SOME REMARKS ON SYSTEMS WITHIN SYSTEMIC LINGUISTICS

Francisco Martín Miguel

Systemic-functional linguistics has grown in importance for the last few decades, struggling its way into the linguistic field as opposed to the American generativist trend, which meant a break-through in the treatment so far given to natural language. One of the main differences between systemic-functional linguistics (henceforth SFL) and other schools of linguistics is the fact that predominance is attached to paradigmatic relations as the basis for language generation (and perhaps, interpretation).[1] The aim of this article is to try to shed some light on how far system networks, the basic tool in systemic analyses, can help us develop a full, comprehensive model of linguistic theory. I will provide an example of how the level of semantics is linked to the level of lexis in the systemic framework as an attempt to merge both into a lexicogrammatical stratum. The example is based on a study on computational linguistics by Danlos (1987), which aims to show the problems that arise on trying to supply an adequate mechanism for semantic representation. I will try to explain how such problems can be accounted for within SFL.

There is ample literature on system networks already from the beginning of SFL.[2] We must remember that 'system' was simply one out of four units proposed by M.A.K. Halliday in the origin of his Scale and Category Grammar,[3] the embrio of SFL. To bring the notion of 'system' to the foreground of the model proposed by Halliday brought about one important consequence: the syntagmatic view of language became second in linguistic analysis. From this moment onwards paradigmatic relations attracted everyone's attention and interest in the field of systemic research. The 'system' became the dominant concept of the theory, hence the name of which was changed into Systemic Grammar (Hudson 1974:5). System networks of a semantic nature began to be proposed for the different levels (strata) of language. In principle, system networks were intended to reflect semantic distinctions. However, systemic linguists began to build both syntactic, lexical, and semantic networks. The reason for this line of research may be the lack of consistency and clarity in Halliday's writings, as noted by Butler (1985:78-80).[4]

To start with, let us comment on some facts related to system networks that raise some questions –if not doubts– about the general validity of systems as research tools in SFL. Because of the very nature of system networks, we should consider two processes: i) recursion, and ii) the insertion of abstract entry conditions (following, perhaps, generativist linguists), posed in Martin (1987). Recursion allows us to step backwards in the depth of delicacy of a system network, in order that two or more equal features could stand in linear order. This is possible by joining a feature and its entry condition. The need for recursive patterns in linguistics is, I believe, obvious. However, stated in the above terms, recursion outputs undesirable results. On the one hand, there is no stipulation as to when recursion must stop, i.e. it is always possible to re-enter the system infinitely. Either intra- or extra-linguistic restrictions should be specified so that recursion could no longer be possible. This apparent drawback could be solved, rather redundantly, if we follow Bateman's suggestion of synoptic and dynamic systems (Bateman 1989). In fact, Bateman favours synoptic systems implemented, when necessary (when is that?), by dynamic systems, i.e. recursive systems accessible by means of connected stars in the systems themselves (Bateman 1989:261-263). On the other hand, system networks are supposed to reflect paradigmatic (semantic) options which have some reflex in (grammatical) form. Recursion is *not* a paradigmatic process at all, but syntagmatic in nature. If system networks are to remain purely paradigmatic, recursion should be dealt with elsewhere in the grammar. Perhaps it would be much simpler to stipulate recursive patterns in the realisation rules, which come after the system networks and turn abstract features into what is traditionally considered to be syntax. Fawcett (1980:200) acknowledges the problems raised by recursive systems and favours a lexical solution, by stipulating *ad hoc* restrictions in the lexical items themselves. One fundamental claim must be stated clearly: whatever the possible solution to this problem might be, paradigmatic and syntagmatic relations within a formal linguistic framework of the kind we are dealing with must be kept apart, if we want to stick to the goal of internal coherence.

