From: Carys Jones and Eija Ventola (Eds) [In Press] *New Developments in the Study of Ideational Meaning: Form Language to Multimodality*. London: Continuum Publishers.

Chapter 3

A survey of process type classification over difficult cases

Mick O'Donnell, Michele Zappavigna, Casey Whitelaw

Abstract

Within SFL, there are often discussions as to how particular clauses should be coded in regard to their process type. In order to better understand how Systemicists differ in how they analyse, we conducted an online survey where we asked practitioners to select the process type of 32 clauses, most of the instances offering some difficulty. In this chapter, we will explore some of the results of this survey, in particular to work towards understanding the underlying coding criteria that are used in the community, but also to understand how our community in reality consists of sub-communities of coding practices.

1 Introduction

One distinct aspect of Systemic Functional Linguistics (SFL) is the analysis of clauses in terms of *process types*. According to the theory, the grammar provides a number of schemas for packaging information into a clause. For instance, *material* clauses consist of an Actor, a Process and a Goal, while *mental* clauses contain a Sensor, a Process and a Phenomenon. Each of these schemas corresponds to a *process type*. Normally, six process types are identified: material, behavioural, verbal, mental, relational and existential.

Process type analysis was first described in Halliday (1976), which stemmed from Halliday's attempt to develop the kind of grammar which would support teachers in teaching language. A fuller, more evolved description was given in *Introduction to Functional Grammar* (Halliday 1985, henceforth IFG), and its two later editions: Halliday (1994) and Halliday and Matthiessen (2004). However, these texts proved too technical for beginners, and easier introductions were introduced (e.g. Downing and Locke 1992; Eggins 1994; Bloor and Bloor 1995; Butt et al 1995; Thompson 1996; Droga and Humphrey 2002). Apart from these works which more directly follow IFG, various works have offered alternative descriptions of process types (e.g. Fawcett 1980; Morley 2000; Neale 2002).

In each case however, the authors differ somewhat in the criteria used to classify process types, often to handle cases not covered by IFG, but sometimes because the author's interpretation differs. Because of this diversity of descriptions, SFL does not provide a single process type classification of any clause. The classification a coder makes rather depends on the model being employed. This situation has been confirmed by ongoing discussions on the Systemic discussion lists, *Sysfling*[1] and *Sysfunc*[2]. Difficult cases have been presented to the lists and rich discussions have followed, revealing a range of coding practices existing within the Systemic community.

The aim of this chapter is to explore how 'standard' is the application of process type analysis: to what degree does the community of practitioners concur in their application of the theory. To this end, we prepared 32 clauses, ranging from those reasonably clear as to process type to those which are very difficult to code. We placed these examples on a web-page, which allowed a coder to assign a process type to each clause, and invited the coder to comment on their reasoning. We then solicited readers of Sysfling and Sys-func to complete the coding. 75 people responded.

The rest of this chapter will use the results of this study to explore the variation in the SFL community. We will look at the data from two directions:

1 In terms of clauses: What range of responses did each clause evoke, and what criteria motivated each coding? Does the community in fact vary in the criteria they use?
2 In terms of coders: Can we see patterns in the ways different people code? Do they form sub-communities within the greater SFL community? Can we talk about 'dialects' of SFL, as often suggested by Robin Fawcett?

Section 2 below will describe in more detail the data collection process. Section 3 will discuss some of the clauses in the study, and how the community coded them, while Section 4 will explore the community structure of the coders. Section 5 will draw conclusions from the findings.

2 Data Collection

The methodology for data collection and cleansing is described in this section.

Selection of sentences. Subscribers to the Sysfling mailing list regularly submit clauses that they have difficulty analysing in terms of process type, for general discussion. We have collected some of these over several years. There was also a 'Tough Clause Workshop' at Sydney University Linguistics Department, (04/04/2003) analysing difficult clauses, and we obtained a list of clauses from the organiser, Geoff Williams. To this list, we added some more easily coded clauses, with the intention that these would allow us to verify that at least those examples were commonly coded (and if not, they would allow us to identify coders who were very different from the rest).

The examples discussed on Sysfling were often given out of their textual context, and we were unsure of any confidentiality restrictions on their use. For these reasons, we decided not to use these examples, but rather searched the web for similar cases and used these instead. Our final set included 32 sentences, most of them on the difficult side.

