

# Implementing an Integration of the Systemic Flowchart Model of Dialogue and Rhetorical Structure Theory

Yuen Q. Lin and Robin P. Fawcett

Computational Linguistics Unit, Aberconway Building, University of Wales, Cardiff CF1 3XB, UK

e-mail: liny@cardiff.ac.uk, fawcett@cardiff.ac.uk, fax: 01222-874806

## 1. Introduction

There are two major types of language generation.<sup>1</sup> The first of these is monologue generation, which focuses on generating monologue text, typically of paragraph length. The second is dialogue generation, the aim of which is to produce co-operative, interactive discourse. Typically projects in natural language generation focus on one or the other, but some researchers are now considering how the two fit together. This paper is one of a series that discuss how this may be done, at the second stage of generation, i.e. the generation of the structure of **discourse**.

The most influential approach to planning discourse monologues in recent years has undoubtedly been Rhetorical Structure Theory (RST). While there is much to discuss about this framework, for our present purposes we shall adopt and accept the 'standard' version, as found in Mann and Thompson (1987) and the many papers that build on it. For dialogue we use what we take to be the most holistic and most explicit development of the Birmingham School of discourse analysis (originating with Sinclair and Coulthard 1975), namely the Systemic Flowchart Model (SFM). This was first described in Fawcett, van der Mije and van Wissen (1988).

The present paper follows on from two earlier papers in particular. The first describes the implementation of the SFM as part of the COMMUNAL Project (Lin, Fawcett and Davies 1993).<sup>2</sup> The second is Fawcett and Davies' programmatic paper (1993), which describes how an integration of RST and the SFM might be attempted. The purpose of the present paper is to give an account of an implementation of the point where the two models are integrated - including a new system network through which this is achieved. We see no reason why other contributions that draw on a broadly systemic functional approach to modelling RST relations, such as Vander Linden, Cumming and Martin (1993) and Maier and Hovy (1993), may not be integrated with the present proposals.

## 2. Some problems in relating RST to the SFM

Fawcett and Davies (1993) provide summary descriptions of the two approaches, so we shall not replicate those descriptions here. That paper also discusses some of the possible difficulties in integrating the SFM and RST.

Fawcett and Davies concluded:

We do not expect a 'seamless join' between the two models to occur effortlessly; there are bound to be difficulties of many kinds as we explore possible ways of bringing the two models together. But, given the richness and flexibility of the descriptive and implementational tools available to us, it is an enterprise on which we can enter with an expectation of some measure of success.

Here we shall report both the way in which we have successfully modelled the point in the overall discourse grammar at which RST and the SFM meet, and the surprising ease with which we were able to do it.

For the Birmingham School of Discourse Analysis, a dialogue consists of a series of transactions, each of which is made up of a number of exchanges. These in turn have constituents called **moves**, and each move consists of one or more **acts**. Thus their model assumes that there is a **rank scale** relationship between these units (Halliday 1961). This concept is found in the SFM too.

In RST, on the other hand, there is the potential for the unlimited recursion of structures consisting of a **nucleus** with a **satellite** in the relation of **elaboration** or **reason**, etc. In the integration of the two to be described here we model this as the recursion of acts within acts.

The SFM model of discourse structure is both richer and at the same time more traditional than the types of structure built in RST. The main difference in 'richness' is that each node is labelled twice. It is shown as both an **element of structure** in the unit above, and by the name of the unit that **fills** it. Thus the concepts of 'nucleus' and 'satellite' are treated here as elements of structure. The SFM is more traditional in that its structure is based on 'constituency' rather than the concept of 'sister dependency', as in RST. This provides a richer and so more informative labelling, with the rhetorical relation shown as the **class** of act.

## 3 A typical structure

We will now look at a typical example of the sort of structure that occurs at the point where the rank-based structure of the SFM meets the potential recursion of RST relations - using for both a 'constituency' approach that has at each node both elements and **units**.

