

## RELATIONAL CLAUSES IN ENGLISH TECHNICAL DISCOURSE: PATTERNS OF VERB CHOICE

Arlene Harvey

### Abstract

This paper reports on patterns of verb choice in identifying relational clauses (e.g. 'X is Y, Y is X') in English technical manuals. While it is obvious that specific lexical verbs will feature in identifying clauses of different functions, e.g. *mean* (defining), *call* (naming), *exemplify* (exemplifying), less transparent is the distribution of these more specific verbs and the general or neutral verb *be*. The findings suggest that verb choice in (technical) identifying clauses is strongly associated with the degree of equivalence constructed between the two central nominal groups in the clause (the Token and Value). Equivalence relations are one-to-one (rather than one-to-many) and exhaustive (rather than semantically open). Major grammatical influences on equivalence include nominal group structure, ergativity of the clause, and the inclusion of features (e.g. interpersonal, logical or textual) that undermine the privileging of an experientially homogeneous world-view. The results challenge the notions that *be* and specific verbs are interchangeable and that *be* is an unmarked choice. On the contrary, the data reveal that under certain conditions *be* is the more marked choice. The results have practical implications for teachers and students of English (in particular, students of English for Academic and/or Specific Purposes) as well as translators.

**Keywords:** English technical discourse, Equative clauses, Relational verbs, Definitions, Markedness

### 1. Introduction

The English verb *be* can function as an auxiliary in the verb group, specifying tense (e.g. *Security violations are increasing*) or voice (e.g. *The data center is being accessed illegally*).<sup>1</sup> Alternatively, *be* can function as the main verb of the clause, signaling a relation of some kind, either attributive (copular) (e.g. *The user password is secure*), equative (e.g. *The dataset manager is also the dataset owner*) or existential (e.g. *There is a problem*). Equative clauses, also referred to as identifying clauses (Halliday 1994 *cf.* Halliday 1967, 1968), are the main focus of this paper. Identifying clauses are a subtype of relational clause in which one participant in the clause, expressed in a nominal group (NG), serves to fix the identity of a second participant, also expressed in a nominal group, as in (1):

---

<sup>1</sup> This research originated in a collaborative project between Sydney University and Fujitsu Australia Ltd. I would like to express my thanks to Christian Matthiessen, Jane Simpson and Guenter Plum for comments on earlier versions of this work. I am indebted to Martina Temmerman for translating parts of her PhD thesis from Dutch into English for me and to the anonymous reviewer who offered constructive and insightful comments on an earlier version of this paper.

- (1) The manager (participant NG) is (identifying relational verb) the person who manages user access to data (participant NG).

Identifying clauses can be distinguished from the copular or 'attributive' type of relational clauses in that the former contain two co-referential nominal groups that can be reversed without any substantive change in experiential meaning. Thus:

- (2) The person who manages user access to data (participant NG) is (identifying relational verb) the manager (participant NG).

Relational clauses, both identifying and attributive, are quite common in formal written registers of English, e.g. in technical, scientific, academic and bureaucratic texts (Halliday & Martin 1993; Lemke 1990a; Wignell et al. 1993); in fact, the proliferation of these clauses is one of the indexical features of formal registers.

Various kinds of relations can be indicated in the clause, such as naming, classifying, defining, or exemplifying, and the relational verb can be either the general verb *be* or more specific lexical verbs, e.g. *mean*, *indicate*, *refer to* (depending, of course, on the specific sub-relation constructed). In general, the distribution of *be* and lexical verbs in English has been little studied. While there have been several studies that have touched on the distribution of *be* and posture verbs (such as *sit*, *stand*, *lie*) in other languages, most notably Dutch (see e.g. the collection of papers in Newman, in prep; Van Oosten 1984), the main focus of such studies has been on the semantic differences between the lexical variants and not on the conditions under which *be* (or its equivalent) might be chosen.

A common assumption seems to be that, in relational clauses at least, *be* is more or less interchangeable with other more specific verbs. Halliday (1994), for instance, notes that in relational clauses (of the intensive 'X is A', rather than the circumstantial or possessive types): "The most typical verb is *be*, and X and A are nominal groups. At the same time, many verbs other than *be* also occur" (1994: 120). These more specific verbs cluster into categories that are either identifying or attributive in function (for instance, identifying clauses contain verbs from the equative classes). Halliday's suggestion that *be* is the most typical verb in relational clauses is echoed in Martin's (1992) argument that *be* is the default, or 'unmarked' choice in identifying clauses; that is, it is the verb chosen in the absence of a reason for selecting otherwise (Halliday 1991a). Unfortunately, while *be* may be the most typical choice, it is also the most ambiguous (Davidse 1996) due to the under-specification of both the intended sub-relation (e.g. naming, defining, classifying) and the direction of the identification, that is, which nominal group is being used to 'fix' the identity of the other. In fact, Halliday (1994), in his discussion of relational verbs, seems to be most concerned with the difficulties that semantic under-specification presents for determining the voice of the clause as either active or passive. The distinction is critical since voice allows the appropriate participant roles to be assigned to the two nominal groups being related (see further Section 4).

Taking Halliday's (1967, 1968, 1994) description of identifying clauses as her starting point, Davidse (1991, 1992a, 1996) has provided a more detailed account of the semiotic function of identifying clauses across different registers of English (in particular, medicine and economics). Like Halliday, she suggests two basic types of identifying clause based on coding orientation: Decoding or encoding. In decoding clauses, the nominal

group being identified is the more familiar and concrete element (as in dictionary definitions, where a term is provided with a gloss: 'An X is a Y that ...'). In encoding clauses, the nominal group being identified is the less familiar and more abstract element (as in naming definitions in which a gloss is being given a technical name: 'A Y that ... is called X') (Davidse 1991, 1992a, 1996; Halliday 1967, 1968, 1994). Although Davidse observes that different subsets of lexical verbs tend to be associated with either the decoding or encoding orientation, she offers no explanation as to why a lexical verb might be preferred over *be* (and vice versa). In her study of defining relational clauses used by students in Dutch-speaking classrooms, Temmerman (1994) moves somewhat closer to an explanation. She describes the frequencies of distribution between *zijn* ('be') and alternative lexical verbs in prototypical and non-prototypical definitions and comments that: "The alternatives for *zijn* are not all mutually interchangeable". Indeed, their appearance depends on whether the clause is in active or passive voice.

