 rat33: g_ref_attr(ref3,st1,location,loc1)
  ref4: g_refer(ln1)
  rtp4: g_ref_type(ref4,ln1,train_line)
  rat4: g_ref_attr(ref4,ln1,name,'Keiō')
  ref5: g_refer(loc1)
  rtp5: g_ref_type(ref5,loc1,station_location)
 rat51: g_ref_attr(ref5,loc1,distance,3)
 rat52: g_ref_attr(ref5,loc1,cardinal_pt,st2)
  ref6: g_refer(st2)
  rtp6: g_ref_type(ref6,st2,station)
  rat6: g_ref_attr(ref6,st2,name,'Shinjuku')

               % interpersonal FUGs
  sub1: g_subgoal(opn1,subgoal,foc1)
  cnf1: g_confirm(foc1)
  cnf2: g_confirm(inf1)
  inf1: g_inform(id1)
  hkg1: g_hearer_knowledge(cs1,inferable)
  hkg2: g_hearer_knowledge(st1,unknown)
  hkg3: g_hearer_knowledge(ln1,known)
  hkg4: g_hearer_knowledge(st2,known)

               % presentational FUGs
  opn1: g_open_explanation_exchange
  foc1: g_focus(id1)
  ord1: g_order(opn1,inf1)
  ord2: g_order(rat32,rat33)
  ord3: g_order(rat51,rat52)
```

Figure 4: An example set of FUGs, from which utterance units P1, P2, and P3 in Figure 2 are generated

domain-dependent concept **station** to linguistic typed feature station, and is applicable to goal **ref1** in Figure 4:

$$\{G\text{:}\mathtt{g\_ref\_type(\_,}X\mathtt{,station)}\}\,\langle\rangle \Rightarrow \left\{ \begin{array}{l} \boxed{1}\begin{bmatrix}\text{inst:}\boxed{3}X \\ \text{sem:}\begin{bmatrix}\text{nominal\_group}\\ \text{thing:}\boxed{4}\text{station}\end{bmatrix}\end{bmatrix}, \\ \boxed{2}\begin{bmatrix}\text{inst:}\boxed{3}\\ \text{sem:}\boxed{4}\end{bmatrix}\end{array}\right\}\langle\rangle \quad (1)$$

Symbols beginning with capital letters denote logical variables. Similarly, the mapping from a semantic role in the domain model to a linguistic semantic role can be described as follows:

$$\{G\text{:}\mathtt{g\_ref\_attr(\_,}X\mathtt{,identified,}Y\mathtt{)}\}\,\langle\rangle \Rightarrow \left\{ \begin{array}{l} \boxed{1}\begin{bmatrix}\text{inst:}X \\ \text{sem:}\begin{bmatrix}\text{clause}\\ \text{identified:}\boxed{3}\end{bmatrix}\end{bmatrix}, \\ \boxed{2}\begin{bmatrix}\text{inst:}Y\\ \text{sem:}\boxed{3}\end{bmatrix}\end{array}\right\}\langle\rangle \quad (2)$$

More than one FUG can appear in the left hand side of a rule. The next example is applicable to the set of goals **ref3**, **art31**, and **hkg2**:

$$\left\{\begin{array}{l} G1\text{:}\mathtt{g\_ref\_attr(G3,}X\mathtt{,name,}Y\mathtt{)}, \\ G2\text{:}\mathtt{g\_hearer\_knowledge(}X\mathtt{,unknown)}\end{array}\right\} \Rightarrow$$
$$\langle G3\text{:}\mathtt{g\_refer(}X\mathtt{)}\rangle \left\{ \begin{array}{l} \boxed{1}\begin{bmatrix}\text{inst:}X \\ \text{sem:}\begin{bmatrix}\text{nominal\_group}\\ \text{identifier:}\boxed{4}\end{bmatrix}\end{bmatrix}, \\ \boxed{2}\begin{bmatrix}\text{inst:}\boxed{3}Y\\ \text{sem:}\boxed{4}\begin{bmatrix}\text{proper\_name}\\ \text{lex:}\boxed{3}\end{bmatrix}\end{bmatrix}\end{array}\right\}\langle\rangle \quad (3)$$

UP constraints are used to describe the constraints that cannot be represented by pure typed feature structures. For example, a confirmation goal may be mapped to the constraint that would suspend the utterance at a certain point to prompt the user's feedback as follows:

$$\{G1\text{:}\mathtt{g\_confirm(}G2\mathtt{)}\}\,\langle \mathrm{G2}\text{:}\mathtt{g\_focus(}X\mathtt{)}\rangle \Rightarrow$$
