The Knowledge Engineering Review, Vol. 8:4, 1993, 329–347

Natural language generation in critiquing

IVAN RANKIN

Department of Computer and Information Science, Linköping University, 581 83 Linköping, Sweden (email: ira@ida.liu.se)

1 Introduction

A central area of application for knowledge-based systems is for giving consultative advice to the user. Such systems engage the user in a dialogue in the process of collecting enough information to be able to infer a conclusion from the knowledge base. Traditionally, then, the main initiative in the consultation process has been allocated to the system

Expert critiquing systems (ECS) present an important complement to such systems. Expert critiquing is based on the idea of supporting the user's own decision-making rather than directly suggesting a recommendation. It is assumed that the user initially proposes a tentative course of action after which the system reviews the proposals relative to known circumstances and provides feedback, for example confirming and supporting correct choices, presenting corrections to user misconceptions or commenting on viable alternatives.

Critiquing is the process of evaluating a solution to a task and providing an appraisal of it to contribute to possible improvements. The main distinguishing feature of critiquing systems is that they are designed to assist the user by reasoning about and evaluating user-given *solutions* to problems rather than presenting solutions to user-given *problems*. This is of particular importance when one particular solution may not always be the best. In many situations a number of different users confronted with a similar problem may not have the same resources to deal with it, e.g., access to the most modern equipment in the biochemical laboratories, or they may as a result of personal experience or preferences tackle the problem in a variety of ways. Thus the most pertinent solution may vary accordingly. The aim, then, of the ECS is to provide the user with expert feedback on the solution he subscribes to, commenting on advantages and drawbacks or possible risks attached to the chosen solution, rather than to propose a course of action that may not always be appropriate for each individual user. Another way of looking at this is that in critiquing the user is actively engaged in problem-solving activity allowing for variety and subjectivity, while at the same time having access to expert feedback.

In this way an ECS is viewed as an assistant to the problem solver (user) accepting the current solution as the most appropriate and point out its strengths and weaknesses. The role of an ECS can be extended to provide appropriate alternative solutions to the one initially suggested, thereby endowing the ECS with the role of a cooperative partner. If the ECS can also take precautionary measures to prevent possible future mistakes by the user (in an incremental problem-solving environment), its role is tending towards instruction rather than just reaction.

Critiquing has been applied in various domains such as medicine, biochemistry and design and while no common methodology has yet emerged, a distinction has been made between two main approaches to critiquing, differential and analytic (Fischer, 1991). In differential critiquing the ECS generates its own solution to the problem and then compares this to the user's suggestions, noting agreements and differences in the two solutions. This is most useful for well-structured problems where solutions can be evaluated according to objective principles, i.e., where one solution can be judged as being better than another in some respect, such as when a proposed solution is less time- or resource-consuming. Analytic critiquing is applied to less structured problems where a solution rests more on subjective principles or guidelines, e.g., in interface design where there is, say, no hard and fast rule as to how many items can be included in a pop-up menu or in traffic regulation, where a roundabout or traffic-lights may be equally effective at a road junction.

Analytic and differential approaches to critiquing are not mutually exclusive, but rather complementary, and both can be used in one and the same system. For example, when administering anaesthetics any choice of drugs involves some risk and there is no single best drug in any situation and thus analytic critiquing can be employed. At the same time, anaesthetizing the patient is just one step in the treatment plan and other steps may have been erroneously omitted by the user of the system and these are picked up by using differential critiquing, i.e., by comparing the user's overall plan with a standard treatment plan containing the essential steps to be included.

The critique can be presented in various ways. A standard means of presenting the critique has been to use parameterized pre-stored texts, sometimes enhanced with visual stimuli such as colour (texts in red frames are warnings, texts in green frames are confirmations of correct decisions) or by disallowing the user to move objects on the screen to certain forbidden positions. Little attention has been paid to text generation techniques, however, as the emphasis in most projects of this kind has been on developing critiquing methods. An early exception to this can be found in Miller's (1988) work, where a text generation module based on Augmented Transition Network techniques was used (see below). Over the last decade, however, there has been a surge of interest in natural language generation techniques enabling the construction of paragraph-length texts. In this paper, we review this development and show how these advances can contribute to the generation of multi-sentence critiques.

In section 2 we provide a brief review of those aspects of text generation which are relevant to the present work. In section 3 we show how these, together with results from other areas such as argumentation theory, can be applied to the generation of critiques. In section 4 we outline a language-game model for the interaction between the user and the ECS.

2 Natural language generation

The aim of research into natural language generation is to provide computer systems with the ability to respond to their users in natural language. The response may be an answer to a user-given query, an explanation of the system's reasoning, a critique, etc., and can range from a single sentence to a multi-paragraph text.

2.1 Background

Language generation involves deciding what to say and how to say it, i.e., the process of transforming communicative intent into a natural language text. The communicative intent comes from the underlying program, such as tutoring system commenting on student behaviour, in reply to a user request for an explanation from an expert system or as a critique of a user-suggested plan of action. Once the intent has been established, strategies are employed to guide the selection of words, phrases and grammatical structures to achieve it. However, first we mention briefly other pertinent issues in language generation.

It is customary to view language generation in two main stages—deep generation and surface generation. Deep generation involves determining the content of the text—deciding what to include and organizing the content in an ordered manner. The second stage, surface generation, involves selecting words, phrases and syntactic structures to realize the ordered message. This division is not uncontested (e.g. Appelt, 1985, p. 8, who calls for an *integration* of knowledge sources involving world knowledge, knowledge about the hearer, goals to be satisfied, syntactic constraints), but it has enabled a clearer focus on the issues involved at each stage.

Early work on language generation concentrated on the surface generation of single sentences. Examples include work on direct translation from an underlying representation, e.g., using an augmented transition network (ATN) (Woods, 1970) to generate sentences from a semantic

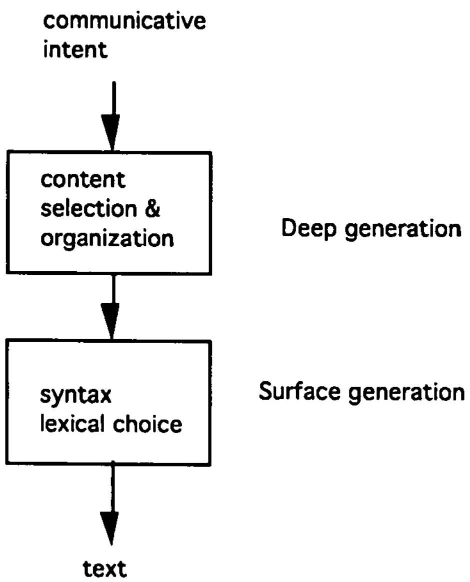

Figure 1 Deep and surface generation

network (Simmons & Slocum, 1972) and writing grammars for generating single sentences, e.g., functional unification grammar (Kay, 1979). These approaches provided alternatives to previous methods of storing canned texts or instantiating text templates (predefined structures whose slots are filled in with appropriate words and phrases), the latter not being considered as belonging to text generation proper. Current work on surface generation covers a variety of topics as presented in recent publications in the natural language generation literature (e.g. Dale et al., 1990, 1992; Zock et al., 1993), for instance generating referring expressions, lexical choice, word order, how to realize syntactically information about topic/comment and focus/background, variations in syntactic structures when translating from one language to another, the expressibility of various types of grammar—unification-based grammars, systemic grammars and head-driven grammars, the possibility of bi-directional grammars (i.e., using one grammar for parsing as well as generating), etc. While these issues of surface generation are still being hotly pursued, the last decade has also witnessed a rapidly growing interest in generating longer passages of text, which requires knowledge of how each unit in the text is related to other units and to the text as a whole, how the units are ordered and, in fact, what the units are. These and other issues mentioned below belong to the deep generation phase.