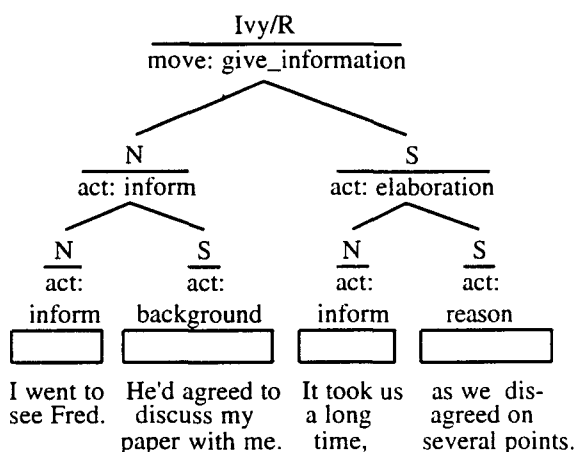


Figure 1: A fragment of a discourse structure at the point where RST and the SFM meet

How is this structure generated? To see this, we must look at the new system network shown in Figure 2 - and also at its associated realization rules, because it is through these that the chosen features are converted into structures.

#### 4 How the structure is generated

The discourse generator that we are about to describe is called GENEDIS, because it GENERates DIScourse (For details see Lin et al. 1993.) Many of its choices are guided by the goals set in the higher planning component, but much of the 'discourse potential' is captured here. As it enters the network in Figure 2, it knows that it has the goal of filling the **Respond** element (hence 'R') of an exchange, and that it is to be uttered by Ivy (i.e. the system). The network operates in the following manner. We enter it at [discourse unit], and find that the probabilities in the initial system are set 100% to [move]. (On (re-entry to the network to fill out the lower part of the tree, however, this choice will be reversed.) As you will see if you look at the realization rules, the selection of [move] inserts first the unit 'move' into the structure, and then locates the element nucleus (N) at Place 2 in that move's structure. (We shall shortly see why it is at Place 2 rather than Place 1.)

We now encounter two 'simultaneous' systems. In the MOVE\_CLASS system, GENEDIS typically chooses one of the five major classes of move. The realization rule on each of these inserts this into the structure (the relationship of 'class' being shown by a colon). Notice that the categories that get inserted into the growing structure may have similar or even identical labels to the features in the network. In principle the two are separate, but there is little point in multiplying terms unnecessarily.

In the lower system, i.e. MOVE\_COMPLEXITY, our choice is [with\_satellite\_m]. (Lack of space prevents us from describing the effects of choosing [simple\_m] or [co\_ordinated\_m], but the rules given cover the former.)

However the typical choice - and the most

interesting one - is [with\_satellite\_m]. The choice of this feature affects the structure in four ways. First, its realization rule (0.4) provides for re-entry to the network to fill N. It is this fact that provides for the recursion typical of RST relations in the present framework, as we shall shortly see.