$$\left\{ \begin{array}{l} \boxed{1}\begin{bmatrix}\text{sem:}\begin{bmatrix}\text{clause}\\ \text{theme:}\boxed{3}\end{bmatrix}\end{bmatrix}, \\ \boxed{2}\begin{bmatrix}\text{inst:}X\\ \text{sem:}\boxed{3}\end{bmatrix}\end{array}\right\} \quad (4)$$
$$\langle\text{c\_suspend\_to\_confirm}(\boxed{2})\rangle$$

c_suspend_to_confirm(aup) denotes the constraint that a feedback prompt should be placed immediately after the execution of aup. Applying this rule to goal **foc1** in Figure 4 would generate an utterance like "*moyorieki-wa* (the closest station is)" (utterance P1 in Figure 2) with a particular prosodic pattern to express a feedback prompt. As for other types of UP constraint, we implemented phrase ordering constraints such as c_order(aup_i,aup_j), which denotes the constraint that aup_i should be executed before aup_j.

The role of AUP generation rules is to bridge the gap between the domain-dependent ontology of the domain resources and the domain-independent ontology of the linguistic resources. That is, we develop AUP generation rules depending on the domain, while maintaining the domain-independency of the ontology of the target AUPs. This facilitates the domain-independent development of the rest of the linguistic resources (see Figure 3). Since each AUP generation rule is associated with only a limited number of FUG(s), this task tends to be rather simple, which facilitates the development and maintenance of self-contained AUP generation rules. Furthermore, this simplicity will not diminish the diversity of utterance plans that the system can generate, because that diversity is expected to emerge from diverse choices in applying AUP generation rules and combining AUPs, which are both executed throughout the aggregation process described below.

5.2 Aggregation of AUPs

As mentioned above, we use the typed unification operation to aggregate AUPs to generate complete utterance plans. For example, applying rules (1) and (2) described above to FUGs **ref3**, **rtp3**, **rat31**, and **hkg2** would generate the following four AUPs:

$$\left\{ \begin{array}{l} \boxed{1}\begin{bmatrix}\text{inst:}\boxed{5}\mathtt{st1} \\ \text{sem:}\begin{bmatrix}\text{nominal\_group}\\ \text{thing:}\boxed{6}\end{bmatrix}\end{bmatrix}, \\ \boxed{2}\begin{bmatrix}\text{inst:}\boxed{5}\\ \text{sem:}\boxed{6}\text{station}\end{bmatrix}, \\ \boxed{3}\begin{bmatrix}\text{inst:}\mathtt{st1} \\ \text{sem:}\begin{bmatrix}\text{nominal\_group}\\ \text{identifier:}\boxed{8}\end{bmatrix}\end{bmatrix}, \\ \boxed{4}\begin{bmatrix}\text{inst:}\boxed{7}\mathtt{'Hatsudai'}\\ \text{sem:}\boxed{8}\begin{bmatrix}\text{proper\_name}\\ \text{lex:}\boxed{7}\end{bmatrix}\end{bmatrix}\end{array}\right\}$$

Among these AUPs, $\boxed{1}$ and $\boxed{3}$ can be unified, in generating:

$$\boxed{1}\boxed{3}\begin{bmatrix}\text{inst:}\boxed{5}\mathtt{st1} \\ \text{sem:}\begin{bmatrix}\text{nominal\_group}\\ \text{thing:}\boxed{6}\text{station}\\ \text{identifier:}\boxed{8}\end{bmatrix}\end{bmatrix}$$

The nondeterminism of the utterance planning process arises from the choices in both applying AUP generation rules and combining AUPs. These choices are guided by the five types of general factors enumerated in Section 1. Each factor is taken into account in the utterance planner as follows.