Web Survey. We then constructed a web-page which presented each clause and its textual context, and allowed the coder to select one of material, behavioural, mental, verbal, relational or existential. There was also space for the coder to type a comment on why they had chosen that option, or what other options appealed. On clicking the "Submit" button at the bottom of the page, the coder's responses were stored in a file online. At the end of the survey, we have had 75 responses.

Eliminating Coders. A small number of coders gave completely different responses to those of other coders. In some cases, this is because they were new to the field, but there were also cases where the coder had an individualistic approach to process types. For Section 3 however, we sought some notion of 'typical' response patterns. So, for that study, the seven most unconventional coders were eliminated. (See Section 4.1 for our study of coder nonconformity which informed these exclusions.)

Eliminating Clause One clause was excluded because our instructions of which verb to code were not clear. In *Escape, he thinks, is Jewish*, 37 coded "think" and 25 coded

"is". Given that the responses in this case did not reflect a particular coding approach, but rather a decision as to what we wanted coded, we excluded this case from the study.

3 Analysis of Clauses

Given limited space, we have chosen to focus only on those clauses which shed light on the coding of verbal processes. We first explore the confusion between behavioural and verbal processes, and then explore some other aspects of verbal processes.

3.1 Behavioural Clauses

According to IFG, behavioural processes are "physiological and psychological behaviour, like breathing, dreaming, smiling, coughing." (Halliday 1985:128). These processes are often a source of confusion, because they border on other processes: they are similar to material processes in that they can include physical manifestation (e.g., *cough*, *dance*); they usually include the physical manifestation of verbal processes (e.g., *talk, yell*); and the physical manifestation of mental processes (*look, listen, worry*, etc.) and mental states (*cry*, *laugh*, *smile*).

There is some difference in the literature as to exactly what does belong in this category. Thompson (1996:100) notes that verbs Halliday includes such as *"dance*" and *"sing"* might just as well be classed as material.

Below we discuss three of the clauses which seem best to fit this category.

(1) *I laughed at that.*

Context: *Until they can find you a council flat. 'But the days of them are gone.' 'There are still some going and you're bound to get preferential treatment of some sort.' 'Why? Because I've had to spend a couple of nights in a porchway?' 'Because you're a genuine case.' I laughed at that. 'The hostels of London are stuffed full of genuine cases. Why pick on me?*
Situation: conceptually a bodily reaction manifesting a mental reaction to some phenomenon.
Result: behavioural: 63, mental: 3, material 2.

Explaining coding differences: this is a very prototypical behavioural clause; *laughing* is often used as an example of behavioural clauses. However, three coders coded it as mental, suggesting that, for them, the fact that the laughter is expressing a mental reaction is most important. Two coders selected material, and these coders did not use the behavioural category for any clauses in the study, suggesting that behaviourals are not part of their process type model. Both coders in fact commented that they tend to subsume 'behavioural' processes within 'material'.

(2) *We talked for hours*

Context: *Later, in the café, we put the flag into the salt cellar and waited. We talked for hours. Then we went back to our room, and Jim played the guitar, and I sang.*
Situation: the situation expressed by the clause involves verbal action, although the lexico-grammatical expression is not a projecting clause.
Result: behavioural: 40, verbal: 24, material: 4

Explaining coding differences: this clause is less clearly behavioural, as it is expressing a situation in which verbal action is taking place. However, the

majority decision here, with 40 coders, is that this is behavioural. IFG mentions 'talk' as a behavioural of the 'near-verbal' kind.

24 of the coders chose to code it as *verbal*. These coders seem to look at the conceptual situation as their primary criteria when coding clauses, and since there is underlying verbal action, code it as such. As one coder commented:

> I define verbal process as a process of communication. I do not accept that (potential) projection is necessary.

The majority pay attention to its grammatical form (there is no projection), and coded as behavioural. They probably choose behavioural rather than material because it is bodily action without any indicated change of state. However, four coders did choose material, but two of these coders, as mentioned above, always code behavioural-like processes as material.