It is the purpose of this paper to make more explicit some of the main contextual and grammatical influences on the distribution of *be* and lexical verbs with a defining function in English technical manuals. My argument is that *be* is not always an unmarked choice, but rather, under certain conditions is in fact the more marked choice. The most critical influence on verb choice appears to be the degree of co-referentiality between the two nominal groups, that is, the extent to which they are, grammatically and semantically speaking, 'equivalent' to one another. I have suggested elsewhere (Harvey 1999) that the presence or mere inference of a human agent (the participant who assigns the relation) is critical to the degree of equivalence constructed. Equivalent relations represent a categorical, closed and exhaustive relation between the two nominal groups and are typically expressed in fully agentive encoding clauses containing a lexical verb (in definitions these tend to be 'prototypical' or formal definitions; cf. Temmerman 1994). Non-equivalent relations represent a non-categorical, open-ended or negotiable relation and tend to be expressed in non-agentive decoding clauses containing the verb *be* (these include 'non-prototypical' definitions such as partial definitions, ostensive definitions and relations of synonymy; Temmerman 1994).

Before moving on to a discussion of the theoretical and descriptive framework used in the analysis and the results of the analysis, I will briefly describe the data.

## 2. Data

The data for this study consist of 166 identifying clauses that appear in three in-house technical manuals of a multinational company, and the glossary that accompanied these manuals. While all three manuals outline various aspects of the resource protection function operating in the company at that time (the *Resource Access Control Facility* or *RACF*), they differ in purpose: One manual is primarily descriptive whereas the other two are more concerned with explaining procedures for operating *RACF* (Plum et al. 1992). Although the 166 identifying clauses in the data set have a variety of functions (including defining, naming and exemplifying) the most frequent type is defining, accounting for 136

of the total 166 clauses (82%).<sup>2</sup> Defining in fact often subsumes a number of other functions, for instance, naming, classifying, exemplifying and describing. Several of these are included in the well-documented Aristotelian definitional structure (or ‘prototypical’ definition, Temmerman 1994).

#### **Aristotelian/Prototypical definition:**

X [Name:implicit] (*Definiendum*) is Y [Class] that ... [Descriptive characteristics] (*Definiens*).

Y [Class] that ... [Descriptive characteristics] (*Definiens*) is called X [Name:explicit] (*Definiendum*).

Following Temmerman (1994), a distinction is made here between prototypical defining clauses and non-prototypical clauses. Non-prototypical definitions include only one or several of the above functions (i.e. naming, classifying or describing in isolation, or a combination of two, but not all three, of these). This distinction is crucial in explaining the degree of equivalence constructed, and in turn, the typical verb choice.

### **3. Theoretical and descriptive framework**

The main theoretical and descriptive framework used in this study, systemic functional linguistics (see e.g. Halliday 1978, 1994; Martin 1992), emphasizes the contributions of three areas of meaning that are expressed in clause structures and text. The two primary ‘metafunctions’ are *ideational* (incorporating both experiential and logical meanings) and interpersonal. *Experiential* meaning encompasses what is commonly thought of as the ‘content’ of a clause or text, that which is being talked about, e.g. processes in the world, the participants (both human and non-human) involved, and the circumstances within which the process take place. Process types give speakers the grammatical means to ‘[impose] order on the endless variation and flow of events’ (Halliday 1994: 106), thereby making the communication of experience manageable. Processes can be linked to one another through *logical* connections such as cause-effect, contrast, time and space, condition and so forth. At the same time, language is used to create *interpersonal* relationships between language users, for example, through the negotiation of meaning, the presentation of differing viewpoints, and so forth. Meanwhile, *textual* meanings (as expressed in grammatical systems such as Theme and Given/New) allow ideational and interpersonal meanings to be expressed as information in text (Matthiessen 1995b). Grammatical systems most relevant to the current analysis from the three metafunctional domains are shown in Table 1.

---

<sup>2</sup> This high percentage of defining clauses is not so much an indicator of the relative frequency of definitions in the manuals than a reflection of the focus on definitions in the original study (Harvey 1996).

**Table 1: Grammatical systems realizing metafunctional meanings**

	Ideational		Interpersonal	Textual
	Experiential	Logical		
<b>Grammatical System</b>	<p>TRANSITIVITY Y [Participant 1] is called X [Participant 2].</p> <p>ERGATIVITY Y is called X [Participant: Agent].</p>	<p>CONJUNCTION If ..., X is called Y.</p>	<p>MOOD X is called Y. Is X Y?</p> <p>MODALITY X may/must be called Y.</p>	<p>THEME X [Theme] is called Y.</p> <p>GIVEN/NEW X [Given] is called Y [New].</p>

Systemic-functional theory does not necessarily privilege ideational meaning (i.e. the experiential ‘content’ of a clause or text) over interpersonal or textual meaning. However, describing the clauses in my data as ‘relational’ (rather than ‘interrogative’, ‘declarative’, ‘infinitive’ and so on) suggests that the main focus of the analysis is on experiential meaning. The two grammatical systems that encode experiential meaning at clause level are transitivity and ergativity.

One aspect of relational experiential meaning presented here is especially crucial to the current analysis; this is the distinction between decoding and encoding identifying clauses (Davidse 1992a, 1996; Halliday 1967, 1968, 1988, 1994). An analysis of coding orientation relies on what Davidse refers to as the “janus-headed” experiential structure of the English clause, that is, the incorporation of both transitivity and ergativity structures (Davidse 1992b; Halliday 1994: 167). According to the transitivity model of experience, the clause contains a Process (e.g. material, mental, relational, and so forth) (realized in the verbal group) and one or more Participants (realized in nominal groups) associated with the particular Process type. This core configuration of Process and Participant(s) can be elaborated through attendant Circumstances e.g. manner, cause, time, and so forth (realized in prepositional phrases and adverbs). A description of transitivity enables relational clauses to be differentiated from clauses expressing other types of processes (e.g. material, mental, etc). Ergativity, on the other hand, explains the presence or absence of ‘causality’ in the clause, and in relational clauses, allows us to determine the direction of the identification. Both transitivity and ergativity are required to ascertain the degree of equivalence constructed in the clause, which, in turn, influences verb choice.

Despite the apparent dominance of experiential meaning in relational clauses in technical discourse, it is in fact the interaction between all three areas of meaning – ideational, interpersonal and textual – that accounts for the subtleties and complexities of verb choice in my data (described in later sections).

#### 4. Transitivity – symbolic correlation and identification

The two nominal groups in identifying clauses have dual functional roles. In the first set of functional roles, one participant (the Token) stands in a relation of symbolic correlation to the second participant (the Value) (Halliday 1967, 1968, 1994; Davidse 1996). The Token and Value also enter into a relation of identification, with one participant (the Identifier) fixing the identity of the other participant (the Identified) (Halliday 1968: 190-92; 1994).