As far as the insertion of abstract entry conditions in the system networks is concerned, one thing can be said in favour of Martin's arguments: by doing so, simpler networks are undoubtedly obtained. Yet in the mid-sixties authors such as Dixon, reviewed by Matthews (1965), thought that simplicity should be one of the basic requirements in language theory. This view is shared in orthodox generative linguistics. However, Martin's networks combine 'meaningful' and 'empty' features as entry conditions to deeper networks. This is not formally adequate, since two (or more) features, which serve as entry conditions to other networks, must reflect some semantic distinction. 'Empty' features resemble variables in the generative literature in the sense that no particular content is assigned to them. Such features play no part in the realisation rules, since they are not semantically contrastive. Fawcett's excuse to include them in a network is clear:

> The guiding principle, then, in introducing features to a semantic network, is that one or more of the features in the relevant system must have some part to play in the realisation rules. In practice, however, all systemic grammarians seem additionally to introduce features to label nodes that are necessary in the network because they provide entry conditions to further systems, in order to aid the 'readability' of the network (Fawcett 1980: 101).

I deem the excuse of readability is not pertinent if we aim to achieve a formal linguistic object.

If we are to validate any system network we propose, I think the most objective, man-neuter procedure is to make the system at issue work in a computer program. Any linguist wants his suggestions to be always fruitful and work well within his project. Therefore, the temptation to reach the 'true' system network for a certain area of linguistic analysis might lead the linguist to, either not validate his hypothesis, or turn a blind eye to possible deficiencies the system may have and present his hypothesis as a valid one. Implementation by means of computers avoids this, and shows that the system in question either works or does not work. It may be a frustrating methodological aid, since it can ruin hours of work by simply pressing a button. Many a linguist may think it is too big a cost for their valuable efforts. However, computers may bring the necessary help to find the right way through in linguistic research.

The relation between SFL and the field of computational linguistics is not a recent one. Yet in the late sixties and early seventies –if not earlier–, linguists such as Winograd, Mann or Matthiessen started working to build a systemic grammar into a computer. Some outstanding models are Nigel or Penman. It happens that SFL seems to lend itself very well to being applied to computers, as Butler (1990:16-17) notes.[5]

Facing 'text' in Halliday's terms, i.e. as a semantic unit, two possible aspects are to be taken into account: text as a process (dynamic) and text as a product (static) (Halliday & Hasan 1989:10). Computer implementation of language favours the dynamic side, rather than the static one, merely because text generation is commonly the goal of computational linguistics. 'It is important to note that the text-production perspective is a dynamic one, while that of text analysis is synoptic' (Lemke 1991:26). Halliday himself recognized three different dynamics: the third one is the dynamic of the unfolding of a text (Halliday 1991:41, 44-45). He suggested a bimodal distribution of systems, according to the relative probability of appearance of the features contained in them: i) equiprobable, and ii) skew systems. For Halliday, such systems bear the key to language change in linguistic interaction. Nowhere does he make a similar distinction as regards synoptic (or static) systems.

So far we have been referring to system networks as the representation of paradigmatic relations. However, some authors remind us that the notion of 'system network' should not be mixed up with that of 'flowchart', the latter understood as embodying syntagmatic relations. In this respect, Fawcett's and Davidse's opinions are of interest here. Davidse (1985:45) remarks that a system network's should not be confused with a flowchart, which also represents interdependent choices but choices that succeed each other in time. As we can see, Davidse introduces the element of time to distinguish 'system' from 'flowchart'. It is clear that text production involves time in some sense: words cannot be uttered all at the same time, but one after the other, in sequential order. This seems to stand in contradiction with the fact that systems networks map concurrent choices, i.e. choices at the same time. Purely speaking then, computational linguistics would operate with flowcharts, not with systems, since they mainly deal with dynamic processes.