> *(3) and talked about his hometown of Motown*
>
> **Context**: *The talented junior sat down with Samantha Kilgore <u>and talked about his hometown of Motown</u> knowing what Coach expects and what its like to be Rick in this edition of Tiger Q&A.*
> **Situation**: this clause is similar to the previous clause, except that there is a circumstance of Matter which seems to be realising the Verbiage.
> **Results**: verbal: 36, behavioural: 27, material: 5.

Explaining coding differences: the presence of the circumstance of Matter swayed 12 coders away from the behavioural interpretation over to a verbal one. This was possibly because the 'about' circumstance could be taken as a type of Projection. As one coder commented:

> The presence of Matter clearly pushes it more towards verbal: there would certainly be a case for saying that 'talking' (also 'chatting', etc.) is behavioural, whereas 'talking about' is verbal.

Those who stayed with *behavioural* often commented on the lack of potential for projection. One person who coded the previous example as behavioural coded this one as material. This could be due to coder irregularity.

Part of the confusion here may stem from the confusion in Halliday and Matthiessen (2004). On page 251, "grumbled about the food" is said to be behavioural. However, on the next page, the following sentence containing "talk about" is said to be verbal: *Chiruma would find any opportunity to <u>talk</u> to that priest <u>about</u> Kukal*. This work was largely a revision of Halliday (1994) by Matthiessen, and we guess that this is one place where the coding difference between Halliday and Matthiessen becomes apparent (Matthiessen is more conceptual in his coding).

To summarise the results for sentences (1)-(3), we can see that there is a cline from hard behaviourals (e.g., *laughing*) to hard verbals (e.g., to say something). Coders fall into one of three groups: the first code *laugh*, *talk*, and *talk about* as behavioural. Group 2 follows group 1, except that they code *talk about* as verbal. Group 3 code any form of *talk* as verbal. See Table 3.1.

Table 3.1 The Behavioural-Verbal cline

	Group 1	Group 2	Group 3
He laughed	Behavioural		
We talked	Behavioural		Verbal
We talked about...	Behavioural	Verbal	
He said...	Verbal		

Here we have explored only the behavioural-verbal cline, but similar clines would probably be observed for near-mentals, and near-materials, with a gradual shift of codings from behavioural to the other category. For instance, in relation to the near-mentals: *I was thinking all day > I was thinking about the weather > I was thinking that I should go*. A brief survey of participants in a workshop supported this hypothesis.

3.2 Verbal Clauses

Verbal processes involve a communication between a Sayer and an Addressee, where some message, the Verbiage, is communicated. The study showed clearly that there is no general agreement as to which clauses fit this description. For some coders, there needs to be an actual presence of grammatical projection (e.g., *He said that he was going; She said to eat*). For others, it is enough that the underlying situation is one of verbal communication, as in the examples above involving 'talk'.

For those who use grammatical criteria, people follow different criteria as to what passes for projection. For some, an "about" adjunct is sufficient, for others, it needs to be a clausal projection. A middle case involves the use of a nominalised report, as in *he asked a question*. This case was not included in our study, but it would be interesting to see the distribution of codings for this example (Halliday and Matthiessen 2004 classes this as verbal, as does Thompson 1996).

Below we consider some cases, some clear, some difficult, which shed light on the criteria people are using to code. The first case, (4), is prototypically verbal. Those which follow are more peripherally verbal.

(4) 'There is nothing I can do,' <u>said</u> the King.

Context: *Despite being repeatedly arrested and brought back home, bruised and beaten by the police, she finally got to see the King by throwing herself into the road in front of the Royal car. 'There is nothing I can do', <u>said the King</u>, as she was dragged away.*
Situation: conceptually verbal and expressed congruently in the grammar as a projecting clause.
Result: All 68 coders coded as verbal.

Explaining coding differences: since both conceptual and grammatical criteria lead to the same coding decision, there was no dissention here.

(5) 'Dear father, thank you so much,' her smile <u>said</u>.