**Table 2: Semantic relations between Token and Value in the RACF data<sup>3</sup>**

Function	Token	Value
<b>Defining</b>	TERM RACF	GLOSS is a resource protection function which controls user access to resources located in a data center
<b>Naming</b>	NAME RACF center staff	REFERENT the resource maintenance personnel and auditor, chief manager included
<b>Exemplifying</b>	EXAMPLE  the authority used to reference or update a dataset	SET  an access authority
<b>Classifying</b>	MEMBER the default, generic profile, unregistered dataset batch protection, and protection by the global check function	SET other protection in addition to specific name protection
<b>Describing</b>	FORM either a user-ID or group name	FUNCTION the first qualifier of a dataset name
	HOLDER the one assigned to the DASD dataset profile with the generic name most conforming to the dataset name	STATUS the valid access authority

<sup>3</sup> The ‘describing’ relations in Table 2 (form/function and holder/status) are simultaneously identifying and descriptive in function. Such clauses are in fact intermediate between identifying and attributive (quality) (such as *the access authority is valid*) (Halliday 1994: 128-29; Quirk et al. 1985, Langacker 1987; see also Harvey 1999).

The Token (Tk) and Value (Vl) are related to one another by a process of symbolic representation. One participant, which is abstract/general, is coded as the realization of the other, more concrete/specific, participant (Davidse 1992a: 111). The main semantic relations that link the Token and Value (and that occur in my data) are shown in Table 2

The Token typically (but not necessarily) conflates with the more concrete participant in the clause. For instance, in (3) below the Token *simple name* is a technical term that encapsulates the more general concept expressed in the gloss, while in (4) the Token *dataset password protection function* functions as an example/member of the more general set expressed in the Value.

- (3) A simple name (Tk) is a dataset name consisting of a set of up to eight alpha-numeric characters, beginning with an alphabetic character (Vl). (158)
- (4) One of the dataset protection methods (Vl) is the dataset password protection function (Tk). (55)

The fact that the Token and Value are (more or less) equivalent to one another is reflected in the potential of the clause to be in either active or passive voice. Although the verb *be* does not itself show voice, a specific lexical verb can reveal it.

- (5) A simple name (Tk) means/refers to a dataset name consisting of a set of up to eight alpha-numeric characters, beginning with an alphabetic character (Vl). (Active) (158c)<sup>4</sup>
- (6) One of the dataset protection methods (Vl) is exemplified by the dataset password protection function (Tk). (Passive) (55c)

At the same time that the identifying clause sets up a coding relation between the Token and Value, the clause constructs a kind of ‘equation’ between these two participants. In this equation, one participant can be interpreted as a constant – that which is being Identified, referred to as the ‘Identified’ (Id) – and the other a variable, the ‘Identifier’ (Ir). This variable, open-ended element can be probed with a wh-question: ‘What/which one is X?’ (Halliday 1967: 67; 1994: 123):

- (7) A simple name (Id/Tk) is a dataset name consisting of a set of up to eight alpha-numeric characters, beginning with an alphabetic character (Ir/Vl). (158)  
(Probe: “what is a simple name?” NOT “what is a dataset name consisting of a set of up to eight alpha numeric characters ...?”)
- (8) The valid access authority (Id/Vl) is the one assigned to the DASD dataset profile with the generic name most conforming to the dataset name (Ir/Tk). (211)  
(Probe: “what is the valid authority access authority? NOT “what is the one assigned to the DASD dataset profile ...?”)

---

<sup>4</sup> Some examples have been modified in order to illustrate a point; these semi-constructed examples include a ‘c’, e.g. (158c).

Identifying clauses can be differentiated along two dimensions: the directionality of the coding and the markedness of the function structure.

#### 4.1. Coding orientation

The directionality of the coding in identifying clauses leads to two coding orientations: decoding or encoding. In decoding clauses, the direction of the identification is from a concrete Token as the element being identified to an abstract Value as that which serves to identify it (Identifier). In encoding clauses, the move is in the opposite direction, that is, from an abstract Value as the element being identified to a concrete Token as Identifier. Further, the conflation of the different functional elements in the clause can be either unmarked or marked. In their unmarked, or typical, realization, the functions of Identified and Identifier conflate with the textual elements of Given and New respectively, as in (9) below.

- (9) A simple name (Tk/Id/Given) a dataset name consisting of a set of up to eight alpha-numeric characters, beginning with an alphabetic character (VI/Ir/New). (158)

The Identified therefore conflates with Given in its unmarked position as Theme of the clause (that is, in clause-initial position) while the Identifier conflates with New in its unmarked position at the end of the clause.<sup>5</sup>

The two sets of participants (Token and Value, Identified and Identifier) yield eight clause types (Halliday 1967: 67-69, 227; 1968: 190-92; Davidse 1992a, 1996). Since four of these assume an abstract Token and a concrete Value (not found in my data), the simpler four-cell model (Table 3) is included here.

**Table 3: The four-cell model of coding orientations**

	<b>Decoding</b>	<b>Encoding</b>
<b>Unmarked</b>	Tk/Id ^ V/Ir	VI/Id ^ Tk/Ir
<b>Marked</b>	VI/Ir ^ Tk/Id	Tk/Ir ^ VI/Id

Note: ^ indicates order of constituents

---

<sup>5</sup> New information can be interpreted as either information that has not been previously mentioned, information that is not recoverable from the context, and/or information that is newsworthy (Halliday 1994: 59, 298; Fries 1994: 230-33). Because the Identified and Identifier functions often conflate with Given and New, it is tempting to consider these functions as expressing similar meaning. However, Identified and Identifier are ideational and determine 'identity', while Given and New are textual and are concerned with information status (Halliday 1994: 124).

Examples of these four confluents, along with the probe clarifying the direction of the identification, are provided in Table 4 (New information is in bold).

**Table 4: Identifying clauses with unmarked and marked coding orientations**

	<b>Decoding</b>	<b>Encoding</b>
<b>Unmarked</b>	<p>Cautions on Attributes: A group-user attribute (Tk/Id) is a <b>user attribute that applies when a user is logged on with a specific group to use the data center</b> (VI/Ir). (92)</p> <p>(Probe: “What is a group user attribute?”)</p>	<p>This security method is called default value protection. Other security methods RACF offers (VI/Id) are <b>batch protection for unregistered datasets, generic name protection, and full resource name protection</b> (Tk/Ir). (27)</p> <p>(Probe: “What are other security methods RACF offers?”)</p>
<b>Marked</b>	<p>Resource: <b>That requiring protection by RACF</b> (VI/Ir) is called a resource (Tk/Id). (150)</p> <p>(Probe: “What is a resource?”)</p>	<p>Each data center user is identified by a user-ID. A user-ID or user-ID card is used to validate a legitimate user of the data center. Only legitimate users can use the data center. <b>The user-password</b> (Tk/Ir) is an important method of validating a user (VI/Id). (205)</p> <p>(Probe: “What is an important method of validating a user?”)</p>

Although identifying clauses in general are reversible (as suggested earlier), there are in fact differences in the voice potential between the decoding and encoding subtypes (Halliday 1994: 165): encoding clauses can be easily made passive whereas decoding clauses display a more ‘strained’ passive (Davidse 1992a). This difference is made clearer when a lexical verb is substituted for *be*. Thus, while the encoding passive structure in (10’) below is quite acceptable, the decoding passive structure in (11’) is clearly awkward and requires grammatical restructuring (*what is meant by*) to make it more acceptable.