Related to this apparent contradiction, a comment is called here on the type of rule proposed by Terry Patten (1988:57). He tries to build a systemic grammar for computers and hypothesizes that, for every feature in a system, there should be one rule of the type:

> if the entry condition of the system is satisfied,
> then choose this feature and perform the actions
> specified by the realisation rules

Such a rule permits movements forth and back alongside both the system network and the realisation rules, since once a certain feature is selected, the computer is told to go and look at the realisation rules to perform what they state. Then, the machine has to go back to the system and repeat this process. This is so because the realisation rules come after the system networks; in fact, the output of a certain system network makes up the selection expression, which is the input for the realisation rules. In principle, a realisation rule cannot work unless the whole of the selection expression is formed, which is so after traversing the whole of the system network. Apart from this, Patten does not say whether the rule he proposes is supposed to be ordered with respect to the system networks and the realisation rules. The very definition of the rule seems to imply that it must be applied the last one of the three, since it refers back to both the system networks and the realisation rules. However, it is not a 'self-contained' rule, i.e. it needs the presence of the system networks and the realisation rules, and adds no new material to the formal model.

Fawcett notes that flowcharts have to do with syntagmatic processes –thus the idea of time sequence or constituency–, whereas systems are related to paradigmatic ones. Even though these two notions should, in principle, be kept apart, he acknowledges that

> the two notions can be related, if we see the points in a flow chart at which decisions are to be made as points at which system networks can be inserted. The main difference between most flow charts that one encounters and a systemic grammar for a language can then be expressed as follows: that a flow chart typically shows MANY sequential relationships with extremely SIMPLE systems (which typically offer just two options in behaviour at each point) while a systemic grammar shows relatively FEW syntagmatic relationships (with 'syntagmatic' being interpreted in terms of 'constituency' rather than, sequence´), but with immensely COMPLEX sets of choices at the decision points (Fawcett 1980:198-199).

Computer implementation has some more advantages. Apart from making the grammar more explicit, it can best describe the dynamic potential of language, as noted by Bateman (1989: 264). He thinks that 'it is possible to show that the process of making a theory computationally explicit can improve considerably our understanding of what the theory is and what it is possible for the theory to say'. Besides he remarks that computer implementation can be a very beneficial methodology in linguistic research (Bateman 1989:266). However we should not be so pessimistic as Terry Patten is when he reminds us that 'one disadvantage of systemic grammar is that it has never been formalized as more traditional grammars have. This makes it difficult to provide a formal model for this approach to text generation' (Patten 1988:5). It is true that SFL has not paid so much attention to formalization as other schools of linguistics, such as, e.g. generative linguistics, but some attempts have been successfully carried out by linguists such as Bateman, Winograd, Mann, Hovy, and others. On the one hand, Patten's view is, perhaps, fostered by the overall attitude of the father of SFL, who overtly denied that criteria such as simplicity, elegance or explicitness belonged to his scientific domain. As Halliday observes, systemic grammar is 'an extravagant theory, not a parsimonious one' (Halliday 1985:xix). On the other hand, it is generally recognized that for a grammar to be generative, elegance, simplicity, explicitness, clarity and precision seem to be high priorities.[6]

Perhaps one of the fields of study in which least work has been done –within SFL, of course– is that of building lexical networks. In 1980 this area was nearly unexplored, apart from some work done by Berry (1977:62-63) and some attempts by Fawcett (1980:152). The need for lexical networks is clear, since they include semantic descriptions of lexical items, which are, by and large, the basic units used in communication. Furthermore, lexical networks are essential to undertake computational linguistics, since 'the aim of automatic generation is the production of texts from "raw data" or from "abstract" semantic representations which must allow us to paraphrase, draw inferences and produce texts in several languages' (Danlos 1987:46). Lexical networks achieve this 'abstract-semantic-representation' goal of automatic generation. Of course, networks at other levels are needed too: syntactic, stylistic, phonological, semantic, and so on.