As systems have become more sophisticated, the kind of responses they issue has become more ambitious than producing single sentences, for example providing explanations in knowledge-based systems, where explanations obviously cannot always be expressed in the form of a single sentence. The traditional way of approaching this problem has been to employ code-transformation techniques, for example by recognizing rules triggered in reaching a solution and premises or conclusions are relevant to a given question and then 'translating' them into natural language to produce explanations. Such attempts have been successful to a certain degree, but at the same time highlighted inadequacies. It has proved difficult to generate coherent text from a component that was designed for another purpose (in this case a rule). The reasoning process usually works in ways that are foreign to the users of the systems and attempting explanation by presenting a rule trace often means that the explanation includes information that is irrelevant (to the explainee), at the wrong level of abstraction or presented in ways alien to the thinking of the explainee.

More is required to capture the structure of a text (e.g., an explanation) to ensure its coherency and relevance to the user of a system. As pointed out by McKeown and Swartout (1988, pp. 1–51), the need to meet these requirements has given rise to a number of research areas. Under the headings of "Discourse Strategies" the aim is to model the structure of discourse based on commonly used patterns of text organization as found in naturally occurring texts, to characterize the way in which each segment of the text is related to the whole text, e.g., describing an object in terms of its generic class and its components (see below). Next, to produce an appropriate text for a particular audience it is important to have an awareness of the hearer[1], e.g., to pitch the text at a suitable level, to include as much detail as required but not more, to use information about the hearer's knowledge of the domain when presenting an argument; these, and other issues, are dealt with in "User modelling" (see, for example, Kass & Finin, 1988; Wahlster & Kobsa, 1989). Planning formalisms have been used to provide formal mechanisms for reasoning about speaker and hearer beliefs when planning a text (e.g., Appelt, 1985). Appelt's perspective is that agents have goals, for instance to get another agent to perform a (physical or linguistic) act and thus the first agent needs to construct a plan to achieve this, e.g. by constructing a request act. As previously mentioned, using components designed for purposes other than text generation as a basis for producing a text has led to limited success. What kinds of knowledge representations are adequate for text generation? What needs to be included in such representations to efficiently support the construction of a text? These issues fall under the heading of "Knowledge representation" for text generation. All of these areas are being actively pursued at present. The interested reader will find a more thorough discussion of these issues, and others, in McKeown and Swartout (1988). In section 2.2, we concentrate on the first of these—discourse structure—showing how content structure is determined and organized.

2.2 Text generation in critiquing

As far as text is concerned, most of today's critiquing systems use pre-stored texts or text template methods to issue responses from the system (see the overview in Silverman, 1992 a, 1992 b, p. 122). This has to do with the fact that the emphasis has been on determining and evaluating critiquing techniques rather than on dynamically generating a critique. Among the first critiquing systems, though, the ATTENDING family employed a 'prose generator' in the form of an ATN to generate multi-paragraph critiques of high quality (Miller, 1986). The generator, PROSENET, aimed to avoid the weaknesses of canned text and the difficulties of "sophisticated natural-language generation" (Miller, 1986, p. 114). The text is built up by traversing the network and outputting strings—words, phrases or sentences. This process is guided by expressive frames which group together information on certain domain topics. An expressive frame consists of a number of comment frames each consisting of one or more preconditions, a pointer to a node in the ATN and possibly a priority marker; when the preconditions are met, processing is started at the given starting node. Apart from the usual actions that can be carried out in an ATN, words and phrases can be output as the result of an action.

For each expressive frame it is also possible to define an ordering function, a prose generating function, to determine how the comments should be sequenced (e.g., higher priority topics before lower priority ones; introductory comments preceding the main point). Higher level functions can also add connectives such as *first*. . ., *second* . . . *and finally* in presenting a sequence of items to polish the text. However, the comments are constructed from pre-stored strings (words, phrases and even sentences). There is no notion of a grammar in PROSENET and the network has to be largely hand-crafted to ensure that the comments are internally constructed in an orderly and correct fashion. Cohesion in the text is built into the language generator in advance rather than being a product of it. Such a "technological" approach may work well in a restricted implementation, but any changes to and extensions of the critiquing system require corresponding changes in the language generator. To provide more general techniques for language production, the "sophisticated" issues have to be faced.

[1]It is common to talk about Speaker and Hearer, though in fact Writer and Reader are more appropriate in most generation systems producing text-length texts.

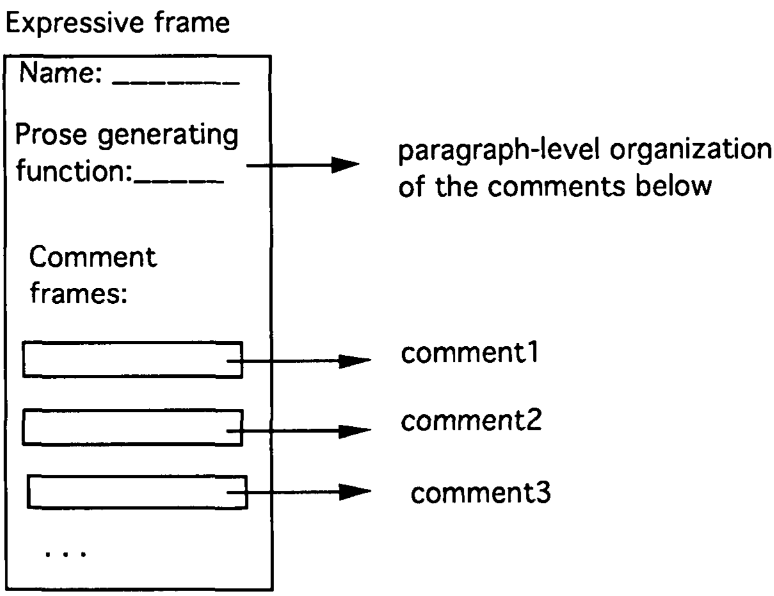

Figure 2 An expressive frame (based on Miller and Rennels, 1988, p. 40)

2.3 Discourse structure

A text is characterized by the fact that each part is related to the whole. The parts are organized in such a way that when the text is interpreted, it exhibits cohesion (Halliday & Hassan, 1976). But it has proved difficult to satisfactorily characterize the parts (clauses or propositions, say) and the relationships that hold between them. While the subject has a long history, we present some of the work most commonly referred to in recent literature on text generation.

One of the first attempts to produce paragraph-length texts was Davey's PROTEUS system, which produced descriptions of tic-tac-toe games (Davey, 1978). The texts exhibited a remarkable fluency, e.g., "The game began with your taking a corner, and I took the middle of an adjacent edge. You threatened me by taking the end of the one adjacent to the corner which you had just taken and to the one I had just taken the middle of but I blocked that . . ." (*ibid*, p. 151). The quality of the texts stems from the system's comprehensive model of the moves available as threats or counter-moves. The language generator is incorporated into the system (employing a systemic grammar) and cannot be reused in other domains. Another early proposal to tackle the problem of connected discourse was the construction of an explanation grammar (Weiner, 1980), which can be seen as a predetermined template of what to include in the text. In the domain of income tax returns, the grammar was intended to emulate naturally occurring explanation, e.g.,

State a fact
Justify the fact
Give an example to support the fact
Provide alternative facts and show their inadequacy

thereby imposing a (naturally occurring) order on the text.