Context: *She smiled at him and then, once again, at her father from whose authority these vows released her. ' Dear father, thank you so much,' <u>her smile said</u> - for he had loved and cherished her most tenderly, claiming her obedience as his right, his due.*

Situation: this seems to present an act of communication, although it is not clear whether the communication was intentional. If an intentional communication, it is through a non-verbal channel. The situation might also be interpreted as one of body expression of internal mental states, and thus behavioural. Syntactically, the expression is a projecting clause, using a verb typically used for expressing verbal processes.
Results: verbal: 50, relational: 10, behavioural 5, mental: 3

Explaining coding differences:
verbal: the majority of coders nominated this as a verbal clause. On one level, this is understandable, because the verb "say" is a prototypical one for verbal processes, and there is projection in the clause. There are however three potential problems in this example which need to be addressed:
a) *Non-human Sayer:* it is not the woman who is saying, but her smile. However, Halliday says explicitly that non-human agents can be Sayer:

> The Sayer can be anything that puts out a signal, like the notice or my watch; cf. the light in the light says stop, the guidebook in the guidebook tells you where everything is. (IFG, p140)

b) Intentionality: for some, communication must be intentional before it can be classified as semiotic, and thus verbal. In this case, it is not clear whether the woman intended to communicate the message, or whether the message is just the interpretation of the observer.
c) Nonverbal communication: some may query whether nonverbal communications should be classified as verbal processes. For example, is *He waved goodbye* a verbal process?
Most coders seem quite happy to accept a non-human sayer, that the message was intentional, and that verbal processes include nonverbal channels. It is also possible that some coders are purely responding to the grammatical container. Some commented that the clause is conceptually relational realised in a verbal container, but most of these coded on the container anyway.
relational: 11 coders chose to ignore the grammatical form, looking more at the conceptual action, which is one of something "meaning" or "indicating" something, and thus coded relational.
Some who coded relational did so because of the issue of intentionality. One coder indicated that if there had been some indication of intentionality in the process, they may have swayed to verbal.
behavioural: there were also 5 cases of behavioural coded here, which perhaps relate to a bodily expression of mental states, (what IFG calls "near mental").
mental: of the few mental coders, one commented that the clause could be viewed in terms of the perceiver, i.e., "he interpreted her smile to mean ...".

(6) A final line of analysis insists that the government has made little difference

Context: *A third interpretation is to say that the strategy has not been implemented and therefore the government is not fully responsible for the outcome. A final line of analysis insists that the government has made little difference, particularly on unemployment.*
Situation: 'insisting' is usually carried out via a verbal channel, but also connotates some element of persistence, which might be mental. The agent of the insisting is non-human. The non-human agent is an abstract object, an "analysis", which could be seen as the product of a process of analysing, which indirectly could make the

people who make the analysis the hidden agents of the insisting. Grammatically, this is a projecting structure.
Results: verbal: 50, relational: 9, material: 5, mental: 3, existential: 1

Explaining coding differences:
verbal: the vast majority coded verbal, as suggested by the presence of a projection. It seems that, as with the prior example, the non-human nature of the Sayer was not a problem, allowing a message-container to be the (personified) Sayer (e.g., *the message said to come*). One coder took "line of analysis" as the Instrument of the process rather than the Sayer, with Sayer left unmentioned.

One argument against the verbal coding is that 'insist' cannot take an addressee, although other verbal processes such as 'demand' are similar in this regard. Another argument against a verbal coding is that the tense, simple present, places this as either a continuing action, or as a recurring action. Both are less likely for a verbal process, but more likely for mental or relational interpretations (see below).
relational: some coders took 'insists' to be similar to 'shows', 'means', 'points to', etc. and the process would thus be relational. One comment was that 'insists' is like 'means' but with a connotation of forcefulness. The simple present tense is the unmarked form for such relationals.
material: it is not clear what criteria lead to this coding. One of the 5 coders in this class commented that:

> in other contexts 'insist' could be verbal, but not in this case where the doer (a final line of analysis) is non-human.

One comment provided a possible criterion for the material coding: *if nothing else fits, code as material.*
mental: it is possible these coders took 'insist' as a form of 'believe', with 'line of analysis' symbolically representing those people who make that analysis. The simple present tense is appropriate for stating permanence of belief.