Let us concentrate on some of the work by L.Danlos for our purposes of exploring system networks in SFL. In her 1987 book Danlos tried to make the reader aware of how important linguistics is to build a computational grammar. The field to explore was too wide, so she focused her argumentation on some aspects of English grammar, so as to make things clearer. Among other things, she analyzed the linguistic properties of five English verbs: *kill, murder, assassinate, injure* and *wound.* They all belong to the same semantic field: 'murder attempt'. Danlos (1987:71-75) provides the following list of examples to draw her conclusions:[7]

(1) Anarchists (assassinated + ?*murdered) the Pope in Paris yesterday

(2) Hooligans (*assassinated + murdered) an old lady in Paris yesterday

(3) John shot Mary once in the leg (wounding + *injuring) her seriously

(4) A lightning bolt (*wounded + injured) Mary

(5) A bomb killed four policemen...

(6) *A bomb assassinated the Pope...

She illustrates the semantic differences by means of labels such as 'target', 'result', 'unknown', 'famous', 'death', 'wound', 'accidental' and 'intentional'. Examples (1) and (2) show that: i) both *assassinate* and *murder* require Agents as grammatical Subjects; and ii) the Object of *assassinate* must be someone of note, whereas that of *murder* must not. Examples (3) and (4) prove that whenever *wound* is used, an intentional action is involved, whereas *injure* implies an accidental one. Finally, examples (5) and (6) demonstrate that, whereas the Subject of *kill* need not be an Agent, it seems that *assassinate* rejects a non-Agent as Subject. Adopting the same semantic descriptions, I tentatively propose the following preliminary systemic network for such a reduced area of the English lexicon (Fig. 1).

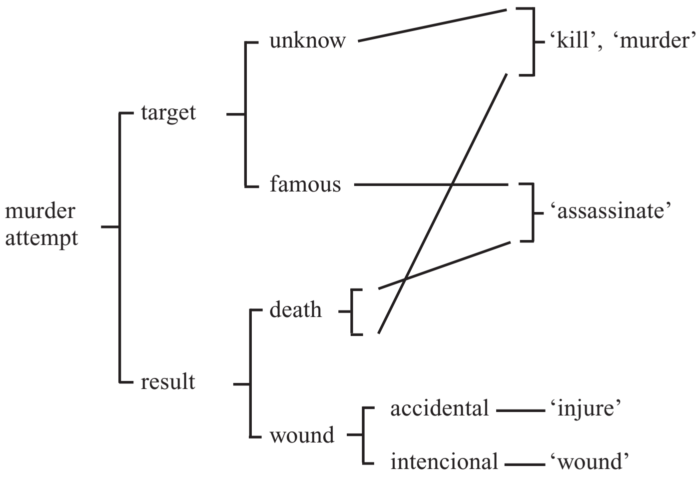

Figure 1: a preliminary attempt

The syntactic restrictions above mentioned are not shown in the system network. That is, according to (6), *assassinate* rejects a non-Agent as grammatical Subject, which must be stipulated if (6) is to be ruled out as ill-formed. Something similar happens with (1) and (2): it seems that *murder* only accepts agentive Subjects –even though Danlos does not provide a conclusive example like that of (6).

Danlos seems to equate *kill* and *murder* –at least, she does not explicitly make any formal distinction between the two verbs–. If we look these verbs up in some dictionaries chosen at random, the definitions they give are as follows:

	The Collins English Dictionary
'kill' 'murder'	to cause the death of (a person or animal) to kill someone unlawfully with premeditation or during the commission of a crime

	Webster's New World Dictionary
'kill' 'murder'	to cause the death of; to make die to kill (a person) unlawfully and with malice

The dictionaries provide us with the label 'unlawfully' to distinguish *kill* from *murder.* I think the best solution to handle this question, as regards modifying the system network proposed, is to regard *kill* as the basic, unmarked verb of those implying death: *kill , murder* and *assassinate.* I think it is not necessary to insert another label. To know about who/what resulted in death is not important in the verb *kill.* This piece of information *is* relevant in *murder* and *assassinate.* Therefore, the selection expression for *kill* seems to suffice with: [murder attempt, result: death]. A slight modification is, then, included in the system network (Fig. 2).