- (10) Batch protection for unregistered datasets, generic name protection, and full resource name protection (Tk/Ir) exemplify other security methods RACF offers (VI/Id). (Encoding: active) (27c)
- (10’) Other security methods RACF offers (VI/Id) are exemplified by batch protection for unregistered datasets, generic name protection, and full resource name protection (Tk/Ir). (Encoding: passive) (27c)
- (11) A group-user attribute (Tk/Id) means a user attribute that applies when a user is logged on with a specific group to use the data center (VI/Ir). (Decoding: active) (92c)

- (11') ?A user attribute that applies when a user is logged on with a specific group to use the data center (VI/Ir) is [what is] meant by/defined by group-user attribute (Tk/Id). (Decoding: strained passive) (92c)

Differences in voice potential in encoding and decoding clauses can be attributed to differences in the ergativity structure of the two clause types: Decoding clauses are middle (non-agentive) while encoding clauses are effective (agentive).

#### 4.2. *Ergativity*

Unlike the transitivity system, which is a process + extension model, the ergative system is an 'instigation of process' model (Davidse 1992b: 109). In encoding identifying clauses, the Token is analyzed as an Agent because it functions as an 'embodiment' or (relatively) concrete expression of the Value (Davidse 1996: 390). In other words, the Token 'causes' the Value to have concrete expression. Thus, in (12) below, the Token/Agent offers several specific, concrete examples of the more general set of security methods offered by RACF.

- (12) Other security methods RACF offers (VI/Medium) are batch protection for unregistered datasets, generic name protection, and full resource name protection (Tk/Agent). (Encoding: effective) (27)<sup>6</sup>

Some encoding clauses are doubly effective in the sense that they allow a secondary Agent, the instigator of the process (called the 'Assigner' in relational identifying clauses) (Halliday 1994), to effect (or assign) the relation between the Token and Value. In (13), for example, the Token *RACF centre staff* is agentive in the sense that it enables the embodiment of the abstract meaning/Value, while the secondary Agent (Assigner) 'X' is responsible for assigning this Token to the Value.<sup>7</sup>

- (13) The resource maintenance personnel and auditor, chief manager included (VI/Medium) are called RACF center staff (Tk/Agent 1) [by X (Assigner/Agent 2)]. (Encoding: effective, double agency) (132c)

The effective ergative structure of encoding clauses reflects the 'tight-fit' between the Token and Value (i.e.  $X = Y$ ;  $Y = X$ ) and the relation can be interpreted as categorical and exhaustive.

---

<sup>6</sup> Just as the Actor is the participant most closely associated with the (material) Process in the transitive model (with the potential for a Goal to be added), it is the Medium that is most closely associated with the Process in the ergative model. The Medium is the "key figure in [the] process; ... the one through which the process is actualized, and without which there would be no process at all." (Halliday 1994: 163) The configuration of Medium + Agent can be expanded to include the instigator of the process.

<sup>7</sup> This example has been partially constructed because there are no instances in my data of an explicit Assigner (although an Assigner can be inferred from the passive construction).

In contrast, middle decoding clauses, as in (14) below, contain neither a primary nor a secondary Agent: The Token is not an embodiment of the Value, and there is no possibility of a secondary Agent instigating the relation. That is, 'Y is meant by X by A' is not possible:

- (14) A group-user attribute (Tk/Medium) means a user attribute that applies when a user is logged on with a specific group to use the data center (VI/Range). (Decoding: middle) (92c)<sup>8</sup>

The middle ergative structure of decoding clauses suggests a 'loose fit' – the relation is more open-ended than the relation constructed in encoding clauses. In other words, decoding clauses, in general, construct a less 'equivalent' relation than do encoding clauses (Harvey 1999).

## 5. Equivalence as experiential deautomatization

The multifunctional structure of the most equivalent of relational clauses – the prototypical definition – reflects, at a micro level, the macro processes operating in technical discourse (Harvey 1999). In technical discourse, the experiential sub-component of the ideational metafunction is mostly foregrounded or 'deautomatized' (to use Mukarovsky's 1977 term) in the structure 'X is Y'. At the same time, logical, interpersonal and textual meanings are typically backgrounded or 'automatized'. Following Mukarovsky, Halliday (1982: 130) describes the process of automatization as one in which "words and structures [function] simply as the most neutral, or unmarked, expression of the meanings that lie behind them" (see also Givón 1995; Jakobson 1932/1984). To put this another way, automatized grammatical structures "[realize] the semantic selections in an unmarked way – getting on with expressing the meanings, without parading themselves in patterns of their own" (Halliday 1982: 135). Experiential deautomatization represents a kind of 'ideational privileging' (Lemke 1990b) which in technical and technocratic discourses constructs an apparently homogeneous world-view, the validity of which is not open to questioning or scrutiny.<sup>9</sup> This impenetrable world-view is most clearly reflected in relations of equivalence, in which 'X is Y' and 'Y is X'.

---

<sup>8</sup> The Range is a 'quasi-participant' that "restates the process or specifies its extent or scope" (Halliday 1994: 149, 167; Davidse 1992b: 125)

<sup>9</sup> In formal definitions, a depersonalizing homogeneity is evidenced in the backgrounding of the source of the definition, 'X = Y' *cf.* 'We call X Y'. However, the encoding defining/naming clause in the passive voice ('Y is called X') is only partially personalized because the human Assigner remains implicit; at the same time, the clause is more effective than its decoding counterpart, precisely due to its potential for an Assigner. The inferred Assigner thus lends the weight of authority to the specified relation, but at the same time does not overtly 'personalize' the text (Harvey 1999).

### 5.1. *The grammatical construction of equivalence and non-equivalence*

Equivalent relations possess a number of distinguishing features. First, the clause must include (at least) two nominal groups, both of which must contain a logical head (noun). For instance, (15) below contains two nominal groups representing 'things' that can be equated. This prototypical (or 'complete') definition includes a *definiendum* (the Token), and a *definiens* (the Value), with the latter containing both a class word and defining characteristics.