(7) They instruct people how to take binding directives

Context: "*Three theses were presented as part of an explanation of the concept of authority. They are supposed to advance our understanding of the concept by showing how authoritative action plays a special role in people's practical reasoning. But the theses are also normative ones. They instruct people how to take binding directives, and when to acknowledge that they are binding."*
Situation: the conceptual situation is again one where we have a message-carrying object functioning as Agent. The 'theses' are probably intended in the 'argument' sense rather than in the 'book' sense. The process of 'instruction' is one which is conceptually complex, typically involving verbal actions (*talking*, *writing*), but also possibly non-verbal elements (material demonstrations, etc.). The grammatical form is of a projecting clause, parallel with more clearly verbal clauses such as *He told her how to get there*.
Results: verbal: 41, material: 19, relational: 4, mental: 2, behavioural: 1

Explaining coding differences:
verbal: this was the majority decision. These respondents focused on the fact that the instructing agent is an abstract message, personified as the Sayer (or perhaps as the Instrument of the saying). The grammatical form (projecting) supports this decision. Halliday (1985:146) gives examples of 'explain' and 'show' as verbal processes, and 'instruct' is not too different. Some coders commented that while instruction involves

material and mental aspects, they coded as verbal because in this case it happens through the medium of words.

material: we think the complex nature of the instruction process (including verbal elements but not entirely) push some people to code this as material. If the context of the clause suggested a more material setting for instruction (e.g., a classroom), then more coders would possibly have taken this path. Several of the comments on this clause suggest that the coders were torn between verbal and material, showing the tension in coding criteria here.

relational: four coders took this path. We are not sure why. One coder commented:

> I don't know!! teach and train and instruct always seem to be tricky, even with human agents, let alone abstract 'teachers' in the form of texts... the general semantic field seems to be that of cognition, understanding, etc. but there is also a relational aspect to this in the 'showing/revealing' of knowledge.

mental: the situation of instruction does involve cognitive processing in the instructed person. One coder glossed "instruct" as "cause people to know how to". However, only two coders chose this option. As the comment above states "the general semantic field seems to be that of cognition, understanding, etc."

3.3 The coding community

In general, there seems to be a general spread of coding practices for most of the clauses discussed here. However, note that most of these clauses were selected exactly because they had given coders problems. Several means of coding sentences are evident, which will be discussed below.

Firstly, the comments and coding patterns of individuals suggests that some coders depend primarily on the syntactic structure of the clause: if there is projection, or potential for projection, then it is a mental or verbal clause, and the choice of the verb decides between these options.

A second set of coders use conceptual criteria: they decide what underlying action is being represented by the clause, and code on that basis. For instance, if the situation expressed by the clause involves verbal action, then code verbal. This approach is problematical where the represented situation contains elements of different process types, e.g., teaching can involve elements of material, verbal and mental activity.

For difficult cases, some coders rely on the paraphrase test: rephrasing to a less difficult wording, and coding as that clause is coded. This approach has been common in discussions on Sysfling, but Martin at least finds it problematic:

> Paraphrase (having the same truth value in some possible world) is a graveyard for transitivity analysis, as case grammar and most other approaches have revealed... and seems to be creeping into systemic descriptions. (Jim Martin, Sysfling email, 2001)

In response, John Haynes asked whether this put into question Halliday's notion of grammatical metaphor: the unpacking of a metaphor is in a sense a paraphrase. Our guess is that those who code on conceptual criteria can use paraphrase without too much danger, while for those who code grammatically, it is totally perilous.

Another criterion for coding is to defer to authority: to check how the textbooks coded similar examples. The problem with this approach is that we end up coding as we do because "that is the way things are done". What we really need to develop are explicit rationales for determining how each and every clause should be coded. As a community, we need to develop explicit statements of coding criteria, detailing how

to determine a clause's type, in a manner similar to that used to classify plants into families (e.g., does it produce flowers or not? Does it have 1 or 2 seed leaves? etc.).

Note that this does not entail the need for a unified coding practice. As Halliday said in "Syntax and the consumer" (Halliday 1964), the way you describe language depends on how you intend to apply the description, and as we as a community are exploring many different applications, differences in coding practice are bound to occur. However, it would be useful if there existed a number of explicit coding guides, to facilitate students learning how to code, and also to provide a basis for discussion as to what criteria should be used.

4 Are there coding dialects?

In the previous section, we looked at a number of *clauses*, and analysed how coders responded to it. Here, we will instead look at each *coder*, and explore how they relate to other coders. We will first explore a measure which attempt to show how (non)conformist each coder is in relation to the other coders. Then we will explore whether the coders can be grouped together into coding *dialects*: where a set of coders follow a particular common set of coding criteria.