The next landmark is the TEXT system (McKeown, 1985). By isolating certain standard patterns of discourse, McKeown was able to construct schemas of rhetorical predicates to produce paragraph-length texts as answers to queries about the contents of a database. A schema can be viewed as representing a strategy typically used to meet a particular communicative goal and provide the means for organizing the predicates into a coherent text. To describe an item in a database (DB), for example, the Constituency schema could be used to (a) identify the generic class of the item, (b) present its constituents, (c) add further information about the constituents, and (d) add analogical information about the item, e.g.

A ship is a water-going vehicle that travels on the surface. Its surface-going capabilities are provided by the DB attributes displacement and draft. Other DB attributes of the ship include maximum-speed, propulsion . . .

(Abbreviated example from McKeown, 1985, p. 211).

Calls to schemas could be made recursively, e.g., the constituency schema could be recalled in step (c) to describe each of the sub-components of a constituent. McKeown's schemas can be seen as flexible text templates which capture the structure of naturally occurring texts in a naval domain.

This approach proved far more versatile than previous work. The main drawback, however, is the need to devise new schemas for different domains or communicative goals. Work is still in progress in reusing schemas for different discourse purposes in order to generalize across domains (McKeown et al., 1990), and thus provide a more extensive description of coherence.

An alternative approach is to define a grammar for a particular purpose, such as Cawsey's explanation grammar (Cawsey, 1989) of which a fragment is reproduced here:

Explain → Describe, Explain Behaviour, Summarize
Explain → Type (of circuit), Summarize
Describe → Type (of circuit), Function, Describe Components
Describe → Analogy
Describe Component → Indicate Comp, Function, I/O behaviour
...

To each left-hand side of a rule a number of conditions apply as to the circumstances in which it can be triggered, e.g., the behaviour of components should be understood before these are used in an explanation, or an analogy can be used to describe a device if the user understands the analogous device (this information being available in the model of the user). Cawsey also incorporates a user-modelling facility and allows two-way interaction between system and user.

A number of other researchers have examined the relationships that hold between stretches of text. The basic notion is that texts are cohesive by virtue of the rhetorical and semantic relationships that hold between units of text. Recognizing such relationships and describing their status with respect to each other and to the function of the text is a prerequisite to using them in the generation of coherent texts. Work in this area includes rhetorical predicates (Grimes, 1975), coherence relationships (Hobbs, 1985) and more recently, rhetorical relations (Mann, 1984, Mann & Thompson, 1987, 1988). All these approaches are presented in the context of the *analysis* (rather than the generation) of natural language, e.g., Hobbs' aim is to "embed a theory of coherency relations within the larger context of a knowledge-based theory of discourse interpretation". The work by Mann & Thompson, Rhetorical Structure Theory (RST), in particular has aroused considerable interest in the natural language generation community, and has been developed in recent years as a basis for text generation (e.g., Hovy, 1988).

RST is a description of relations that hold between successive pieces of text, such as clauses or larger units. For example, a "circumstance" relation holds if one part of the text "establishes a circumstance or situation, and the other part is interpreted within or relative to that circumstance or situation" (Mann & Thompson, 1987). As an example, taken from a biochemical domain:

The pH value of the solution is less than 2.0. Anion exchange should be avoided as the protein will be destroyed irreversibly.

Here the first sentence provides the circumstance under which the second sentence is to be interpreted, i.e., the solution has high acidity, and in this circumstance the anion exchange technique is to be avoided. Other relations in RST include "cause", "condition", "justification", "motivation", "evidence", "contrast", "justify". (We return below to the relations that have been recognized and employed.)

RST relations can be described diagrammatically as in Figure 3 to represent how a text can be decomposed into units. The nuclear unit (marked by a vertical line) at a given level represents the

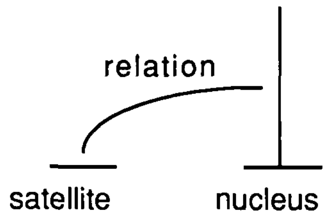

Figure 3 A generic RST schema

central item of the text at that level and the satellite(s) serve to support the nuclear component or, as Mann says, "satellite text is used to make it more likely that the nuclear text will succeed" (Mann, 1984, p. 369). There may be one or several satellites related to each nucleus, or even none at all. The order between the satellites and the nucleus is not pre-determined; the most typical ordering found in naturally occurring texts is given in Mann and Thompson (1988), and can be used as a guideline in generation; for example, a "background" or "justification" satellite is likely to precede its nucleus, while an "elaboration" or "purpose" satellite will tend to follow it.

Note that the schemas are recursive: a nucleus or a satellite may in turn be composed of another schema, thus characterizing smaller and smaller stretches of text (Figure 4). As Hovy (1990) has pointed out, the RST relations include most of Hobbs' and Grimes' relations and support McKeown's schemas.

The diagrams provide an intuitive account of an RST schema. A more rigourous definition includes the following (Mann & Thompson, 1987):

(1) Constraints on the nucleus.
(2) Constraints on the satellite.
(3) Constraints on the combination of nucleus and satellite.
(4) The effect.

By way of illustration, we reproduce the Evidence relation schema example given in Mann and Thompson (1987, p 87), where information known to the hearer (assuming a user model) is used as a satellite to support a claim (nucleus) that the user may not yet have accepted:

(1) *Constraints on the nucleus* (the claim)
The reader possibly does not believe the claim.
(2) *Constraints on the satellite* (the evidence)
The reader either already believes the satellite or will find it credible.

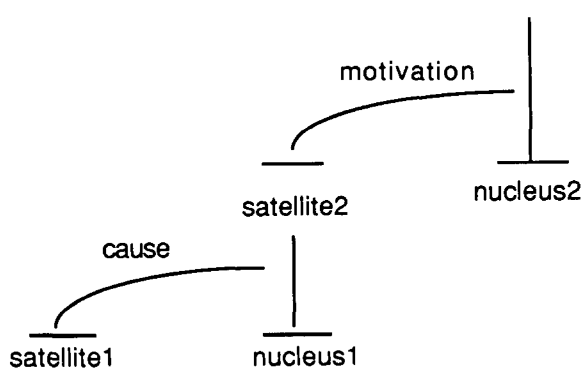

Figure 4 A recursive RST schema

(3) *Constraints on the combination of nucleus and satellite*
Comprehending the evidence will increase the reader's belief in the claim.
(4) *The effect*
The reader's belief in the claim is increased.

For example, the medicine seems to be working (nucleus/claim). The patient has shown signs of improvement (satellite/evidence).

In generation RST, schemas have been used as plans (Hovy, 1988; Moore & Swartout, 1989; Moore & Paris, 1989). Once a communicative goal has been given by an underlying application, e.g., Persuade the hearer of fact A, a search is made in a library of RST schema/plans to find one whose effects include persuading the user of a fact. Then the planner gathers the entities (usually clause-sized chunks) that meet with the constraints which hold for both the nucleus, the fact A, and the supporting satellite(s) described in the body of the plans.

Note that the constraints and the effect are expressed in terms of the knowledge state of the reader. This approach ties in neatly with work done on modelling the belief states of speaker and hearer (Cohen & Perrault, 1979; Cohen & Levesque, 1985) as described in, for example, Hovy (1988). For instance, once the user has been notified of a certain fact (claim) using the Evidence schema/plan to persuade the user that there is evidence to support the fact, it is assumed that the fact has been accepted by the user, and this can be noted in the user model. This fact can then be used, say, as justification for further facts in future interactions.