Efficiency The efficiency of communication is an important factor that strongly influences the choices in utterance planning. One of the simplest criteria of communication efficiency is the number of FUGs each utterance unit attains. In order for an utterance unit to attain as many FUGs as possible, the system prefers:

- AUP generation rules that consume larger numbers of FUGs,
- AUP generation rules whose target AUPs can be merged with other AUPs, and
- combinations of AUPs that generate smaller numbers of UPs.

Since searching the whole search space for the globally optimal decision is prohibitively expensive, the current system makes choices according to local preference. For example, AUP generation rules are roughly ordered in terms of the static preference.

Conciseness The comprehensibility of each utterance unit should also be taken into account. That is, each utterance unit should be concise enough to be easily understood by the hearer. For example, in the dialogue in Figure 2, P3 was uttered separately from P2 although they could have been merged into a single utterance unit using a relative clause. This can be explained according to this conciseness preference. At present, we consider only the structural complexity of each utterance unit, by prohibiting aggregation operations that would produce any embedded structure in a UP whose depth is greater than a certain threshold.

Feedback prompting As illustrated in AUP generation rule (3), a confirmation goal g_confirm(*fug*) is normally mapped to UP constraint c_suspend_to_confirm(*aup*). Given a UP with one or more of this type of constraint(s), the utterance realizer first realizes the utterance corresponding to the whole UP, and then articulates only its first subpart delimited by the suspending constraint(s). For example, the following is the UP corresponding to utterances P1 and P2 of the dialogue in Figure 2:

$$\left\{\begin{array}{l}
\boxed{1}\left[\text{sem: }\begin{array}{l}\text{connective}\\ \&\ \text{opening\_exchange}\end{array}\right],\\
\boxed{2}\left[\begin{array}{ll}\text{inst:} & \texttt{id1}\\ \text{sem:} & \left[\begin{array}{l}\text{declarative}\\ \&\ \text{identifying}\\ \&\ \text{continuing\_clause}\\ \text{identified:}\,\boxed{3}\\ \text{identifier:}\,\boxed{4}\\ \text{theme:}\,\boxed{3}\end{array}\right]\end{array}\right],\\
\boxed{3}\left[\begin{array}{ll}\text{inst:} & \texttt{cs1}\\ \text{sem:} & \left[\begin{array}{l}\text{nominal\_group}\\ \&\ \text{continuing\_PP}\\ \text{thing: closest\_station}\end{array}\right]\end{array}\right],\\
\boxed{4}\left[\begin{array}{ll}\text{inst:} & \texttt{st1}\\ \text{sem:} & \left[\begin{array}{l}\text{nominal\_group}\\ \text{thing: station}\\ \text{identifier:}\,\boxed{5}\end{array}\right]\end{array}\right],\\
\boxed{5}\left[\begin{array}{ll}\text{inst:} & \boxed{6}\ \texttt{'Hatsudai'}\\ \text{sem:} & \boxed{5}\left[\begin{array}{l}\text{proper\_name}\\ \text{lex:}\,\boxed{6}\end{array}\right]\end{array}\right]
\end{array}\right\}$$

$$\left\langle\begin{array}{l}
\text{c\_suspend\_tp\_simplify}(\boxed{2}),\\
\text{c\_suspend\_to\_confirm}(\boxed{3}),\\
\text{c\_order}(\boxed{1},\boxed{2})
\end{array}\right\rangle$$

Receiving this UP, the utterance realizer would first generate the surface structure of the utterance corresponding to it, and then articulate only the subconstituent realizing subplan $\boxed{1}$ and $\boxed{3}$ due to UP constraint c_suspend_to_confirm($\boxed{3}$). In this example, $\boxed{4}$ does not precede $\boxed{3}$ since $\boxed{3}$ functions as the theme role of clause $\boxed{2}$ as well as the identified role, and the lexico-grammatical constraints require a theme role to be placed at the beginning of the clause. It should be here noted that the utterance realizer receives the whole UP, which specifies the context of the subconstituents to articulate. This is based on our intuition that human interlocutors seem to prepare for the subsequent utterances to a certain extent when they start to articulate the current utterance unit.