Robin Fawcett often talks of 'dialects' of SFL, distinguishing the 'Sydney dialect' (after Halliday, etc.) from the 'Cardiff dialect', (after Fawcett), and other dialects (although he attributes the concept to Halliday himself). In this section, we wish to explore whether the responses to our survey support the idea that there are such dialects in the SFL community.

The term 'dialect' suggests that there are groupings of individuals with common practices. The practices do not have to be exactly identical across all individuals in the grouping, but each individual should exhibit a sufficient number of the group's common practices for that individual to be recognised as part of the group.

We can define a *coding dialect* as a common set of criteria to use in the making of grammatical decisions, e.g., in the present case, as to what process type a given clause represents.

One of our goals is to identify if the data of our study supports the hypothesis that there are actually distinct coding dialects, i.e., are their groupings of coders which tend to code a given set of clauses in the same way.

Unfortunately, the study included only one contributor from the 'Cardiff dialect'. The setup of the study itself biased against this group, as it assumed six primary process types, while the Cardiff grammar uses a different set of labels, and more of them. Our study thus really explores the dialectal variation within the Hallidayan strand of the school.

4.1 Measuring coder conformity

In any population of coders, there will be those who conform to a standard (conformists), and those who follow their own rules (individualists or nonconformists). It will help us to identify our dialects (standards) if we first eliminate all of those coders who do not belong to any dialect, whose responses are on the whole nonconformist.

We thus introduce a formula which allows us to rate each coder according to the degree to which their set of responses conform with other coders – their *conformity level*. We calculate conformity as follows:

We firstly calculate the probability of each response for each of the sentences (e.g., the probability of coding "we talked for hours" as verbal). The formula here is simple:

$$\text{Prob}(Rij) = \frac{\text{Count of response i to clause j}}{\text{Number of responses to clause j}}$$

This provides a number between 0.0 and 1.0, with 1.0 representing 100%. The probabilities of all responses to a given clause will sum to 1.0.

For each coder, we sum the probabilities of each of their responses, and then divide by the number of clauses they responded to. This gives us the *average response probability* (ARP) for the coder.

We can then rank all coders in terms of their ARP. The higher their ARP, the more conformist they are (their choices follow the general population more frequently than those with lower ARPs).

This approach does favour those who select the most popular response. However, coders who sometimes code the second or third most popular response may not necessarily come out as nonconformist. If the second-most popular choice for a clause is close in popularity to the favourite, then the difference in conformity ratings for that item will not be great. Only coders who regularly have few agreements with other coders will have low conformity ratings.

There is nothing good or bad about being 'conformist' or 'nonconformist' in regard to coding. It is just that for our purposes (modelling what it is that conformists are conforming to), we need to identify those who are not conforming to any dialect, so that they can be put aside for this study.

In preparation for the next stage, grouping users into dialects, we put aside the data for the 6 least conformist coders, since we wish to measure what people are conforming to.

Using the above measure, it is possible that a small set of coders with common coding practices (a coding dialect) all come out as nonconformists, and would thus be eliminated. A second means of identifying nonconformist coders was developed, which avoided this problem. Due to lack of space, no details can be given, but we note that the same 6 coders come out bottom using either approach.

4.2 Grouping users

The technique for grouping used here does not at first try to group coders. Rather, it makes the assumption that each coder approximates some ideal coder (a 'coding model'), and varies from that model because of errors, or idiosyncratic decisions on particular clauses. We then examine the data with the aim of positing a set of coding models which best explains the actual coders.

In the simplest case, we assume there are two coding models, and each coder approximates either one or the other. We assign each coder to the model which they are closest to in their choices. We measure the number of coder responses which differ from the model. A coding model is another term for the 'dialects' we mentioned earlier.

Ideally, we seek two coding models such that, over all coders, there are as few differences from the coder's closest model as possible. We call each case where a coder's response does not agree with their assigned coding model an 'unexplained coding'. We can thus rephrase our goal as seeking to minimise the sum of unexplained codings over all coders

Rather than two models, we might instead assume three or four, etc. In these cases, each model represents a distinct set of responses to the clauses to be coded.