- (15) Group (Tk/Id) (*definiendum*) is a user-group (class) that is formally registered in the RACF management dataset (defining characteristics) (VI/Ir) (*definiens*). (Decoding) (84)

On the other hand, the nominal group in the Value in (16) below does not contain a logical head (class) and is thus a non-prototypical (or 'partial') definition; the Value is constructed as a process rather than as a thing that can be equated to another thing.

- (16) Authority revoked (Tk/Id) (*definiendum*) means [0 class] that a user cannot use the data center (defining characteristics) (VI/Ir) (*definiens*). (Decoding) (20)

Even if the clause does contain two nominal groups that can be equated, and both contain a logical head, the relation need not be equivalent. To express equivalence, the relation between the Token and Value nominal groups must also be exhaustive (rather than open), one-to-one (as opposed to one-to-many), and presented as universally valid and non-negotiable. Whereas (complete) prototypical definitions in the default case express a one-to-one and exhaustive relation, other functions appear to be inherently non-exhaustive (e.g. describing relations) while yet others seem to be intrinsically one-to-many (e.g. exemplifying relations).<sup>10</sup>

Identifying relations incorporating descriptive aspects typically express non-equivalence since the quality/attribute is often non-intrinsic, transient or situationally specific (i.e. not universal). For instance, in (17) below (a holder-status relation) it is possible to imagine a situation in which a new user-password will violate the rule specified in the Token; in (18) (a form-function relation), the first qualifier need not be a group name, although it usually is.

---

<sup>10</sup> Compare this with the following expanded definition (i.e. extending over more than one sentence), the final clause of which contains two non-coreferential nominal groups. These Tokens are both associated with their own Values and thus construct two one-to-one relations rather than a one-to-many relation (i.e. the Values are the two operations referred to anaphorically): *The attribute of authority revoked (REVOKE) is designed to temporarily prevent a user from using the data center. The attribute of authority resumed (RESUME) restores the authority of a user whose authority has been revoked. A user in this position can no longer use the data center (once the current job or TSS session is completed). The date can also be specified to revoke or resume the user's authority from a particular date. These operations (VI/Id) are referred to as authority auto-revoked with a date specification and authority auto-resumed with a date specification (Tk/Ir). (Encoding) (17)*

- (17) Sometimes a current user-password must be changed because it has become known to someone or for some other reason. Also, users who have just been registered can only use the data center if they specify a new password during their first initial logon. A new user-password (VI/Id) must be a character string that has not been used (or “specified”) for some generations (Tk/Ir). (Encoding) (108)
- (18) The first qualifier of the VSAM catalog dataset name (VI/Id) is usually a group name (Tk/Ir). (Encoding) (89)

Notice that both of these clauses contain interpersonally deautomatizing features (the modal verb *must* and modal adjunct *usually* respectively), whose inclusion has the effect of making the validity or necessity of the relation negotiable (Flowerdew 1991).

Exemplifying clauses, on the other hand, are essentially non-equivalent since the example offered is assumed to be a representative of a larger set. One example may be offered in the clause, either implicitly, as in (19) below (in which we assume there are other access authorities) or explicitly, as in (20) (in which the example cited is presented as just one of the dataset protection methods on offer). Alternatively, several examples may be provided, as in (21). In each case, though, the examples do not exhaust the full set.<sup>11</sup>

- (19) The access authority is an authority that enables a user to access resources. For example, the authority used to reference or update a dataset (Tk/Ir) is regarded as an access authority (VI/Id). (Encoding) (5)
- (20) For each dataset, the dataset manager registers the users whose use of the dataset is authorized. Only those users whose access to the data center is authorized can access the approved datasets. One of the dataset protection methods (VI/Id) is the dataset password protection function (Tk/Ir). (Encoding) (55)
- (21) Data center users have a variety of objectives such as to calculate wages, and to calculate sales. Users are grouped according to their objective. Groups (VI/Id) may be divisions or departments, laboratories, or project teams (Tk/Ir). (Encoding) (94)

The unmarked equivalence status of the relational functions (e.g. defining, exemplifying, and so forth) can be disrupted. Functions that are typically presented as one-to-one can be construed as one-to-many, for instance. The disruption of our expectations regarding the typical equivalence status of a function is usually signaled by deautomatization of one or more of the non-experiential types of meaning: Logical, interpersonal and textual.

Logical deautomatization is achieved in two main ways. There may be a dependent

---

<sup>11</sup> Although it is unusual for all examples of a set to be included, in identifying clauses constructing a member-set relation, all members can be provided. For instance: *RACF provides other protection in addition to specific name protection. These functions (VI/Id) are the default, generic profile, unregistered dataset batch protection, and protection by the global check function (Tk/Ir). (Encoding) (192)* Thus, a relation can be one-to-many but non-exhaustive (as in example-set relations) or one-to-many and exhaustive (as in member-set relations).

clause or prepositional phrase expressing a condition or specific situation. This has the effect of reducing the universality of the relation by suggesting conditions under which the relation holds or through the specification of situations within which the relation is valid.

- a) If [Condition] is the case, X = Y (or Y = X)
- b) In the case of [Condition], X = Y (or Y = X)

- (22) **When the RACF data set is duplicated**, one of them (VI/Id) is called the main management data set (Tk/Ir), and the other (VI/Id) is called the sub-management data set (Tk/Ir). (Encoding) (103)

Alternatively, a one-to-many (rather than one-to-one) relation may be constructed, with the Token constituted by more than one element, or alternatively, the Value constituted by more than one element. That is, rather than one of the two nominal groups lacking a logical head (as suggested earlier) (and thus two things that can effectively be equated), one of the nominal groups contains two or more (typically non-coreferential) heads:

- c) Token (A, B ...) = Value
- d) Value (A, B ...) = Token

The following definition exhibits both types of logical deautomatization: Specific conditions under which the relation holds (expressed in the dependent *if* and *when* clauses) along with logical extension at nominal group level (i.e. two non-coreferential heads within the Token).<sup>12</sup>

- (23) The current group (VI/Id) is the group name specified **when** the user is used, and the default group **if** the group name is not specified (Tk/Ir). (Encoding) (37)

In summary, some relational functions can be considered to be default one-to-one (e.g. defining and naming) while others can be regarded as default one-to-many (e.g. exemplifying, describing, and so on). Functions such as definitions, which are typically expressed as equivalent (in the sense of being both one-to-one and exhaustive) can, in marked cases, be non-equivalent (i.e. either non-exhaustive or one-to-many). However, non-equivalence in such cases tends to be accompanied by the deautomatization of non-experiential features as a way of signaling the disruption.