Realizability We also take the realizability constraint into account. Since arbitrary aggregation of AUPs might produce a linguistically ill-formed UP, the utterance planner is required to check the linguistic realizability of each resultant UP. However, it is obvious that a naive generate-and-test methodology would make the whole process prohibitively expensive. We need some efficient mechanism to examine the realizability of any given UPs. To solve this problem, we define the ontology of AUPs in a systemic fashion, i.e. we typologize the typed features of AUPs (and UPs) according to system networks [11], which are shared with the utterance realizer. The systemic typology is linguistically motivated, and explicitly describes the linguistic inconsistency between features. For example, a confirmation goal might be mapped through the following rule, which is different from rule (3) described in Section 5.1:

$$\{G\mathtt{:g\_confirm}(G1)\}\,\langle \mathrm{G1}\mathtt{:g\_focus}(X)\rangle \Rightarrow$$
$$\left\{\begin{array}{l} \boxed{1}\left[\begin{array}{ll} \mathsf{inst}: & \boxed{3}\,X \\ \mathsf{sem}: & \left[\begin{array}{l} \mathsf{declarative} \\ \&\ \mathsf{pointing} \\ \&\ \mathsf{continuing\text{-}type} \\ \ \mathsf{goal}:\boxed{4} \end{array}\right] \end{array}\right], \\ \boxed{2}\left[\begin{array}{ll} \mathsf{inst}: & \boxed{3} \\ \mathsf{sem}: & \boxed{4} \end{array}\right] \end{array}\right\} \qquad (5)$$
$$\langle \mathsf{c\_cut}(\boxed{1})\rangle$$

Applying this rule (5) to FUG `foc1` in Figure 4, in place of rule (3), would generate the following constituent:

$$\boxed{1}\left[\begin{array}{ll} \mathsf{inst}: & \boxed{3}\,\mathsf{cs1} \\ \mathsf{sem}: & \left[\begin{array}{l} \mathsf{declarative} \\ \&\ \mathsf{pointing} \\ \&\ \mathsf{continuing\_clause} \\ \ \mathsf{goal}:\boxed{4} \end{array}\right] \end{array}\right]$$

which would be mapped to an utterance like "*mazu moyorieki na-n-desu-ga* (first, [let us start with] the closest station)". However, unlike the case shown in (5), since constituent $\boxed{1}$ is of the declarative type, and thus of the clause type (declarative is a subtype of clause), constituent $\boxed{1}$ cannot be unified with the filler of the identified role of the identifying process due to the type constraint that the constituent in the identified role of an identifying process must be of the postpositional_phrase type. With such a typology, the unifiability of AUPs guarantees the realizability of the merged UP, which prevents the system from merging linguistically inconsistent AUPs. As claimed by Henschel [13], such a systemic typology can be straightforwardly represented (implemented) if one employs a typed unification systems like CUF.

Cohesiveness In terms of cohesive devices for fine-grained interaction, we currently consider the following five classes:

- *Syntactic connections*: Even though each of the provider's utterances appears to be fragmental, it is often the case that a sequence of utterance units would constitute one or more clause(s). For example, in the dialogue in Figure 1, the phrases distributed over utterance units P1, P2, and P3, would constitute a clause like "*de* (and) *soko-kara* (from there) *mata* (again) *hidari-ni* (to the left) *massugu* (straight) *gorira-no atari-made* (to the gorilla) *it-te-kudasai* (please move!)". In other words, the cohesiveness between these utterance units is maintained by means of syntactic head-complement relations, which are typically expressed, for example, by case markers attached to the complements. In our model, such kinds of connections can be easily maintained since the system generates a whole clause first, and then articulates only a part of it according to suspending constraints.