RST is a theory under development and provides a promising framework for text generation, as reflected in the amount of current work involving RST in the literature. Attempts have been made not only to refine the theory, but also to extend it, for example by using diagrams as satellites in explanatory text (André & Rist, 1990), or applying the theory to dialogue as well as monologue (Baker, forthcoming). However, a number of issues remain to be resolved and are the study of ongoing work. For example, how many relations are there? In Mann and Thompson (1988) over 20 relations are presented. They state that "We see [the list] as an open set, susceptible to extension and modification for the purposes of different genres. . ." and add that the ones presented have proved the most useful for the data examined. Hovy (1990) collected over 350 (overlapping) relations used by about 30 researchers in the field, and proposed a hierarchy of relations (later refined in Maier and Hovy (1991) of limited width but unlimited depth. The idea is to present a limited number of relations (width) which can be specified in more detail as the hierarchy is descended, (depth) e.g., "elaboration" can be specified further to "elaborate-object-description", which is also further specified to describe the function or component part of an object. This work is developed further in Maier and Hovy (1991). Dahl and Hellman (1990) argue that finding criteria for a taxonomy of rhetorical relations is not enough; they postulate two different levels of discourse structure, the propositional and the illocutionary, where relations on the propositional level (for example "cause" in p causes q) can serve as the basis for relations on the illocutionary level (p can be used as *evidence* when claiming q). In an influential paper, Gross and Sidner (1986) pointed out that discourse structure is composed of three interrelated components: the structure of the sequence of utterances, a structure of purposes, and the state of focus of attention. Moore and Pollack (1992) show that this leads to difficulties in RST, where one relation between discourse elements is primary and thus cannot account for relations at the various levels.

The question has also been raised as to the units of text the relations hold between. As seen in Figure 4, larger segments of text can act as nuclei or satellites to other segments. But what is a minimal unit? Mann and Thompson (1987) define a unit as having "independent functional integrity", which could suggest a clause or a complete proposition. In the pH example above, the first sentence provided the circumstances under which the second sentence was to be interpreted. The first sentence could also have been expressed: "At a pH value less than 2.0, anion exchange should be avoided as the protein will be destroyed irreversibly" expressing the same content, but this time with the clause replaced by a phrase (cf. the discussion on unit size in Rösner & Stede, 1992). Note that this is not just a difficulty facing RST; the same holds for other approaches: "While the need for segmentation is almost universally agreed upon, there is little consensus on what the segments of a particular discourse should be. . ." (Allen, 1987).

Hovy has also pointed out the difficulty of defining RST relations in generation systems so that they are specific enough to be useful in a particular domain and at the same time general enough to apply across domains (Hovy, 1990). This is also noted by Horacek when he states "It is an unsolved problem how the selection process determining the information to be included in a response can be described in a systematic way independent from concrete applications" (Horacek, 1992). This is a point discussed with respect to critiques in more detail in the next section.

While these and other issues are still under discussion, it is clear from the questions raised here and from the number of proposals for further developments that RST is a leading candidate for describing inter-textual relationships, an obvious requirement in text generation. We return to RST below when discussing text generation in critiquing.

3 Applying advances in text generation to critiquing

Having outlined trends in text generation and the notion of critiquing, the obvious question to ask in this context is to what extent recent advances in text generation can be utilized in critiquing generally. The aim is to avoid the difficulties of being overly domain-specific while at the same time giving the text generator enough information to contribute adequately to the production of the critique. First we provide some comments on characteristics of critiques in general, and then present some results from Argumentation Theory, which will subsequently provide a framework for dealing with some of the phenomena commonly occurring in critiquing texts.

3.1 Characteristics of critiques

As stated in the introduction, critiquing is the process of evaluating a solution to a problem and providing an appraisal of it to contribute to possible improvements. While the evaluation process may differ from one ECS to another (along the spectrum from reacting to tutoring), we state some of the most commonly occurring situations that arise in the evaluation process and then exemplify how an appraisal can be generated.

- The easiest case occurs when there is *complete agreement* between user and ECS on individual proposals or on the whole solution.
- The opposite extreme is the case of *complete disagreement* on individual proposals or the whole solution.
- The user may select a *correct strategy* but choose the *wrong means* to achieve it, for example when an infection is to be treated with antibiotics and the user has suggested an antibiotic (correctly), but suggested one that is unsuitable for the particular patient (the wrong means).
- *Superfluous proposals* may be made by the user, for instance if he suggests two steps to achieve a desired state when one suffices.
- *Omissions* may occur, i.e., essential steps in the solution are missing from the user's proposals.
- *Sub-optimal proposals* may be made, i.e., when an otherwise correct proposal can be exchanged for a better one (e.g., a less resource-consuming one).
- *Adversive interaction* may occur between proposals, as when prescribing more than one drug in the treatment of a number of symptoms.

This list is by no means exhaustive, but all critiquing systems will probably have to deal with some or most of them. In section 3.3 we show the case of complete disagreement over a given proposal between the user and the ECS. In this case, as both the user and the ECS are assumed to be competent in the domain, the interaction resembles an argument (i.e., a critical discussion, rather than a personal quarrel, in Walton's (1989, pp 3–13) terminology. As a preliminary, then, we examine a particular model taken from Argumentation Theory—the Toulmin model of argumentation (Toulmin, 1958; Toulmin et al., 1979), and show the kind of effects it will have on an excerpt of a critique.

3.2 Toulmin's model of argumentation

Toulmin's interest is in "... justificatory arguments brought forward in support of assertions" (Toulmin, 1958, p. 12), and he points out that "certain basic similarities of pattern and process can be recognized ... among justificatory arguments (*ibid*, p. 17). The model consists of the following components with the structure given:

Claim—What is being asserted.
Grounds—What grounds is the claim based on.
Warrants—What general rules or principles make the grounds relevant to the claim.
Backing—What other information can back up a particular warrant. Backing is used to demonstrate that the warrant is sound and relevant.
Modal qualifiers—Warrants may not invariably lead to the required conclusion; qualifications include "usually", "possibly", "presumably".
Possible rebuttals—If the claim is not presented as certain or necessary, it may be necessary to mention in what circumstances the claim may not hold (e.g., "unless the patient is allergic to penicillin").

Schematically the argument is represented as shown in Figure 5.

Toulmin emphasizes the interdependence of the various components of the argument: it is not sufficient for the grounds to be correct, they must also have bearing on the claim, a requirement found in the warrant, which demonstrates the applicability of the grounds to a given claim. Likewise, the backing provides support to the soundness of the warrant, and consists of the scientific facts of the domain. Furthermore, it must be made clear whether the claim is being made as a necessary conclusion, a reliable assumption, a mere possibility, or whatever, as expressed by the qualifiers (which include, for instance, "usually", "possibly", "in most cases"). Rebuttals point to special circumstances under which the claim may not hold. These are shown in the example given in Figure 6.

In this example, the claim that penicillin should be administered is supported by the fact that the patient has a straightforward urinary infection. The grounds and claim refer to a particular instance of the principle expressed by the warrant. The warrant, too, must be supported by backing. The backing consists of medical knowledge of how to treat bacterial infections (here expressed in very general terms). The model qualifier expresses what kind and to what degree reliance is to be placed on the claim. Finally, the rebuttal is introduced to express circumstances which might undermine the force of the argument, and which should be taken into consideration.