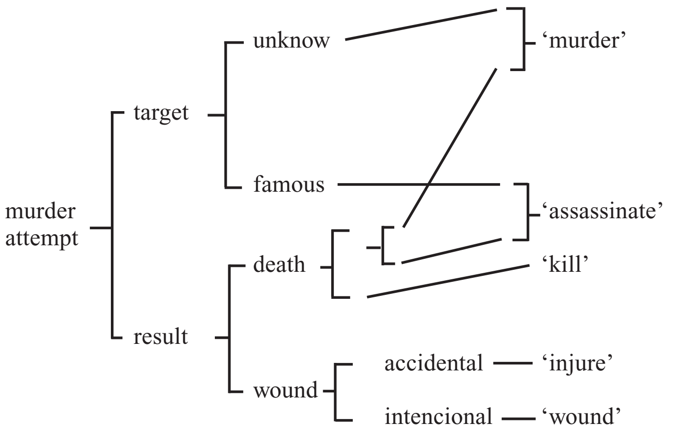

Figure 2: 'murder attempt' system modified

However, the system is not accurate enough yet. If we collate the definitions given for the verbs *murder* and *assassinate*,[8] *a conclusion can be drawn: both verbs involve an unlawful action. This common feature allows us to make further readjustments to the system in Figure 2. These concern: i) the insertion of a new label ('unlawful') to distinguish kill* (the unmarked form) from *murder* and *assassinate;* and ii) the re-positioning of the 'target'system, which only affects 'death', not the whole 'murder attempt' system. These modifications are, then, included in (hopefully!) the final 'murder attempt' system (Fig. 3).

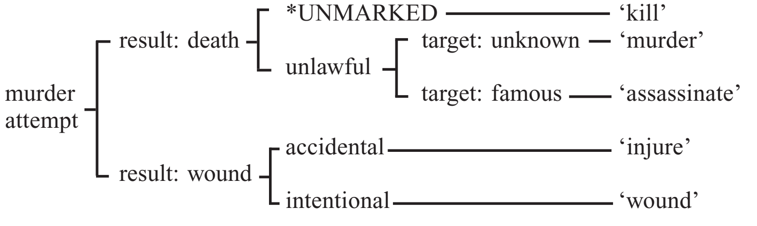

Figure 3

The scope of the semantic span analized here is very reduced. Ideally the system network proposed in Figure 3 should be included in a much larger system including wider areas of the English lexicon. This larger system would, then, operate simultaneously with the rest of the systems in the lexicogrammatical level of the theory. The features used in the networks seem to suffice to supply a semantic description of the lexical items in question. There is still one problem pending: how to formalize the fact that *assassinate* subcategorizes for an agentive Subject. This piece of information is dealt with in the realisation rules. The formalism employed in SFL is of the kind AG: SUBJECT, which is added to the lexical item *assassinate.*

In summary, I have tried to give a critical account of some theoretical problems concerning system networks in SFL. Problems such as the use of void features together with semantic ones, or how to come to terms with recursion in a paradigmatic account of the language, are still unsolved. The field of computational linguistics is constantly asking for ways to describe the language in a systematic, semantic fashion. Here is where SFL best fulfils this requirement. The system network I propose is by no means exhaustive. It simply served as an example of the need to start building lexical networks within a systemic framework. It is true that the more words we include in the network, the more intricate and complex it becomes. In this way, the theory itself will have the tools to explain why certain combinations of words are, or are not, possible in language. Every language has its own peculiarities as regards how certain words combine with others in an idiomatic way. Some of such peculiarities –if not all– can be captured by means of system networks of a lexical nature. Lexical networks would, then, serve the purpose of relating the level of semantics –in its pure abstract sense– with that of lexicogrammar.