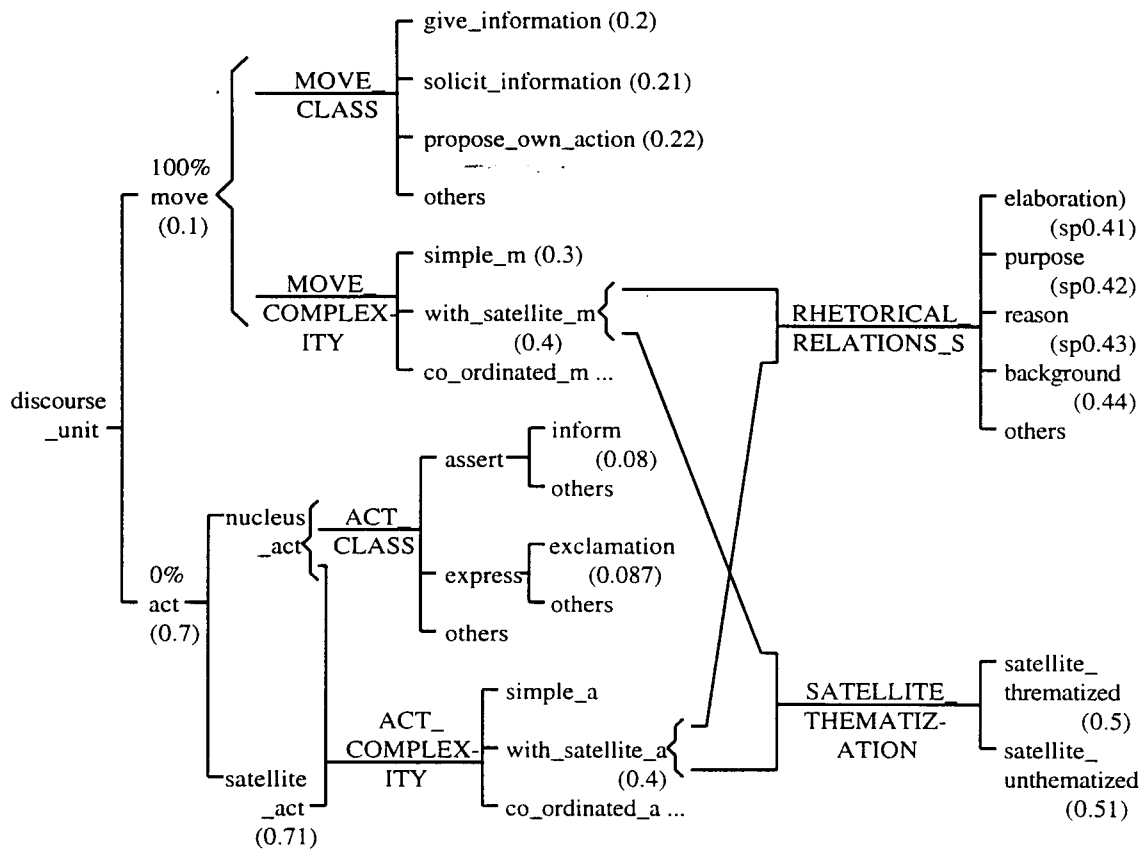
But this feature also leads on to two further simultaneous systems: one for RHETORICAL RELATIONS and a second for SATELLITE THEMATIZATION. We will take the latter first. The concept of 'thematizing' an element in the structure of a unit comes from systemic functional grammar, where it is mainly associated with the clause. We use the same term here because the motivation is similar in both.

Typically, the satellite follows the nucleus, but the different types of rhetorical relation have different typical patterns in this respect. So how does GENEDIS know which to choose? The answer lies in the upper system, and the entry to this is the third effect of choosing [with\_satellite\_m]. As you will see, the features here are precisely those of the rhetorical relations of RST. But notice that these features have attached to them **same pass (sp) preference resetting rules** (i.e. these rules are not realization rules). It is these that express the likelihood that the satellite will precede or follow the nucleus - as the sp rules below the network show. One advantage of using probabilities on features in systems is that they can be varied, e.g. for different types of genre (cp. the prevalence of thematized 'purposes' in certain types of instructions). In some cases the probabilities are absolute, as in our case, where the sp rule on [elaboration] makes it 100% certain that the S will follow. In any given instance, of course, the Performer may have a reason to thematize a satellite that overrides the probabilities, but in a sophisticated model he/she should be able to set this against the knowledge of the general probabilities for a given type of rhetorical relation.

The final effect of selecting [with\_satellite\_m] is to provide that the network will be re-entered to fill S, as in Rules 0.5 and 0.51.

We have now generated all of the structure of the move. This part of the network is non-recursive, but as we now re-enter the network to generate the acts that will fill the N and S of the move, we meet a recursive system network. The recursion is modelled through the realization rules; a new layer of structure is added by each re-entry to the network. And the potential for recursion is, in principle, infinite.