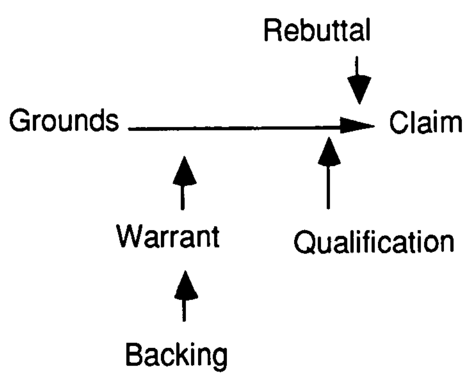

Figure 5 The structure of a justificatory argument

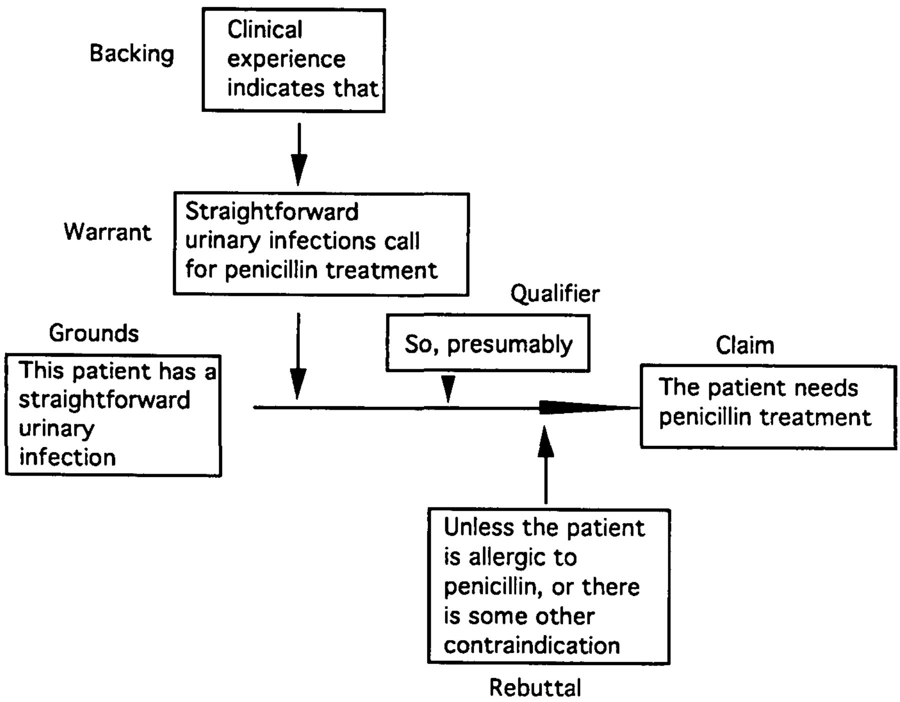

Figure 6 Toulmin's argumentation model (adapted from Toulmin et al., 1979, p. 77)

Considering the example of the model patient in Figure 6 in terms of RST, the claim corresponds to the nucleus of the text. Similarly, just as the claim is supported by grounds, the nucleus is supported by a satellite (the grounds) with a motivation relation, i.e., the grounds motivate the claim ("motivation" being one of the relations defined in RST). The backing can be viewed as the satellites supporting the warrant and consist largely of causal relations in the domain. The (possible) rebuttal can also be seen as a satellite—an "elaboration", though it is of a particular kind, noting possible exceptions to the claim being made. The qualification can be employed by those systems which reason with, say, statistical measures of the validity of the claim with respect to the given grounds.

So far we have seen how the intuitively appealing RST schema can be mapped onto an argument structure which provides guidelines as to the kind of information to select for inclusion in the critique. Before continuing we present an example of what effect this has on the critique.

3.2.1 An example of applying the argumentation model to a critique

The following extract is taken from Miller-type ECS in the domain of urinary tract infections (translated from Swedish by current author) and discussed in Rankin (1989). The first sentence deals with test to be carried out on a particular patient, the second with type and length of medication required. The user correctly suggested the first two tests mentioned, but omitted the following two (omission). The user then correctly selected a medicine, but suggested too short a treatment—3 days, when it should be 7 (correct strategy, but wrong means). The critique which was automatically generated by the PROSENET generator described in section 2.2 was based on the intuitions of the authors as far as content, level of detail and organization are concerned:

A urine culture test and routine blood status should be performed, but a urography and cystoscopy should also be carried out.

Selexid is an appropriate medicine, though the treatment should last for 7 days.

In the first sentence, the first two tests are claimed to be correct, and then there follow two more claims, i.e., that two more tests should be carried out. In the second sentence, the correct medicine is confirmed, and then the claim that the treatment should last for 7 days is made. No claim in either sentence is supported in any way. If we now add the grounds for all these decisions, the critique appears thus (grounds in plain text):

> *A urine culture test and routine blood status should be performed* as this patient has had several recent infections, *but a urography and cystoscopy should also be carried out* for the same reason.
>
> As this patient has not recently taken any antibiotics and as this patient is not sensitive to penicillin, *Selexid is an appropriate medicine, though the treatment should last for 7 days* to allow the medicine to take full effect.

(Note that the additions have been hand-written as our current work concentrates on the deep structure of the text.)

This critique is intuitively more consistent in that each claim has been motivated. Remember that the grounds and claim refer to an instance, in this case a certain patient with particular symptoms and medical history, while the warrant (next example) expresses general information, i.e., the general case. Finally, we add the warrant (in plain text), to the claims and grounds as above (both in italics):

> In the case of chronic infection certain tests should be carried out to determine the kind and intensity of the infection. *A urine culture test and routine blood status should be performed as this patient has had several recent infections, but a urography and cystoscopy should also be carried out for the same reason.*
>
> In the case of chronic infection the patient should be medicated with an appropriate medicine to relieve discomfort. *As this patient has not recently taken any antibiotics and as the patient is not sensitive to penicillin, Selexid is an appropriate medicine, though the treatment should last for 7 days to allow the medicine to take full effect.* To achieve full effect of medication antibiotics should be taken for a minimum period of 7 days.

As described above, we assume that the user of an ECS is competent in the domain and the amount of support given in this final version of the critique may appear excessive. It has been pointed out by Dahl and Hellman (1990) that even if we restrict ourselves to the claim, grounds and warrant, these three components are rarely presented together as two are usually enough to get the message across, the listener being able to infer the third. The same conclusion was reached in the presentation of a complete syllogism as an explanation (Cohen, 1987). Obviously, a model of the user would facilitate selecting an appropriate level of detail to support the claim.

When two domain experts (unaware of the argumentation model) were asked to provide lengthy, hand-written critiques, they provided many warrants as introductory information and then gave the grounds and claims as particular instances of the warrant. At the same time, the users of the critiquing system expressed a desire that the critique should deal with the particular case they were dealing with, suggesting that the claims and grounds should be emphasized and the general case (the warrant) could be used in follow-up questions. (At present this is being tested empirically.)

Furthermore, the model allows for other types of follow-up questions. If the warrant is questioned, a relevant response will be found in the backing (domain facts or principles). If the grounds is/are questioned, new arguments are constructed in the same way, but with the grounds now as a claims, i.e., the ECS claims that the grounds are valid and generates arguments to support them. Possible exceptions are captured in the rebuttal 'satellite'.

