Chapter 6

Mapping Semantics and Lexico-grammar

1. What is Inter-Stratal Mapping?

An important part of any multi-stratal theory is the component which maps between the strata - the inter-stratal mapping resources. These resources represent the regular correspondences between each pair of strata. I will focus here on the mapping between the semantic and the lexico-grammatical strata, shown diagrammatically in Figure 6.1.

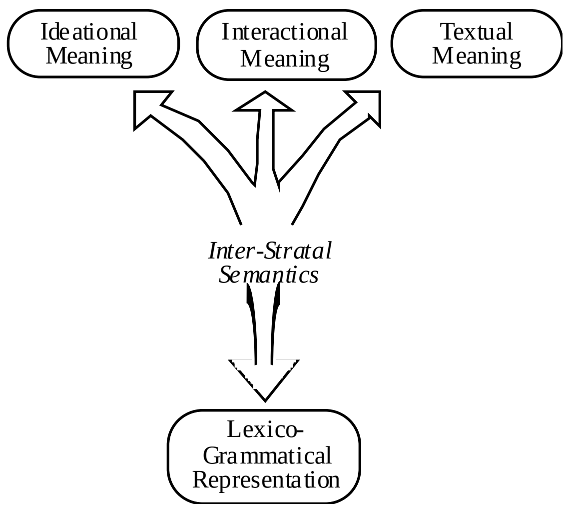

Figure 6.1: An Encoding (Generation) View of Inter-stratal Mapping

From a generation point of view, the mapping is interpreted as the encoding of ideational, interactional and textual meaning in a lexico-grammatical structure:

> "Grammar is the level at which the various strands of meaning potential are woven into a fabric; or, to express this non-metaphorically, the level at which the different meaning selections are integrated so as to form structures." (Halliday 1973, p93).

From an analysis point of view, the mapping is interpreted as decoding lexico-grammatical structure to recover the speaker's meaning. A hearer recovers the speaker's meanings by interpreting the choices of lexical items, and the ways in which these are put together (the grammatical structure).

The use of strata in Systemic Linguistics, and the mapping between these strata, draws heavily on the work of the Stratification school of linguistics (cf. Lamb 1966). My own approach to inter-stratal mapping is influenced in particular by Lockwood (1972) and Lamb (1971).

1.1 Inter-stratal Mapping Without Semantic Structure

Most of the Systemic discussion of inter-stratal mapping has assumed a paradigmatic semantics without any syntagmatic element: the semantics consists of a system network, but there is no semantic structure. A single pass through the network will produce a complete semantic specification. A semantic specification thus consists simply of a set of features: a selection-expression from the semantic network. Semantico-grammatical mapping is then simply a matter of relating each of these semantic features to their grammatical correlate. The mechanism is usually one of preselection - each semantic feature preselects one or more features of the clause which expresses the semantics:

> "In general, the options in a semantic network will be realised by selections of features in the grammar..." (Halliday 1973, p85).

Most non-computational Systemic work has followed this approach (cf. Hasan 1987). Some computational work has also (e.g., Patten 1988). Other computational approaches (discussed later) introduce semantic structure.

Most descriptions of the inter-stratal mappings have not been very detailed, showing only the typical realisations of semantic features (for instance, Halliday 1984; Patten 1988). Hasan (1987) is an exception, offering detailed socio-semantic networks, which motivate a wide range of lexico-grammatical forms. Lexico-grammatical realisations are provided for each feature.

1.1.1 Mapping Complex Sets of Features

Some mapping approaches associate lexico-grammatical constraints with a single semantic feature. However, Halliday points out that inter-stratal relations are not always that simple, in his discussion of the grammar-phonology interface:

> "The pattern is rather complex, because there is no one-to-one correspondence between options in the grammar and options in the phonology: a large number of different grammatical systems are realized by means of selection in the phonological system of tone." (Halliday 1973, p94).

The semantic-grammar interface is no less complex: when we consider the wide range of means available to express a given meaning, we see that the mapping resources need to be logically complex. A single semantic feature will be perhaps realised by this *or* that grammatical form, or a semantic feature will have particular realisation only in conjunction with other semantic features.

1.1.2 Preselection through Role-Chains

For the most part, semantic features are realised by preselecting lexico-grammatical features, particularly at clause-level. For instance, Patten's inter-stratal mapping resources consist of rules like the following (see Patten 1988, p198):

Feature	Realisation
deprivation	Threat: marked-negative Threat: nonplace Threat: residual

The semantic feature *deprivation* is realised by preselecting the filler of the role *Threat* to be a clause with features: *marked-negative*, *nonplace* and *residual*. The *Threat* role is an inter-stratal relation, mapping from the semantic content to its grammatical realisate.