On re-entry to the network to fill N, the preferences have been reset to [act] and [nucleus\_act]. Similarly, on re-entry to fill S they are reset to [act] and [satellite\_act]. Rule 0.7 inserts the unit 'act' in both cases. The class of act to fill N comes from the network for ACT\_CLASS, and in our case Rule 0.8 inserts 'inform' after 'act'. It is Rule 0.71 on [satellite\_act] that determines what the class of the act filling S will be. Thus, since in our case [elaboration] has been chosen on the 'mother'



#### SELECTED SAME-PASS PREFERENCE RE-SETTING RULES

- sp0.41 : elaboration : for same\_pass prefer [satellite\_unthematized].  
 sp0.42 : purpose : for same\_pass prefer [40% satellite\_thematized / 60% satellite\_unthematized].  
 sp0.43 : reason : for same\_pass prefer [20% satellite\_thematized / 80% satellite\_unthematized].  
 sp0.44 : background : for same\_pass prefer [1% satellite\_thematized / 99% satellite\_unthematized].

#### SELECTED REALIZATION RULES

- 0.1 : move : N @ 2.  
 0.2 : give\_information : give information.  
 0.21 : solicit\_information : solicit information.  
 0.3 : simple\_m : for N prefer [act, nucleus\_act, simple], for N re\_enter\_at discourse\_unit.  
 0.4 : with\_satellite\_m or with\_satellite\_a :  
   if with\_satellite\_a then N @ 2,  
   for N prefer [act, nucleus\_act, for N re\_enter\_at discourse\_unit].  
 0.5 : satellite\_thematized :  
   S @ 1, for S prefer [act, satellite\_act, for S re\_enter\_at discourse\_unit].  
 0.51 : satellite\_unthematized :  
   S @ 3, for S prefer [act, satellite\_act, for S re\_enter\_at discourse\_unit].  
 0.7 : act : act.  
 0.71 : satellite\_act :  
   if on\_mother\_pass elaboration then act\_class elaboration,  
   if on\_mother\_pass purpose then act\_class purpose,  
   if on\_mother\_pass reason then act\_class reason,  
   if on\_mother\_pass background then act\_class background.  
 0.8 : inform : act\_class inform.  
 0.87 : exclamation : act\_class exclamation.

Figure 2: A system network (and its realization rules) that integrates the Systemic Flowchart Model of dialogue structure with the relations of Rhetorical Structure Theory

pass through the network the **class** of the act will be an 'elaboration'.

The final two systems are ACT\_COMPLEXITY and SATELLITE\_THEMATIZATION - and it is these that provide for the recursion. As you will see, the first is entered from both [nucleus\_act] and [satellite\_act]. And one of its features is [with\_satellite\_a], whose Rule 0.4 provides for re-entry to fill N. Finally, it leads on to choices in 'thematizing' the S, and to the realization rules that provide for re-entry - like those for the S in a move.

Thus the network and realization rules together provide for the recursive embedding of acts with further acts at either N or S or at both. In this way, then, we do indeed move virtually 'seamlessly' from the SFM structure to RST relations.

## 5. Conclusions

The generation of the discourse structure of monologues and dialogues are usually treated as separate research paradigms, but in this paper we have shown how the gap between the two can be bridged. As we have seen, RST relations can be fully integrated with the richer SFM framework by restating the 'sister-dependency' relations of RST as occurring via their mother unit, i.e. in an 'element and unit' model. In this way we have arrived at a unified framework for generating both dialogues and monologues.

How was this integration accomplished so relatively easily? The answer seems to be that GENEDIS, which is itself adapted from the sentence generator GENESYS, provides an appropriately rich and relevant array of operators. It can therefore be extended, in a principled manner, to incorporate the simpler set of relations of RST. This suggests in turn the value of the particular set of concepts incorporated into the Cardiff Grammar, and of the grammar-writing tool DEFREL, in which both GENESYS and GENEDIS are written.