3.3 The disagreement aggregate

What we have seen so far is that using a model of argumentation allows us to address situations which commonly occur in critiquing. The next step is to use the idea of an RST schema and conflate it with the model of argumentation presented to provide general, structured units—rhetorical

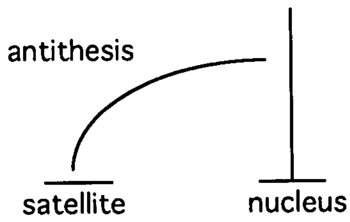

Figure 7 The antithesis schema

aggregates—which can be used across domains in the planning of response to the situations named above. We examine the case of disagreement between ECS and user. The aim of the ECS is, obviously, to argue against the user's point of view and for its own. RST provides the starting point—the antithesis schema:

Relation: Antithesis
Constraints on nucleus: Speaker has positive regard for situation presented in nucleus
Constraints on satellite: none
Constraints on nucleus + satellite combination: the situations presented in nucleus and satellites are in contrast . . . because of an incompatibility that arises from the contrast, one cannot have positive regard for both the situations presented in nucleus and satellite . . .
Effect: Hearer's positive regard for nucleus is increased
Locus of the effect: nucleus

(slightly adapted version of example given in Mann and Thompson, 1988).

Expanding the schema's nucleus and satellite gives the results in Figure 8.

The antithesis schema consists of two nuclei related to each other by the antithesis relation (as shown in Figure 8). Each nucleus is supported by its satellite(s). The positive argument (+) argues for a claim that runs counter to the user's suggestion (i.e., positive from the system's point of view); the negative argument (−) argues against the user's suggestion.

Finally, we specify in more detail the function of the nuclei and the satellites as suggested by the argumentation model: the respective nuclei express the claim being made on either side of the argument; the satellites express the grounds, etc. for making the claim.

The resulting structure—a rhetorical aggregate—specifies more closely the kind of information that needs to be gathered in meeting the requirements of the disagreement situation than the original RST schema. The communicative intent—to disagree with the user's proposal—is triggered when the situation of complete disagreement is encountered.

The aggregate described in this section shows some similarities with other work mentioned in section 3. There is an obvious RST basis to the aggregates, but our work is limited to capturing the structure of the argument. Note, however, that the argument structure was, in addition to the case

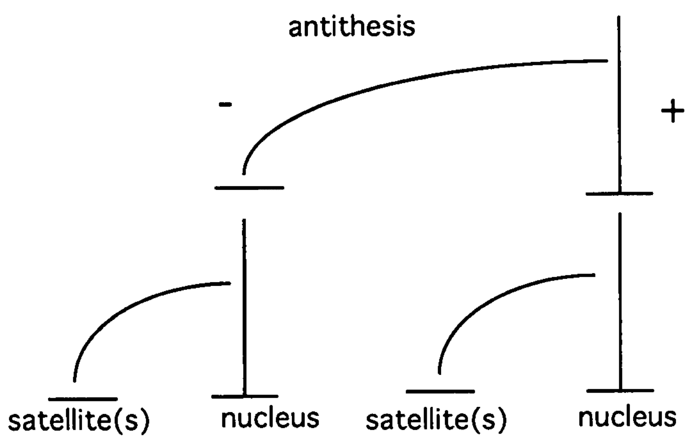

Figure 8 Antithesis schema expanded

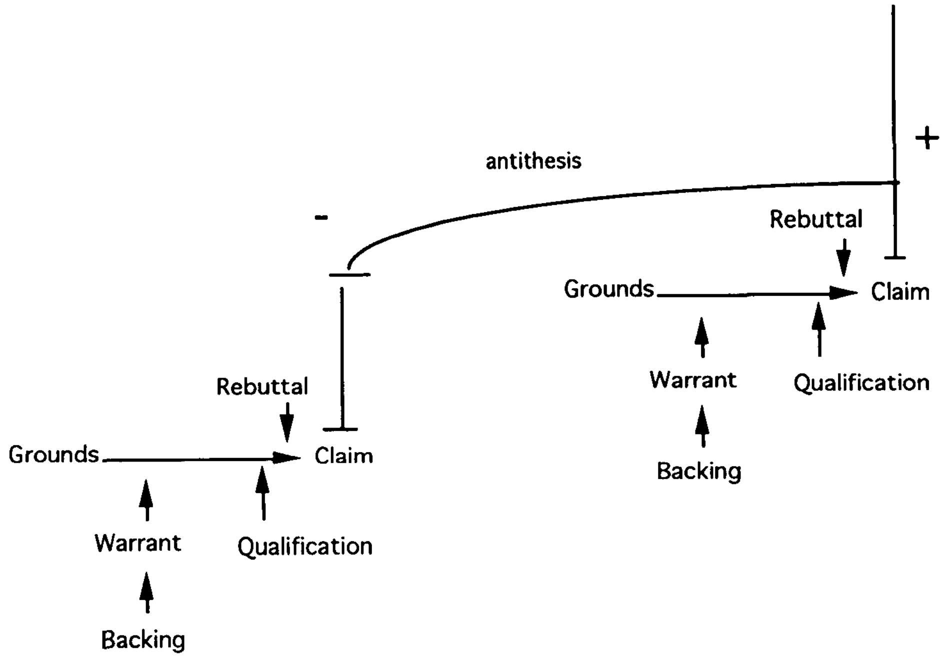

Figure 9 Antithesis rhetorical aggregate expanded to reflect the argumentation model

of disagreement, also used to generate the examples of critiques dealing with omissions and the correct strategy/wrong means case. This has to do with the fact that argumentation and explanation are closely related (the difference being that argumentation is employed when there is an explicit difference of opinion, which is not necessarily the case in explanation). Just as in argumentation, an explanation often consists of providing support for a point of view. In contrast to RST, we prefer to view the aggregates (several other are defined) in illocutionary terms, i.e., in terms of the act the ECS intends to perform by presenting the argument, whereas RST schemas are defined in terms of the effect of the act on the Hearer. Like McKeown's schemas, the aggregates have a predetermined structure, but can be used across domains.

4 Critiquing as a language game

So far the arguments produced have formed one-off responses to user proposals. The need to cater for follow-up questions has been noted by several researchers (e.g., Moore & Swartout, 1989). In this section, we suggest that a suitable model of the dialogue between user and ECS can be described in terms of a language game. By doing so we gain clarity in two mutually related issues: initiative-taking and interactional style. First, should the critiquing system respond to each user proposal as it is entered, or should it wait until a complete plan of action has been suggested, i.e., incremental or batch response? Responding incrementally can be viewed as either helpfully providing the user with information on his solution so far before he makes further proposals or, on the other hand, it may be seen as intrusive and disturb the user in the formulation of a complete plan of action. Second, to what extent should the system take the initiative when interacting with the user? For example, should the system make suggestions to guide future proposals the user might make based on those already made? There is no single correct answer to these questions; today's critiquing systems tend to implicitly use one style or another. (Silverman has provided a discussion of interactional issues in critiquing, including those mentioned here—timing and initiative, Silverman, 1991).

Making these two notions explicit will help to clarify the discussion of interactional style, and to this end we have introduced the Wittgenstein metaphor of language games in our current work.

The advantage of this is that, by viewing critiquing as a language game[2], we can define rules to cater for the style of interaction and to define the communicate goals the critique is to achieve explicitly.

4.1 Language games

Wittgenstein's proposal was that language use can be analysed in terms of (language) games. Language use is structured according to an overall format involving rules for actions performed by the participants in a given activity. There is a systematic sequencing of events. Interaction is rule-governed, and we build up expectations during the course of an interaction depending upon the kind of activity engaged in. In a particular game there are one or more actors who act according to a coherent system of normative rules with respect to a given activity.

In a given activity or game one of the actors takes the initiative. This initiative bears with it the responsibility to take into account other actor's contributions and to carry the game to an end. The latter is known as closure, where an initiative has reached a conclusion and no new initiative has been taken.

Another notion is that of reciprocity which means that the participants are aware of the on-going activity. The participants can therefore assume and refer to shared knowledge about and within the activity.