## 5.2. Relational verbs as reflection of equivalence

My claim is that the degree of equivalence that is grammatically constructed between the two (main) nominal groups in identifying clauses is the main motivator of verb choice, at

---

<sup>12</sup> Compare this with clauses in which there are two coreferential heads, for example: **Users** who administer RACF operations and **those** who manage groups and resources (VI) are called managers. (Tk).

least in my data set. Two types of verbs are distinguished here: The general/neutral verb *be* and specific lexical verbs. *Be* (along with graphological relators such as the comma, used parenthetically, and the colon) signals neither the specific sub-relation intended (e.g. defining, naming, exemplifying) nor voice; specific verbs (e.g. *call*, *know as*, *refer to as*, *mean*, *explain*, *regard as*, *indicate*, refer to), in contrast, are unambiguous in these respects. The first main finding of the analysis (Table 5) is that decoding clauses usually contain the verb *be* (or a graphological relator) (84%) while encoding clauses typically contain a specific verb (72%).

**Table 5: Distribution of verbs (relators) and coding orientation**

	Decoding	Encoding
<b>General</b> ( <i>Be</i> /Graphic)	70 (84%)	23 (28%)
<b>Specific</b>	13 (16%)	60 (72%)
<b>Total</b>	83 (100%)	83 (100%)

These combinations may be the result of semantic compatibility between coding orientation and verb type (Harvey 1999). For instance, the general and non-exhaustive verb *be* (and the graphological relators) complements the semantic open-endedness and lack of exhaustiveness inherent in (middle) decoding clauses. In contrast, specific verbs appear to complement the closure and exhaustiveness of (effective) encoding clauses. Moreover, the concept of semantic compatibility can be related to the notion of equivalence to make the more general observation that equivalent relations prefer specific verbs, while less equivalent relations prefer *be* (or relator). This explains not only the major coding orientation/verb combinations mentioned above, but also accounts for the 16% of decoding clauses that take a specific verb and (all but one instance of) the 28% of decoding clauses containing *be*.<sup>13</sup>

The results suggest that decoding clauses (Table 6) construct non-equivalent relations in 100% of the total 83 decoding clauses, regardless of whether the verb is neutral or specific. This pattern holds for functions that, by default, express one-to-one relations (e.g. definitions) as well as those that, by default, express one-to-many relations (e.g. describing and exemplifying). In particular, the majority of decoding clauses in the data set are defining (but not defining/naming) and express one-to-one relations that are open (rather than exhaustive) (94% of the total 80 instances). The defining clause may be partial (usually missing a class word in the Value) (24) or complete but semantically open-ended as reflected in its middle ergativity (25).

- (24) The revoke attribute (REVOKE) (Tk/Id) indicates that a user cannot access the data center (Vl/Ir). (Decoding: middle/partial) (156)

---

<sup>13</sup> For discussion of grammatical probabilities from a systemic-functional perspective, see Halliday (1991b), Halliday & James (1993), Nesbitt & Plum (1988), and Plum & Cowling (1987).

- (25) Library (Tk/Id) is the DASD dataset holding the program specified by the program path control function (VI/Ir). (Decoding: middle/complete) (97)

**Table 6: Decoding clauses**

DECODING	Function	NEUTRAL				SPECIFIC				
		Exhaustive		Non-exhaust		Exhaustive		Non-exhaust		
		A	D	A	D	A	D	A	D	
Default: 1-1 (EQ)	Def/naming	-	-	-	-	-	-	-	2	2
	Defining (complete)	-	-	55	1	-	-	-	-	56
	Defining (partial)	-	-	10	-	-	-	10	-	20
	Naming	-	-	2	-	-	-	-	-	2
	<b>Subtotal</b>	0	0	67	1	0	0	10	2	80
Default: 1-Many (Non-EQ)	Exemplifying	-	-	-	1	-	-	1	-	2
	ID/Describe	-	-	-	1	-	-	-	-	1
	ID/Other	-	-	-	-	-	-	-	-	0
	<b>Subtotal</b>	0	0	0	2	0	0	1	0	3
	<b>Total</b>	0	0	67	3	0	0	11	2	83

Note: A = Automatized; D = Deautomatized

The results for encoding clauses (Table 7) suggest more variability depending on the function of the relation. Unlike decoding clauses, encoding clauses that construct a one-to-one relation tend to be exhaustive rather than open (92% of the total 83 encoding clauses). The most common pattern for one-to-one encoding clauses is that they contain a specific (passive) defining/naming or naming verb (93% of the total 60 one-to-one clauses). The most common pattern for one-to-many encoding clauses is that they consist of functions other than defining (which is unsurprising) and contain *be* (91% of the total 23 one-to-many clauses).

The majority of clauses in the data therefore conform to the pattern of equivalent relations combined with specific verbs, and non-equivalent relations combined with the neutral verb/relators. However, there are six clauses – two decoding and four encoding (indicated in Tables 6 and 7 in bold-type) – that require discussion. Although five of these clauses can be explained with reference to the general principle, they illustrate how default patterns can be disrupted, and how such disruption is signaled. (The sixth clause, asterisked in Table 7, does not conform to the pattern and is discussed further below.)

First, relational clauses with a combined defining/naming function are mostly expressed in encoding clauses and grammatically construct the highest degree of equivalence. However, there are two decoding clauses in Table 6 that are defining/naming

in function, contain a specific verb, but are non-equivalent. According to the general principle, this seems to be a semantically incompatible combination. The first of these two decoding clauses appears below in clause (d) of (26):

**Table 7: Encoding clauses**

Encoding	Function	NEUTRAL RELATOR				SPECIFIC VERB				
		Exhaustive		Non-exhaust		Exhaustive		Non-exhaust		
		A	D	A	D	A	D	A	D	
Default:	Def/naming	-	-	-	-	52	-	-	1	53
1-1 (EQ)	Defining (complete)	1*	-	-	1	-	-	-	-	2
	(partial)	-	-	-	-	-	-	-	1	1
	Naming	-	-	-	-	-	4	-	-	4
	<b>Subtotal</b>	1	0	0	1	52	4	0	2	60
Default:	Exemplifying	-	-	4	3	-	-	-	1	8
1-Many (Non-EQ)	ID/Describe	-	-	3	6	-	-	-	1	10
	ID/Other	-	-	1	4	-	-	-	-	5
	<b>Subtotal</b>	0	0	8	13	0	0	0	2	23
	<b>Total</b>	1	0	8	14	52	3	1	4	83

Note: A = Automatized; D = Deautomatized

- (26) Standard DASD dataset naming rules
- a) An alphanumeric character string consisting of a maximum of eight characters starting with an alphabetic character is known as a simple name.
  - b) A character string linking two or more simple names with a period “.” is called a qualified name.
  - c) If a dataset name consists of a simple name, the simple name itself (VI/Id) is called a first qualifier (Tk/Ir). (Encoding)
  - c) **If** the dataset consists of a qualified name, the first simple name in the dataset name (VI/Id) **is called** a first qualifier (Tk/Id). (Decoding) (66-67)
- (Probe for clause (c): “what is the simple name itself [called]?” Probe for clause (d): “in this case what is called the first qualifier”)