Patten's implementation also allows preselection of the constituents of the clause. For instance, the *deprivation* feature has additional realisations: [1]

Feature	Realisation
deprivation	Threat.Process: give Threat.Medium: noun Threat.Medium.Head: sweet

These rules makes use of *role-chaining* -- allowing realisation rules to refer not only to the immediate roles of a unit, but also to the roles of roles. *Threat.Medium*, for instance, refers to the *Medium* role of the *Threat* role of the present unit. Role-chains can be of any length: *Threat.Medium.Head*, for instance, refers to the *Head* of the *Medium* of the *Threat* role. This is a powerful extension of the Systemic formalism, which has also been used by Cross (1991), Kasper & O'Donnell (1990), and in the WAG system.

1.2 Introducing Semantic Structure

Halliday points out the necessity of semantic structure:

> "...it has been possible to by-pass the level of semantic structure and go straight into lexico-grammatical constituent structure. That's all right for certain limited purposes. But there is an obviously a limitation here, and when we attempt semantic representation for anything other than these highly restricted fields, it is almost certainly going to be necessary to build in some concept of semantic structure. But what it will look like exactly I don't know. I don't think we can tell yet." (Halliday 1978, p41).

In the time since this quote was published, computational implementations of Systemic grammar have begun to utilise semantic structure, for instance, ideational structuring of the sentence (e.g., Penman), and sometimes, interactional and textual structure (e.g., Cross 1991; WAG).

The introduction of semantic structure complicates the inter-stratal mapping problem: we no longer have just a set of features to be realised, but rather a configuration of semantic units which need to be related to the configuration of lexico-grammatical units. The mapping is complicated in two ways:

- **Realising more than one selection expression**: We now need to relate the features of <u>each unit</u> of the semantic structure to the lexico-grammatical form, rather than realising a single selection expression. The realisation of these different units need to be integrated together in a single lexico-grammatical structure.
- **Need for structural mappings**: We need to relate the elements of the semantic structure to those of lexico-grammatical structure. For instance, we need to show that the Actor[2] of a mental process (semantic structure) is realised as the Subject of the clause in an active clause, or, in a passive clause, as the Complement. Structural information thus needs to be included in our mapping rules to map semantic and grammatical units.

---

[1]Patten actually uses a '<' symbol to mark role-chains. I use however WAG's role-chains marker: '.', to be consistent with later discussion.

[2]I am talking about Actor as a semantic role -- part of the ideational structure. This role name should not be confused with term Actor used as grammatical roles, for instance, in Halliday (1985) and in the Nigel grammar.

1.2.1 System-Based Inter-Stratal Mapping

During the Eighties, the Penman project developed an inter-stratal architecture which addressed these problems (cf. Mann 1983b, 1985a; Matthiessen 1984, 1987a, 1987b, 1990). It was based on the idea of having a ‘chooser’ sitting on top of each lexico-grammatical system, a *decision-tree* which can ask questions of the semantics in order to make the appropriate lexico-grammatical decision. This approach resulted in quite a powerful inter-stratal resource, which handles ideational and (limited) interactional mappings quite successfully. This approach will be explored in section 2 of this chapter.

Cross (1991) also provides a system-based constraint system, as one of several inter-stratal mechanisms she uses. The basic mechanism consists of a data-structure for each system which defines a set of condition/choice pairs. If the first condition is met, then the associated feature is chosen, else the next condition is tested.

1.2.2 Feature-Based Inter-Stratal Mapping

One problem of the chooser-inquiry architecture is that it is procedurally implemented - the *inquiries* which query the semantics are lisp function calls. The major problem of procedural implementations of linguistic resources is that they are not reversible - the chooser-inquiry interface could not be used for analysis as well as generation.

Kasper (1989), as part of his ongoing work in Systemic analysis, desired to use these inter-stratal resources for analysis, rather than generation. This task required the declarativisation of the chooser-inquiry interface. The declarativisation was carried out by Kasper and myself in 1990 (Kasper & O’Donnell 1990). Rather than perform this task by hand, we wrote a program to translate as much as possible of the existing chooser-inquiry interface into a more declarative form. The inquiry calls, although implemented in lisp, were regular enough in form so that as many as 80% of them could be translated automatically. The resulting constraints on grammatical features were represented in *Loom*, a knowledge representation language (MacGregor & Bates 1987).

Apart from the declarativisation of the inquiries, Kasper required the semantic constraints on the grammar to be expressed as *constraints on lexico-grammatical features*, rather than *decision trees attached to systems*. This represents a shift of inter-stratal focus from the system to the features themselves.