For each language game there is an overall goal. This is the intention of the participants who accepted participation in the game. An instantiation of the game above involves an intention of the participants to see the game through.

A game can be described as consisting of a number of rounds. These are typical (sequences of) actions expected by the participants, e.g., if an initiator requests information, another participant may provide him with it; if a suggestion is made, it is either agreed on or otherwise, i.e., each makes a move in a round.

In language analysis in which the term language game was first used, the aim was to determine how to explain (linguistic) phenomena in terms of the current activity. However, in language generation the aim of using the language game metaphor is to stipulate rules as to what a move or round is, how it is conducted, who can initiate and when is closure attained.

4.2 The critiquing game

Having accepted the game metaphor, it is necessary to define the various components of the game, *viz.* the goal of the game, the rules, the participants, what a round consists of, and how a game is initiated and closed.

In the critiquing game we have two participants—the initiator and the respondent. The initiator is the user who presents his solution to a problem to the critiquing system, the respondent.

A move is initiated when the user presents a proposal (or set of proposals). To complete a round (see Figure 10) the critiquing system must respond. In the case of one proposal, then the response deals with that proposal. When a complete solution is presented, a strategy for selecting the order in which the proposals are dealt with will need to be defined. One such strategy would be to deal with the proposals in the same order as the user presented them as this may reflect his way of viewing the problem. To simplify the discussion, we assume that one proposal has been made by the user and the ECS responds, as shown in Figure 10.

In move 2 of Figure 10, the first two possibilities could conclude the dialogue. If so, this corresponds to a one-off critiquing. The third possibility, requesting information, suggests two features of the critiquing game: first, the ECS is allowed to take the initiative and, second, the game has not reached a conclusion yet (an answer to the request is expected), and a 'real' dialogue is under way (i.e., not a one-off response).

[2]The notion of The Critiquing Game was first introduced by Dahl and Hellman (1993).

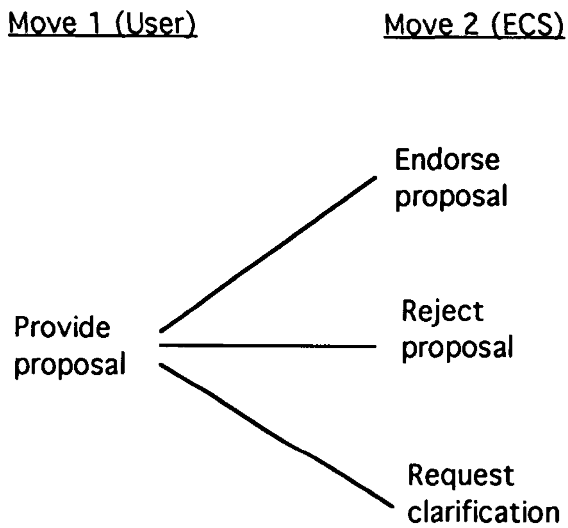

Figure 10 The opening round in the critiquing game (taken from an example in Dahl and Hellman, 1993)

Assuming that the ECS found the proposal to be incorrect and rejected it, the next move in the game could be defined as in Figure 11. (To deal with the "reject proposal" move, for example, the Disagreement aggregate presented in the previous section is applied, arguing against the user's proposal and for the ECS's—other aggregates can be defined to meet the possible moves in the game.)

Note that as the rules are normative, they can be defined purely at the discretion of the system constructor. In Figure 11 move 3, it might be more appropriate to see the "accept rejection" as the closing move of a 3-move round, while the "request clarification" is obviously an initiative move requiring a response, and could be viewed as a new round dominated by the first round.

Hopefully this example may suffice to show how a game can proceed. The ongoing activity is critiquing, and both partners are aware of their respective roles (the user provides a solution and the ECS critiques it). The aim of the game is to contribute to improvements in the user's solution. We have shown how a round can be defined, consisting of moves. In this example, the partners take alternate turns to make a move and the moves that are allowed are pre-defined by the rules (exemplified in Figures 10 and 11). Concluding the game is also defined in the rules. One suggestion

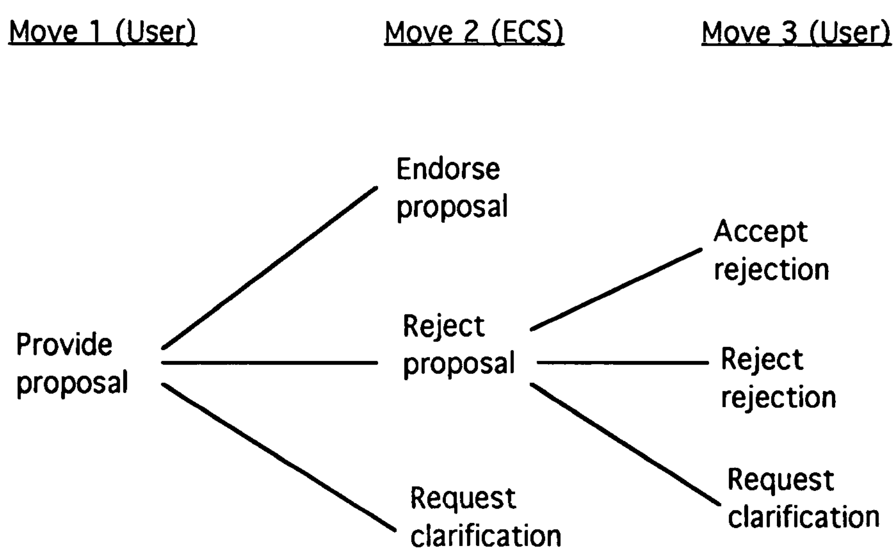

Figure 11 A subsequent move in the critiquing game (also taken from an example in Dahl and Hellman, 1993)

would be to allow the user to conclude the game at any time he pleased (he probably will anyway!) or, in more sedate terms, when he is satisfied that he has received enough feedback from the ECS.

5 Conclusions

It was noted in section 2 that much progress has been made over the last decade as far as natural language generation is concerned. This development has spawned a number of sub-areas dealing with specific aspects of generation and specific types of texts. In this paper, we have noted the ubiquitous use of argumentation in critiquing texts, and have shown how an argumentation model can easily be incorporated with RST.

This is perhaps one of the strengths of RST—that it is flexible enough to be applied in a variety of ways. However, we also noted the difficulty in taming this flexibility in language generation; defining RST schemas too specifically leads to domain-dependent generation, while defining them too generally does not provide enough information to the text generator to retrieve the content and organize it at an appropriate level of detail. We have chosen to describe aggregate schemas in general terms so that they can be applied across domains, and have utilized work done in argumentation theory to support this effort. This, in turn, leaves more to be done in other components of the generator, particularly as far as knowledge representation goes. Knowledge has to be represented in such a way that it is clear how grounds are related to claims, and our experience has shown that in a rule-based system, extra information has to be included in the rules to signify the status of various grounds, e.g., that a particular premise is linked to its conclusion by a causal link or an enablement link. While this requires overhead when acquiring and representing knowledge in the system, this information is essential in the generation of critiques and texts in general.

The language game metaphor allows for an explicit description of the way the interaction between user and ECS can be performed. Knowing what to expect from the ECS (and what not to expect) may facilitate the interaction as far as the user is concerned. It also allows us to regulate two of the issues raised by Silverman (*op. cit.*), namely initiative and timing in the critiquing process.

Acknowledgements

The author would like to thank Sture Hägglund for discussions on critiquing, and for comments on an early draft of this paper which have led to improvements. Jonni Harrius is also involved in the development of this work, in particular in helping to define and implement the rhetorical aggregates. Östen Dahl and Christina Hellman have also contributed their ideas freely during recent years, for which I am very grateful.