The defining clauses in (c) and (d) together set up a non-equivalent relation. Clause (c) is a typical encoding defining/naming clause with an unmarked Id ^ Ir structure. The element being sought in this definition is the name that is given to the simple name in a dataset name that consists of a simple name only – *the first qualifier*. In clause (d),

however, the variable being sought is not the name ‘the first qualifier’ (since we have already been introduced to this name), but rather, which element is the first qualifier in a qualified name (in this case, it is the first simple name in the qualified name). The marked Ir ^ Id information structure in clause (d) (a case of textual deautomatization) suggests a disruption. Two different Values have, in effect, been assigned to the same Token (*the first qualifier*) (i.e. Value [A, B] = Token). The resulting lack of exhaustiveness between the Token and Value is signaled by the logical deautomatization in both clauses (c) and (d) (in the conditional clauses *if ...*). Thus, although clause (d) is unusual in the combination of decoding and specific naming/defining verb, we are alerted to its atypicality by the deautomatization of both textual and logical features.

The decoding glossary definition in (27) below is also unusual. Although it combines a one-to-one relation and a specific naming/defining verb, the textual deautomatization, reflected in the marked information structure (Ir ^ Id), alerts us to the atypical lack of equivalence. A specific class word is missing (its place is held by the demonstrative *that*; see Litowitz 1977) and the definition is incomplete.<sup>14</sup>

- (27) Resource: That [0 class] requiring protection by RACF (Value/Ir) **is called** a resource (Tk/Id). (Decoding) (150)  
 (Probe: “what is that requiring protection by RACF called?” NOT “what is a resource?”)

The encoding (defining) clause in (28) contains the verb *be* rather than a specific verb and constructs a relation of non-equivalence. The term *unidentified user* has two (related) meanings and this logical extension (deautomatization) is explicitly signaled by the adverb *also*, and a reason for the extension is provided (*because ...*).

- (28) A user who uses the data center without specifying the user-ID (VI/Ir) is called an unidentified user (Tk/Id). A user who is not registered in the RACF management dataset (VI/Id) is **also** an unidentified user (Tk/Ir) || **because** the user does not have a user-ID. (Encoding) (175)

The following encoding (defining/naming) clause is also incomplete: The Value nominal group does not contain a head and represents a logical under-extension signaling non-equivalence.

- (29) The protection method for shared DASD datasets depends on how these datasets are named. How the datasets are named (VI/Id) [0 Head] is called the naming rule (Tk/Ir). (Encoding) (107)

The clause in (30) below represents the only exception to the general pattern described in this paper. This clause is an encoding defining clause. It contains the verb *be*

---

<sup>14</sup> The marked information structure (Ir ^ Id) may have been influenced by the fact that demonstratives do not usually appear in a position of tonic prominence in the clause (i.e. in clause-final position) (Halliday and Hasan 1976).

but nonetheless expresses a relation of equivalence between the two nominal groups (which the previous examples do not). The definition is complete and the relation between the Token and Value is exhaustive, categorical and one-to-one. Nor is there any deautomatization of interpersonal, textual, or logical features.

(30) User-Password Change Inhibit Interval

Certain restrictions apply to the user-passwords recorded. If a user-password is changed many times, the same user-password may be reset. To prevent a user-password from being constantly changed, an interval is set during which the user-password cannot be changed. This interval (VI/Id) is the user-password change inhibit interval (Tk/Ir) (Encoding) (206)

This clause is fascinating since, during the initial data collection (from the manuals and glossary), I copied it down, incorrectly as I later discovered, as:

(30') This interval is called the user-password change inhibit interval.

Although I did not realize it at that time, my instincts about verb choice in identifying clauses may well have been responsible for the copying error.

## 7. Conclusion

The most common-sense explanation for choosing specific verbs in relational clauses over the verb *be* is to avoid the ambiguity associated with *be*. Given its potential for ambiguity, it is somewhat surprising that *be* is often perceived to be the default choice, that is, the option chosen unless there is some reason for choosing otherwise. In this paper I have attempted to shed some light on this issue by analysing verb choice in a relatively small corpus of identifying clauses from English technical manuals. I have argued that a semantic complementarity exists between the general and under-specified verb *be* and clauses in the decoding orientation, whose middle ergativity renders them semantically non-exhaustive. On the other hand, specific lexical verbs (e.g. *call*, *mean*) are apparently more semantically compatible with clauses in the encoding orientation, especially those whose double agency allows for a human to assign the Value to the Token, and *vice versa*.

More generally, I have proposed that verb choice patterns in my data can be explained by invoking the notion of equivalence. Equivalent relations are both one-to-one and exhaustive, and relational functions (e.g. defining, naming, exemplifying, and so on) tend to be, in the default case, either one-to-one or one-to-many (but exhibit more variability with respect to exhaustiveness). Simply stated, equivalent relations prefer specific verbs, while non-equivalent relations prefer *be* (or graphological relators), regardless of coding orientation (but keeping in mind the typical combinations). What is perceived to be the typical equivalence status of a particular function can be disrupted and the incongruence between our expectations of the relationship between equivalence, coding orientation and verb type is usually signaled by the deautomatization (foregrounding) of non-experiential features in the clause (e.g. logical, interpersonal or textual). Deautomatization of these meanings in general makes the relation more open and negotiable, and less impenetrable.

To test these principles further, the analysis of a larger corpus of identifying clauses in technical discourse is necessary. A cross-comparison with other formal registers, e.g. scientific, academic, bureaucratic, would also help explain registerial variation in the area of knowledge construction. The kinds of patterns that occur in formal registers in languages other than English could also be studied. One issue to keep in mind in interpreting the results from this study is that the source manuals were drafts (not finished products), and were, in their original form, translated from Japanese. This may account for some of the more unusual patterns and could well have implications for teachers and learners of English, as well as translators.

With respect to the probabilities with which various grammatical features appear in different registers, Halliday (1991a) has noted that speakers of a language are able to make informed guesses about the relative frequencies of particular words. This informed guesswork represents one “aspect of knowing the language” (1991a: 35). Native speakers or writers of English may well have an instinctive understanding of when to use *be* and when to use more specific verbs based on probabilities as well as a deeply rooted (but difficult to explain) knowledge of what is, and what is not.