Only a prototype of this work was ever implemented. However, the general design put forward by Kasper has formed the basis of the Semantico-Grammar constraint system used in the WAG system, which is both declarative and feature-based. This system will be discussed in section 3 of this chapter.

1.3 Semantic-Based and Grammar-Based Mapping

The mappings between two strata tend to be associated with one strata or the other. Patten (1988), for instance, represent these mappings as realisations of semantic choices, in other words, as lexico-grammatical constraints on semantic features. I call this a *semantic-based* approach to semantico-grammatical mapping.

Most Systemic systems take a *grammar-based* approach: the mappings are stated as semantic conditions on grammatical features (as in the Kasper-O’Donnell approach), or attached to lexico-grammatical systems (as in the Penman approach).

Bateman *et al.* (1992) suggests a third possibility -- not attaching the mappings to either the semantics or the grammar, but rather providing a separate mapping resource, which associates semantic *configurations* with lexico-grammatical *configurations*. A small set of choosers and inquiries were re-represented in such a form using TFS, a unification-based formalism (Emele & Zajac 1990). This translation of choosers was partially based upon the preliminary work by Kasper & O’Donnell (1990).

While each of these approaches has its merits, I will explore only grammar-based approaches below.

2. The Chooser-Inquiry Interface

The chooser-inquiry interface, Penman's solution to the inter-stratal mapping problem, associates a decision tree with each grammatical system, where the decisions are made by referring to the semantics, and each leaf of the decision tree nominates the selection of a particular grammatical feature.

2.1 An Example

To demonstrate how the chooser-inquiry interface works, I will provide an example drawn from the Nigel grammar. The example involves the choice of feature from within the *Recipiency* system. This system determines whether a Beneficiary role is included in the grammatical structure. The system is shown in figure 6.2.

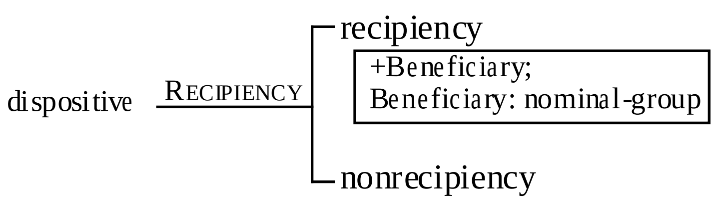

Figure 6.2: Penman's Recipiency System

The semantic constraints on these features can be stated as follows:

a) *recipiency* is appropriate when:

- the semantic referent has either a *Recipient* or *Beneficiary* role, and this role is selected for expression.

b) *nonrecipiency* is appropriate when either:

- the semantic referent has neither a *Recipient* nor a *Beneficiary* role; or
- the semantic referent has either of these roles, but the roles are not selected for expression.

The ideational systems which condition the Recipiency system are shown in figure 6.3. Processes can have a Recipient role (the person to whom something is given), and possibly a beneficiary role (the person for whose benefit the process is performed).

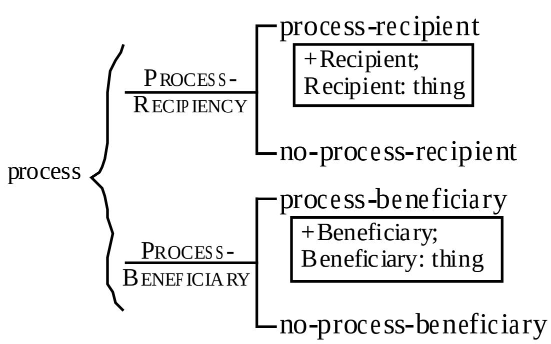

Figure 6.3: Ideational Systems Conditioning the Agency System

A problem with this description is that a process might have both a Recipient and a Beneficiary, as in "I gave the book to John for Mary". The Penman grammar allows only one of these roles to be expressed. However, it would be a simple matter to correct this problem.

2.2 Choosers & Inquiries

In Penman, the selection of a feature in a system is determined by a chooser - a decision tree. The decision tree is traversed by making inquiries of the semantics at each branch-point, and taking one branch or the other depending on the result returned by the inquiry. Each leaf of the tree nominates a particular grammatical feature for selection. The chooser for the *Recipiency* system is shown in figure 6.4.

```
(chooser
 :name recipiency-chooser
 :definition
   ((ask (possession-onset-q Process)
      (nopossessioncreated (choose nonrecipiency))
      (possessioncreated
        (ask (possession-onset-specification-q Process)
          (specified (identify Beneficiary
                          (beneficiary-id Process)
                    (choose recipiency))
          (notspecified (choose nonrecipiency))))))
```