Around each model will cluster a number of coders, in that the coders are closer to that model than they are to any other model.

The problem thus becomes: how to locate the ideal coding positions which together minimise the unexplained codings. Our solution was to use a simple hill-climbing method, as described below. Note however, that such a hill-climbing method suffers from the problem of sometimes discovering a local minima, rather than a globally best solution. We leave it to later work to improve on this approach.

Step 1: Initiation:

a) Start with a single model, which is arbitrarily given as the most popular coding of each clause. For each coder, we measure the number of cases where the coder differs from the model, and sum these together to get the Total Unexplained Codings (TUC).
b) Introduce a second model, identical to the first.
c) Search for a single change in this second coding model which will improve the TUC: for instance, changing the coding of the fifth sentence from material to mental might decrease the TUC by 5.
d) Take the change which caused the highest reduction in TUC, and actually make that change to the second model. In case of a tie, the first found is used.

Step 2: Iteration: Repeat steps 1c/1d but this time looking for the best change in *either* model. Repeat this process until no single change in either model decreases the TUC.

4.2.1 Two model results

Assuming a single model (to which all coders are assigned), we have a TUC (Total Unexplained Codings) of 493. For this sub-study, we included 60 coders, and 31 clauses per coder, giving 1860 responses. As 493 are unexplained, this means the model explains 1367 responses, or 73.5%.

By introducing a second model, the number of unexplained codings was reduced to 438 (55 more codings are explained), which means 1422 are explained, or 76.5%. The users were almost equally divided between the two models.

The next question is, are there qualities of the models that we can identify? In 24 of the sentences, both models agree. However, they differ in regards to the coding of 7 sentences. See Table 3.2.

Table 3.2 Differences between coder 'dialects': the 2-model case

Text	**Model 1**	**Model 2**
the letter draws attention to the arrest ...	material	verbal
A roar greeted his effort at authority	material	verbal
We talked for hours.	behavioural	verbal
and talked about his hometown of Motown	behavioural	verbal
If you've gotta count the sheep	material	mental
The connoisseurship demonstrated in these two examples is built up from an accumulation of work by many scholars	relational	material
the Hamadryas makes do with only a handful	relational	material

Model 1 prefers material/behavioural codings in some cases where Model 2 codes verbal (and in one case mental). In all of these cases, the clause is not a projecting clause, but does to some degree involve conceptually a verbal/mental element. It seems then that to followers of Model 1, syntactic criteria are important (no projection means it cannot be verbal/material), while to followers of Model 2, it is the conceptual structure that is important.

The other difference between the models is that the coders of Model 1 code two complex cases as relational, while the Model 2 coders treat them as material. It is difficult to understand why, but i) it is good that these two similar cases are treated identically within the models, and ii) these cases are so difficult to code that they may represent noise to some degree, the responses not representing the coder's underlying model, but rather a random response where their understanding fails them.

4.2.2 Three model results

Assuming three models results in a reduction of the TUC from 493 to 415 (77.7% of the responses fit the models). Model 1 was by far the most popular, with twice as many members as either other model.

Table 3.3 Differences between coder 'dialects': the 3-model case

	Text	**Model 1**	**Model 2**	**Model 3**
1	*the letter draws attention to the arrest ...*	material	material	verbal
2	*A roar greeted his effort at authority*	material	verbal	verbal
3	*We talked for hours.*	behavioural	verbal	verbal
4	*and talked about his hometown of Motown*	behavioural	verbal	verbal
5	*If you've gotta count the sheep*	material	verbal	mental
6	*The connoisseurship demonstrated in these two examples is built up from an accumulation of work by many scholars*	relational	relational	material
7	*the Hamadryas makes do with only a handful*	relational	material	material
8	*When we come to god through Jesus Christ*	material	material	mental
9	*Nanny stood on the bridge that spanned the ornamental lake*	material	behavioural	material

Table 3.3 shows the differences between the three models. Model 1 and 3 are basically identical to the models in the 2-model case. Model 2 falls somewhere between these cases, sometimes siding with Model 1, sometimes with Model 3. Only in two cases (5 and 9) does it represent a distinct opinion. It agrees with Model 3 in those cases which are clearly conceptually verbal (*talking*, *roaring*), but sides with Model 1 in less clear cases (*coming to God*, *drawing attention*).