- *Common anaphoric expressions*: At present, the system employs a simple focusing model to control the generation of anaphoric expressions such as pronouns, definite noun phrases, etc. The utterance planner is responsible for this task.

- *Connective expressions*: In the current implementation, connectives such as "*mazu* (first)", "*de* (and then)", etc. are, in most cases, derived from presentational FUGs associated with discourse segments like g_open_explanation_exchange. The utterance planner is responsible for this task.

- *Utterance ending types*: Based on our corpus analysis, the provider's utterances can be classified into two types, according to their ending patterns: the continuing and closing types. A continuing unit indicates that the provider will convey more information about the current topic, while a closing unit indicates that the provider means to close the current discourse segment. The recipient can tell which type a given unit belongs to by analyzing its linguistic form and prosodic pattern; for example, "*migi-ni* ([move] to the left)" and "*ikun-desu-ga* ([you should] move)" (continuing unit marker) are of the continuing type, while "*migi-desu* ([the direction is] to the left)" and "*it-te-kudasai* (Please move!)" are of the closing type. In our model, the utterance planner assigns either ending type to each utterance unit. A type of continuing_clause, as appears in rule (5), illustrates a typical means for this assignment task.

- *Anaphoric presentations*: In fine-grained interaction, a sequence of phrases that would constitute a clause is likely to be interrupted by subdialogues. In such a case, that clause is often repeated after the completion of those subdialogues, which makes salient the relationships between the individual phrases of that phrase. Such repeated parts can be considered to function as a sort of "reference" to other parts. Thus, we refer to such utterance unit parts as anaphoric presentations. Although anaphoric presentations make a significant contribution to the maintenance of cohesiveness, we have not fully implemented the mechanism for producing them yet. The task of generating anaphoric presentations is not as simple as it appears since, in most cases, an anaphoric presentation is supposed to give a summary of the preceding part, rather than a simple verbatim repetition. For further discussion on anaphoric presentations, see [15].

6 Conclusion

This paper discussed the issues that need to be considered in order to realize fine-grained explanatory utterance generation, particularly focusing on the task of utterance planning. including utterance delimitation and utterance organization. For this purpose, we presented an enhanced three-layered generation model. Our model is distinct from previous utterance generation models in the following respects. First, the content planner communicates with the realization component through a set of utterance goals that are more fine-grained than any conventional speech act goals. This facilitates fine-grained dialogue management on the content planning side, and the generation of fine-grained utterances on the realization side. Second, in the realization component, the utterance planner performs utterance delimitation and organization by aggregating fragmental utterance plans, by way of which flexible combination between informational, interpersonal, and presentational utterance goals are facilitated. Third, such aggregation operations are executed in a unification-based framework that employs a linguistically well-motivated semantic feature typology, allowing the utterance planner to aggregate fragmental utterance plans without consulting lexico-grammatical resources. This facilitates the control of the aggregation process, as well as the maintenance of the modularity of the utterance realizer.

A number of issues still remain to be explored. For utterance planning, we first need to further refine the mechanism for the generation of cohesive devices. We also need to improve the mechanism for controlling the aggregation process. In the current implementation, the preference over aggregation operations is largely encoded in the procedural part of the system except the static preference assigned to AUP generation rules. To facilitate the refinement and maintenance of those preferences, we need a mechanism that enables us to describe them more declaratively. As for content planning, we need to explore how the system should adapt the current FUG set when it receives negative user feedback on a certain fine-grained level. Furthermore, it is still an open issue how the content planner and the utterance planner should interact, although we currently assume that these two modules ideally work in parallel.