## References

- Davidse, K. (1991) *Categories of experiential grammar*. Ph.D. thesis, University of Leuven.
- Davidse, K. (1992a) A semiotic approach to relational clauses. *Occasional Papers in Systemic Linguistics* 6: 99-131.
- Davidse, K. (1992b) Transitivity/ergativity: The janus-headed grammar of actions and events. In M. Davies & L. Ravelli (eds.), *Advances in systemic linguistics: Recent theory and practice*. London: Pinter, pp. 103-135.
- Davidse, K. (1996) Turning grammar on itself: Identifying clauses in linguistic discourse. In M. Berry, C. Butler, R. Fawcett & G. Huang (eds.), *Meaning and form: Systemic functional interpretations (Meaning and choice in language: Studies for Michael Halliday)*. Ablex: Norwood, New Jersey, pp. 367-393.
- Flowerdew, J. (1991) Pragmatic modifications on the ‘representative’ speech act of defining. *Journal of Pragmatics* 15: 253-264.
- Fries, P. (1994) On theme, rheme and discourse goals. In M. Coulthard (ed.), *Advances in written text analysis*. London: Routledge, pp. 229-249.
- Givón, T. (1995) *Functionalism and grammar*. Amsterdam: John Benjamins.
- Halliday, M.A.K. (1967) Notes on transitivity and theme in English (Parts 1 and 2). *Journal of Linguistics* 3.1: 37-81 & 3.2: 199-244.
- Halliday, M.A.K. (1968) Notes on transitivity and theme in English (Part 3). *Journal of Linguistics* 4.2: 179-215.
- Halliday, M.A.K. (1978) *Language as social semiotic*. London: Edward Arnold.

- Halliday, M.A.K. (1982) The de-automatization of grammar: From Priestly's 'An Inspector Calls'. In J. Anderson (ed.), *Language form and linguistic variation: Papers dedicated to Angus McIntosh*. Amsterdam: Benjamins, pp. 129-59.
- Halliday, M.A.K. (1988) On the ineffability of grammatical categories. In J. Benson, M. Cummings & W. Greaves (eds.), *Linguistics in a systemic perspective*. Amsterdam: Benjamins, pp. 27-51.
- Halliday, M.A.K. (1991a) Corpus studies and probabilistic grammar. In K. Aijer & B. Altenberg (eds.), *English corpus linguistics: Studies in honour of Jan Svartvik*. London & New York: Longman, pp. 30-43.
- Halliday, M.A.K. (1991b) Towards probabilistic interpretations. In E. Ventola (ed.) *Functional and systemic linguistics: Approaches and uses*. Berlin, New York: Mouton de Gruyter, pp. 39-61.
- Halliday, M.A.K. (1994) *Introduction to functional grammar*. (2<sup>nd</sup> Ed.) London: Edward Arnold.
- Halliday, M.A.K. & R. Hasan (1976) *Cohesion in English*. London & New York: Longman.
- Halliday, M.A.K. & J. R. Martin (1993) *Writing science: Literacy and discursive power*. London: Falmer.
- Halliday, M.A.K. & Z. James (1993) A quantitative study of polarity and primary tense in the English finite clause. In J. Sinclair, M. Hoey & G. Fox (eds.), *Techniques of description: Spoken and written discourse*. London: Routledge, pp. 32-66.
- Harvey, A. (1996) *Equivalence and depersonalisation in definitions: An exploration of lexicogrammatical and rhetorical patterns in English technical discourse*. Ph.D. thesis, University of Sydney.
- Harvey, A. (1999) Definitions in English technical discourse: A study in metafunctional dominance and interaction. *Functions of Language* 6.1: 53-94.
- Jakobson, R. (1932/1984) The structure of the Russian verb. In L. Waugh & M. Halle (eds.), *Russian and Slavic grammar studies, 1931-1981*. Berlin, New York & Amsterdam: Mouton, pp. 1-14.
- Langacker, R. (1987) *Foundations of cognitive grammar*. Stanford, California: Stanford University Press.
- Lemke, J. (1990a) *Talking science: Language, learning and values*. Norwood, N.J.: Ablex.
- Lemke, J. (1990b) Technical discourse and technocratic ideology. In M.A.K. Halliday, J. Gibbons & H. Nicholas (eds.), *Learning, keeping, and using language*. Amsterdam & Philadelphia: John Benjamins, pp. 435-460.
- Litowitz, B. (1977) Learning to make definitions. *Journal of Child Language* 4: 289-304.
- Martin, J.R. (1992) *English text: System and structure*. Amsterdam: Benjamins.
- Matthiessen, C. (1995a) *Lexicogrammatical cartography: English systems*. Tokyo: International Language Sciences Publishers.
- Matthiessen, C. (1995b) Theme as an enabling resource in ideational 'knowledge' construction. In M. Ghadessy (ed.), *Thematic development in English texts*. London: Pinter, pp. 20-55.
- Mukarovsky, J. (1977) *Word and verbal art: Selected essays*. New Haven: Yale University Press.
- Nesbitt, C. & G. Plum (1988) Probabilities in a systemic grammar: The clause complex in English. In R. Fawcett & D. Young (eds.), *New developments in systemic linguistics, Vol 2: Theory and application*. London: Frances Pinter, pp. 6-38.

Newman, J. (in prep) (ed.) *The linguistics of sitting, standing and lying*.

Plum, G. and A. Cowling (1987) Social constraints on grammatical variables: Tense choice in English. In R. Steele & T. Threadgold (eds.), *Language topics: Essays in honour of Michael Halliday*. Amsterdam: John Benjamins, pp. 281-305.

Plum, G., C. Matthiessen, M. O'Donnell, L. Zeng, C. Nesbitt, A. Harvey, M.A.K. Halliday & J. Bateman (1992) *EDA research and development: Final report*. Internal Document, Fujitsu Australia Ltd.

Quirk, R., S. Greenbaum, G. Leech & J. Svartvik (1985) *A comprehensive grammar of the English language*. London: Longman.

Temmerman, M. (1994) *Definities in schooltaal: Communicatieve, cognitieve en tekstuele aspecten van definiëren in de derde graad van het basisonderwijs*. (2 volumes). Ph.D. Thesis, Antwerpen: Universitaire Instelling Antwerpen.

Van Oosten, J. (1984) Sitting, standing and lying in Dutch: A cognitive approach to the distribution of the verbs *zitten*, *staan*, and *liggen*. In J. van Oosten & J. Snapper (eds.), *Dutch linguistics in Berkeley*. Berkeley: UCB, pp. 137-160.

Wignell, P., J.R. Martin & S. Eggins (1993) The discourse of geography: Ordering and explaining the experiential world. In M.A.K. Halliday & J.R. Martin (eds.), *Writing Science: Literacy and discursive power*. London & Washington, DC: Falmer.