4.3 Interpretation

The goal of this section was to explore the issue of whether coding dialects exist within the community. To this end, we firstly applied techniques to identify those coders who were least conformist, and put these coders aside, as it is the conformists we are trying to categorise.

Secondly, we applied a technique to find the optimal division of our coders into sub-groups, firstly trying a 2-group split, and then a 3-group split. This grouping offered strong support for the hypothesis which we derived in Section 2, that some coders use conceptual criteria to code, and others code on grammatical ground. The sentences which divide the groups are largely those which conceptually involve mental or verbal action, but syntactically do not involve projection.

However, on examining the locality of the coders in each group, there does not seem to be any geographical commonality within the groups. In some cases, teacher and student are in different groups, and each group is equally seeded with the big names. We thus conclude from this that, at least within the Hallidayan tradition (the Cardiff tradition was not represented enough to measure), there are not geographically defined 'dialects' of SFL. Rather, within each locale, each practitioner chooses for themselves from the variety of criteria which are available in the community as a whole.

5 Conclusions

Both our analysis of individual clauses (Section 3) and of the grouping of coders (Section 4) show that the divide between using conceptual vs. syntactic criteria is widespread throughout the community as a whole, and each individual chooses which path they follow. This is, we believe, the result of the lack of explicit coding criteria in general, and argue that what the community needs is explicitly stated sets of criteria for coding practices, and perhaps distinct criteria descriptions for particular applications. To a degree, this has already started: for instance, Robin Fawcett in Cardiff has developed coding criteria for his grammar, although these have not been published.

As a community, we need to develop our criteria, get them published, then use these published criteria as a basis for general discussion as to what criteria are valid (for particular applications), what problems they may have, etc. Only in this way can we avoid the claim of those outside SFL that our practice does not meet the scientific requirement of 'repeatability': any two coders should produce the same analysis of the same text.

This chapter has only addressed one area of SFL analysis, transitivity, but we believe, similar conclusions would be reached for the analysis of multiple coder surveys of other areas of analysis, such as theme.

Notes

1 http://www.isfla.org/Systemics/Contact/Sysfling.html
2 http://listserv.uts.edu.au/mailman/listinfo/sys-func

References

Bloor, T and Bloor, M 1995. *The Functional Analysis of English: a Hallidayan Approach*. London: Arnold.

Butt, D., Fahey R., Spinks, S., and Yallop, C. 1995 *Using Functional Grammar: an Explorer's Guide*. NCELTR, Macquarie University.

Downing, A. and Locke, P. 1992. *A University Course in English Grammar*. New York: Prentice Hall.

Droga, L. and Humphrey, S. 2002. *Getting Started with Functional Grammar*. Sydney: Target Texts.

Eggins, S. 1994. *An Introduction to Systemic Functional Linguistics*. London: Frances Pinter.

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 Groos and Exeter University.

Halliday, M. A. K. 1964. "Syntax and the consumer". In C. Stuart (ed.), *Report of the Fifteenth Annual Round Table Meeting on Linguistics and Language Study*. (Monograph Series in Languages and Linguistics 17.) Washington, D.C.: Georgetown University Press.

Halliday, M. A. K. 1976. "Types of process". In G. Kress (Ed.), *Halliday: System and Function in Language*. Oxford: Oxford University Press.

Halliday, M. A. K. 1985. *Introduction to Functional Grammar*, 1st Edition. London: Edward Arnold.

Halliday, M. A. K. 1994. *Introduction to Functional Grammar*, 2nd Edition, London: Edward Arnold.

Halliday, M. A. K. and M. I. M. Matthiessen, C. M. I. M. 2004. *Introduction to Functional Grammar*, 3rd Edition, London: Edward Arnold.

Morley, G. D. 2000 *Syntax in functional grammar: an introduction to lexicogrammar in systemic linguistics*. London and New York: Continuum.
Neale, A. 2002 'More Delicate TRANSITIVITY: Extending the PROCESS TYPE system networks for English to include full semantic classifications'. PhD thesis. Cardiff: School of English, Communication and Philosophy, Cardiff University.
Thompson, G. 1996 *Introducing Functional Grammar*. Arnold.