Acknowledgements

The authors would like to thank the NTT Information and Communication Systems Labs and the MapTask dialogue project group of Chiba University for allowing us to use their dialogue corpora. They would also like to thank Dr. KATO Tsuneaki (NTT Information and Communication Systems Labs) for his suggestive comments and Mr. Timothy Baldwin for his help in writing this paper.

References

[1] D. E. Appelt. *Planning English Sentences*. Cambridge University Press, 1985.

[2] J. Bateman. KPML: the KOMET-Penman multilingual linguistic resource development environment. In *Proceedings of the Fifth European Workshop on Natural Language Generation*, 1995.

[3] J. Carletta. *Risk-taking and recovery in task-oriented dialogue*. Ph.D. thesis, University of Edinburgh, 1992.

[4] A. Cawsey. *Explanation and interaction*. The MIT Press, 1992.

[5] P. Clancy. Written and spoken style in Japanese narratives. In T. Deborah, editor, *Spoken and Written Language*, pp. 55–76. 1982.

[6] R. Dale. *Generating Referring Expressions*. The MIT Press, 1992.

[7] K. de Smedt and G. Kempen. Incremental sentence production, self-correction and coordination. In G. Kempen, editor, *Natural Language Generation*, chapter 23, pp. 365-376. Martinus Nijhoff, 1987.

[8] K. de Smedt and G. Kempen. Segment grammar: a formalism for incremental sentence generation. In Paris, et al. [19], chapter 13, pp. 329–349.

[9] K. Dohsaka and A. Shimazu. A computational model of incremental utterance production in task-oriented dialogues. In *Proceedings of the 16th International Conference on Computational Linguistics*, pp. 304–309, 1996.

[10] J. Dörre, M. Dorna, and J. Junger. The CUF user's manual. *Institute für maschinelle Sparchverarbeitung (IMS), Universität Stuttgart*, 1996.

[11] M. A. K. Halliday. *An Introduction to Functional Grammar*. Edward Arnold, 1994.

[12] P. A. Heeman and G. Hirst. Collaborating of referring expressions. *Computational Linguistics*, Vol. 21, No. 3, 1995.

[13] R. Henschel. Traversing the labyrinth of feature logics for a declarative implementation of large scale systemic grammars. In *Proceedings of the CLNLP'95*, 1995.

[14] E. H. Hovy and L. Wanner. Managing sentence planning requirements. In *Proceedings of ECAI'96 Workshop: Gap and Bridge: New Dirrections in Planning and Natural Language Generation*, 1996.

[15] K. Inui, A. Sugiyama, T. Tokunaga, and H. Tanaka. Fine-grained incremental and interactive elaboration in explanatory dialogue. In *Proceedings of ECAI'96 Workshop: Gap and Bridge: New Dirrections in Planning and Natural Language Generation*, 1996.

[16] H. Koiso, Y. Horiuchi, S. Tsuchiya, and K. Ichikawa. The accoustic properties of "subutterance units" and their relevance to the corresponding follow-up interjections in Japanese. 1995.

[17] J. D. Moore. *Participating in Explanatory Dialogues.* MIT Press, 1995.

[18] Y. Nakano (Ishikawa). Communicative mode dependent contribution from the recipient in information providing dialogue. In *Proceedings of the International Conference on Spoken Language Processing*, pp. 959–962, 1994.

[19] C. L. Paris, W. R. Swartout, and W. C. Mann, editors. *Natural Language Generation in Artificial Intelligence and Computational Linguistics.* Kluwer Academic Publishers, 1991.

[20] S. Prevost. An information structural approach to spoken language generation. In *Proceedings of the 34th Annual Meeting of the Association for Computational Linguistics*, 1996.

[21] O. Rambow and T. Korelsky. Applied text generation. In *Proceedings of the Conference on Applied Natural Language Processing*, pp. 40–47, 1992.

[22] N. Reithinger. POPEL — a parallel and incremental natural language generation system. In Paris, et al. [19], chapter 7, pp. 179–199.