References

Allen, J, 1987. *Natural Language Understanding*, Benjamin/Cummings.

André, E and Rist, T, 1990. "Towards a plan-based synthesis of illustrated documents" In: *Proceedings of ECAI*, Stockholm.

Appelt, D, 1985. *Planning English Sentences*, Cambridge University Press.

Austin, J, 1962. *How to do things with words*, Oxford University Press.

Baker, M. "Analysing rhetorical relations in collaborative problem solving dialogues" (forthcoming).

Cawsey, A, 1989. "Explanatory dialogues" *Interacting with Computers* **1** (1) 69–92.

Cohen, P and Perrault, C, 1979. "Elements of a plan-based theory of speech acts" *Cognitive Sci.* **3** (3).

Cohen, P and Levesque, H, 1985. "Speech acts and rationality" *Proceedings of the 23rd Annual Meeting of the Association for Computational Linguistics*, Chicago, IL.

Cohen, R, 1987. "Analyzing the structure of argumentative discourse" *Computational Linguistics* **13** January–June.

Dahl, Ö and Hellman, C, 1990. "Rhetorical relations in discourse" *Papers from the Second Nordic Conference on Text Comprehension in Man and Machine*, Täby, Sweden.

Dahl, Ö and Hellman, C, 1993. "A language game analysis of critiquing" In: *Papers from the Third Nordic Conference on Text Comprehension in Man and Machine*, Linköping, Sweden (held in 1992).

Dale, R, Mellish, C and Zock, M (eds.), 1990. *Current Research in Natural Language Generation*, Academic Press.

Dale, R, Hovy, E, Rösner, D and Stock, O (eds.), 1992. *Aspects of Automated Natural Language Generation: 6th International Workshop on Natural Language Generation*, Trento, Italy.

Davey, A, 1978. *Discourse Production*, Edinburgh University Press.

Fischer, G, Lemke, A, Mastaglio, T and Morch, A, 1991. "The role of critiquing in cooperative problem solving" *ACM Trans. on Infor. Syst.* **9** (2) 123–151.

Grimes, J, 1975. *The Thread of Discourse*, Mouton Publishers.

Grosz, B and Sidner, C, 1986. "Attention, intention, and the structure of discourse" *Computational Linguistics* **12** (3) 175–204.

Halliday, M and Hassan, R, 1976. *Cohesion in English*, Longman.

Hobbs, J, 1985. *On the Coherence and Structure of Discourse*, CSLI Report CSLI-85-37, Stanford University.

Horacek, H, 1990. "An integrated view of text planning" In: Dale, R, Mellish, C and Zock, M (eds.) *Current Research in Natural Language Generation*, Academic Press.

Hovy, E, 1988. "Approaches to the planning of coherent text" In: *Proceedings 4th International Workshop on Natural Language Generation*, Los Angeles, CA.

Hovy, E, 1990. "Unresolved issues in paragraph planning" In: Dale, R, Mellish, C and Zock, M (eds.) *Current Research in Natural Language Generation*, Academic Press.

Hovy, E, 1990. "Parsimonious and profligate approaches to the question of discourse structure relations" *5th International Workshop on Text Generation*, Pittsburgh.

Kass, R and Finin, T, 1988. "The need for user models in generating expert system explanations" *International Journal of Expert Systems* **1** (4) 345–375.

Kay, M, 1979. "Functional grammar" In: *Proceedings of the 5th Annual Meeting*, Berkeley Linguistic Society.

McDonald, D, 1980. *Natural language production as a process of decision making under constraints*, PhD Thesis, MIT, Cambridge, MA.

McKeown, K, 1985. *Text Generation*, Cambridge University Press.

McKeown, K and Swartout, W, 1988. "Language generation and explanation" In: Zock, M and Sabah, G (eds.) *Advances in Natural Language Generation*, Pinter Publishers.

McKeown, K, Elhadad, M, Fukumoto, Y, Lim, J, Lombardi, C, Robin, J and Smadja, F, 1990. "Language generation in COMET" In: Mellish, C, Dale, R and Zock, M (eds.) *Current Research in Language Generation*, Academic Press.

Maier, E and Hovy, E, 1991. "A metafunctionally motivated taxonomy for discourse structure relations" In: *Proceedings of The Third European Workshop on Language Generation* pp 38–45, Judenstein, Austria.

Mann, W, 1984. "Discourse structures for text generation" In: *Proceedings of 10th International Conference of Computational Linguistics*, Stanford, CA.

Mann, W and Thompson, S, 1987. "Rhetorical structure theory: Description and construction of text structures" In: Kempen, G (ed.) *Natural Language Generation*, Martinus Nijhoff.

Mann, W and Thompson, S, 1988. "Rhetorical structure theory: toward a functional theory of text organization" *Text* **8**, 243–281.

Miller, P, 1986. *Expert Critiquing Systems*, Springer-Verlag.

Miller, P and Rennels, G, 1988. "Prose generation from expert systems" *AI Magazine* **9** (3), 37–44.

Moore, J and Paris, C, 1989. "Planning text for advisory dialogues" In: *Proceedings of the 27th Annual Meeting of the Association for Computational Linguistics*, Vancouver, Canada.

Moore, J and Pollack, M, 1992. "A problem for RST: The need for multi-level discourse analysis" *Computational Linguistics* **18** (4) 537–544.

Moore, J and Swartout, W, 1989. "A reactive approach to explanation" In: *Proceedings of IJCAI* pp 1504–1510, Detroit, MI.

Rankin, I, 1989. *The Deep Generation of Text in Expert Critiquing Systems*, Licentiate Thesis No. 184, Department of Computer and Information Science, Linköping University.

Rösner, D and Stede, M, 1992. "Customizing RST for the automatic production of technical manuals" In: Dale, R, Hovy, E, Rösner, D and Stock, O (eds.) *Aspects of Automated Natural Language Generation* pp 199–214, Springer-Verlag.

Searle, J, 1969. *Speech Acts. An Essay in the Philosophy of Language*, Cambridge University Press.

Silverman, B, 1991. "Expert critics: operationalizing the judgement/decision making literature as a theory of 'bugs' and repair strategies" *Knowledge Acquisition* **3**, 175–214.

Silverman, B, 1992a. "Survey of expert critiquing systems" *Communications of the ACM* **35** (4).

Silverman, B, 1992b. *Critiquing Human Error*, Academic Press.

Simmons, R and Slocum, J, 1972. "Generating English discourse from semantic networks" *Communications of the ACM* **15** (10).

Toulmin, S, 1958. *The Uses of Argument*, Cambridge University Press.
Toulmin, S, Rieke, R and Janik, A, 1979. *An Introduction to Reasoning*, Macmillan.
van Eemeren, H and Grootendorst, R, 1983. *Speech Acts in Argumentative Discussions*, Foris Publications.
Wahlster, W and Kobsa, A, 1989. "User models in dialog systems" In: Kobsa, A and Wahlster, W (eds.) *User Models in Dialog Systems*, Springer-Verlag.
Walton, D, 1989. *Informal Logic. A Handbook for Critical Argumentation* pp 3–13, Cambridge University Press.
Weiner, J, 1980. "BLAH, a system which explains its reasoning" *Artificial Intelligence* **35** (15).
Woods, W, 1970. "Transition network grammar for natural language analysis" *Communications of the ACM* **13** 591–606.
Zock, M, Adorni, G and Ferrari, G (eds.), 1993. *Preprints from the Fourth European Workshop on Natural Language Generation*, Pisa, Italy.