Notas

1. Authors such as Fawcett (1980: 6) take semantics to be the generative base of the language system. Later on in the book he suggests that his generative systems could be used to interpret language as well, i.e. to decode the message, by simply reversing the directionality of the procedure.
2. A good summarized version on system networks is available in Berry (1975). Some important developments by J.R. Martin appear in a later book, Halliday & Fawcett (1987).
3. The main outline of this stage of the theory appears in Halliday (1961).
4. Three (in some writings, four) metafunctions were proposed by Halliday, which were said to belong to the lexicogrammatical level first, then to the semantic stratum. This, together with the very few examples included in Halliday's writings, has led some systemic linguists to divert from the orthodox track of SFL, in Halliday's opinion (personal communication).
5. Unlike Dik's Functional Grammar, Halliday's model is applied 'in several fields of research: computational linguistics, description of child language acquisition, stylistics, educational linguistics.'
6. Even though Paul Ziff (1969: 223) recognizes that formal languages, however perfect, elegant, precise, and the like, are not valid tools for communication at all. He states this in stronger terms: 'the utter inutility of the formal language as a language with which to communicate in the world in which we find ourselves'.
7. The list is slightly abridged, since not all the examples are relevant here.
8. Assassinate: 'to murder (a person, esp. a public or political figure), usually by a surprise attack.' (Hans 1979).

References

Bateman, J.A. (1989) "Dynamic Systemic-Functional Grammar: A New Frontier," *Word*, 40, pp. 263-286.

Berry, M. (1975) *Introduction to Systemic Linguistics*. [I] New York: St. Martin's Press.

Berry M. (1977) *Introduction to Systemic Lingusitics.* [II] New York: St. Martin's Press.

Butler, C.S. (1985) *Systemic Linguistics: Theory and Applications*. London: Batsford.

Butler, C.S. (1990) "Functional Grammar and Systemic Functional Grammar: A Preliminary Comparison," *Working Papers in Functional Grammar*. [39]. Amsterdam: University of Amsterdam.

Danlos, L. (1987) *The Linguistic Basis of Text Generation*. Cambridge: Cambridge University Press.

Davidse, K. (1985) "M.A.K. Halliday's Functional Grammar and the Prague School," Dirven, R. & Fried, V. Eds. *Functionalism in Linguistics*. Amsterdam: Benjamins, pp. 39-79.

Fawcett, R.P. (1980) *Cognitive Linguistics and Social Interaction. Towards an Integrated Model of a Systemic Functional Grammar and the Other Components of a Communicating Mind*. Heidelberg: Julius Gross.

Halliday, M.A.K. (1961) "Categories of the Theory of Grammar," *Word*, 17, pp. 241-292.

Halliday, M.A.K. (1985) *An Introduction to Functional Grammar*. London: Edward Arnold.

Halliday, M.A.K. (1991) "Towards Probabilistic Interpretations," Ventola, E. Ed., pp. 39-6

Halliday, M.A.K. & R.P. Fawcett (Eds.) (1987) *New Developments in Systemic Linguistics*, [I]. London:Frances Pinter.

Halliday, M.A.K. & R. Hasan (1989) *Language, Context and Text: Aspects of Language in a Social-Semiotic Perspective*. London: Oxford University Press.

Hans, P. Ed. (1979) *The Collins English Dictionary*. Glasgow.

Hudson, R.A. (1974) "Systemic Generative Grammar," *Linguistics*, 139, pp. 5-42.

Lemke, J. L. (1991) "Text Production and Dynamic Text Semantics," Ventola, E. Ed., pp. 23-38.

Martin, J.R. (1987) "The Meaning of Features in Systemic Linguistics," Halliday, M.A.K. & R.P. Fawcett Eds., pp. 14-41.

Matthews, P. (1965) "Review of Dixon, R.M.W. *Linguistic Science and Logic," Journal of Linguistics*, 1, pp. 61-68.

Neufeldt, V. Ed. (1988) *Webster's New World Dictionary*. New York.

Patten, T. (1988) *Systemic Text Generation as Problem Solving*. Cambridge: Cambridge University Press.

Ventola, E. Ed. (1991) *Functional and Systemic Linguistics: Approaches and Uses*. Berlin: Mouton de Gruyter.

Ziff, P. (1969) "Natural and Formal Languages," Hook, S. Ed. *Language and Philosophy*. New York: New York University Press, pp. 223-240.