Essentially, we have here incorporated RST within the SFM rather than the other way about. There is a good reason for this. It is that dialogues regularly contain monologues, but monologues (while they can be interrupted by 'mini-dialogues' to clear up misunderstandings and to challenge, etc.) occur - either explicitly or implicitly - within a dialogue.

The fundamental principle underlying all discourse generation is that there is always an Addressee - so that there is always at least an implicit potential for dialogue.

## References

- Dale, R., Hovy, E.H., Roesner, D., and Stock, O., (eds.) 1992. *Aspects of Automated Natural Language Generation*. Berlin: Springer.
- Fawcett, R.P., and Davies, B.D., 1992. 'Monologue as a turn in dialogue: towards an integration of exchange structure and rhetorical structure theory'. In Dale et al., (eds.) 1992, pp. 151-66.
- Fawcett, R.P., Tucker, G.H., and Lin, Y.Q., 1993. 'How a systemic functional grammar works: the role

of realization in realization'. In Horacek and Zock 1993, pp. 114-86.

Fawcett, R.P., van der Mije, A., and van Wissen, C., 1988. 'Towards a systemic flowchart model for local discourse structure'. In Fawcett, R.P., and Young, D.J., (eds.) 1988. *New Developments in Systemic Linguistics, Vol 2: Theory and Application*. London: Pinter, pp. 116-43.

Halliday, M.A.K., 1961. 'Categories of the theory of grammar'. In *Word* 17, pp. 241-92.

Horacek, H., and Zock, M., (eds.), 1993. *New Concepts in Natural Language Generation*. London: Pinter.

Lin, Y. Q., Fawcett, R. P. and Davies, B. L. 'GENEDIS: the discourse generator in COMMUNAL'. In A. Sloman, D. Hoggs, G. Humpherys, A. Ramsay, and D. Partridge (eds.) *Frontiers in Artificial Intelligence and Applications*. Amsterdam: IOS Press. pp. 148-57.

Maier, E., and Hovy, E., 1993. 'Organizing discourse structure relations using metafunctions'. In Horacek and Zock 1993, pp. 69-86.

Mann, W.C., and Thompson, S.A., 1987. 'Rhetorical structure theory'. In Polyani, L., (ed.) *Discourse structure*. Norwood, N.J., Ablex.

Sinclair, J. and Coulthard, M. (1975) *Towards an Analysis of Discourse*. Oxford: Oxford University Press.

Matthiessen, C.M.I.M., and Bateman, J.A., 1991. *Text Generation and Systemic Functional Linguistics*. London: Pinter.

Vander Linden, K., Cumming, S., and Martin, J., 1992. 'Using system networks to build rhetorical structures'. In Dale et al. 1992, pp. 183-98.

## Notes

1. The research reported here is part of the COMMUNAL Project. COMMUNAL is or has been supported by grants from the Speech Research Unit at DRA Malvern as part of Assignment No. ASO4BP44, on Spoken Language Understanding and Dialogue (SLUD), from the University Research Council of International Computers Ltd and from Longman, and directly by the University of Wales, Cardiff. We also wish to thank the anonymous reviewers of this paper, whose comments, advice and queries have led to a complete re-writing of it. We think that the result is a clearer presentation of the ideas.

2. COMMUNAL stands for CONvivial Man-Machine Understanding through NATural Language, and it is a long-term project in building a system for communication with computers that draws on systemic functional linguistics (supplemented by other concepts when they are found useful). At its heart lies the GENESYS sentence generator, so called because it GENERates SYStemically, i.e. using a systemic functional grammar (SFG). A complete revision of the earlier version is now being developed, incorporating many new concepts. For what is, surprisingly, the only complete published account of how a SFG generates a sentence, see Fawcett, Tucker and Lin (1993). The lexicogrammar used in GENESYS — and also for work in textual description — is increasingly being referred to as 'the Cardiff Grammar'. For an overview of the Penman SFG generator see Matthiessen and Bateman 1991.