Figure 6.4: The Recipiency Chooser

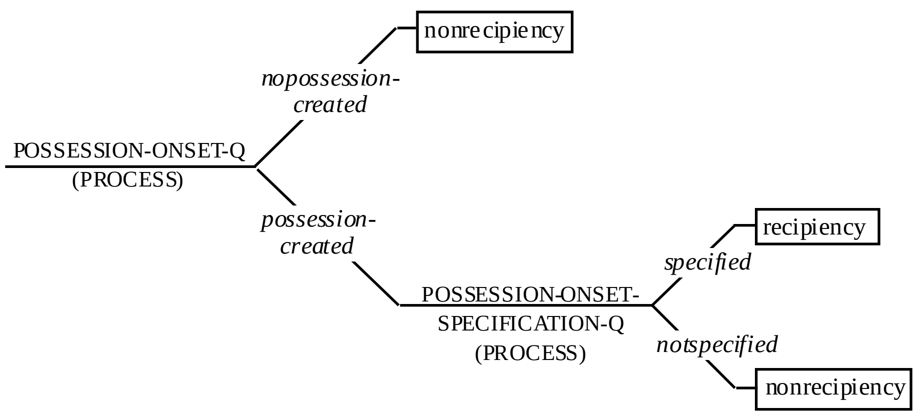

Figure 6.5: The Recipiency Chooser in Graphic Form

Figure 6.5 shows the chooser in graphic form. Graphical representation better reveals the decision-tree nature of the chooser. The boxes at the leaves of the tree represent the feature choice resulting from that path through the tree.

2.2.1 Branching in the Chooser

The movement through the chooser depends on the result returned by a call to a *decision inquiry* (ending in '-q'). Each decision inquiry represents a question about the semantics being expressed[3]. The value that the inquiry returns depends on the configuration of the current semantic representation. For instance, *(possession-onset-q Process)* is the first decision query called. It returns *possessioncreated* if the process being expressed contains either a Beneficiary or a Recipient role; or *notpossessioncreated* otherwise.

[3]Theoretically, decision inquiries also query the macro-semantic structure, and the global context, although none of the inquiries in the Nigel resources do so.

Decision inquiries are procedurally implemented, in Lisp. The definition for the *possession-onset-q* inquiry is:

```
(defun Possession-Onset-Q-Code (Process)
  "Is there a recipient?"
  (if (or (fetch-feature 'Beneficiary Process)
        (fetch-feature 'Recipient   Process))
    'possessioncreated
    'notpossessioncreated))
```

The second inquiry in the chooser -- *possession-onset-specification-q* -- tests whether the Beneficiary/Recipient of the process is actually to be expressed, even though the role exists (a form of content selection). At present, there is no code for this inquiry -- it is defaulted: it always returns *specified* -- if the role exists, then it is expressed.

2.2.2 Structural Operations

During the generation of each grammatical unit, the choosers establish the ‘referent’ of each of the unit’s constituents: the ideational unit which the grammatical unit is realising. There are various chooser operators to create these mappings. The two most common types are:

identify, e.g., *(identify Beneficiary (beneficiary-id Process)*: The identify operator associates a grammatical role with a semantic unit, its referent. The identify operation uses *identification inquiries* to recover a particular role in the semantics, e.g., *beneficiary-id* in the example above. According to Bateman *et al.*(1992), “The function of [an identification inquiry] is precisely to locate particular semantic entities with respect to semantic entities which are already known.” (p918). As with decision inquiries, they are implemented procedurally. The *beneficiary-id* inquiry is shown below, it returns a pointer to the semantic Beneficiary if it exists, or a Recipient otherwise:

```
(defun Beneficiary-ID-Code (process)
  "Return the beneficiary or recipient."
  (or (fetch-feature 'Beneficiary Process)
     (fetch-feature 'Recipient   Process)))
```

copyhub, e.g., *(copyhub Sayer Agent):* the copyhub function assigns the referent role of one grammatical function to that of another. It is not used in the chooser above.

2.3 Problems with the Chooser-Inquiry Architecture

There are various problems with the chooser-inquiry formalism:

- **Procedural Implementation of Inquiries**: The inquiries are at present implemented as lisp code, which limits their transportability.
- **Non-Bidirectionality of Choosers**: Choosers are expressed in a manner which is useful for forward-traversal through a system. Parsing, however, relies on backward-traversal (finding a path), so choosers are not expressed in a manner which a parser can use. The procedural implementation of inquiries also hinders their bi-directional use.
- **Lack of Simultaneity in Choosers**: Matthiessen (1987b) points out that choosers are limited to a tree structure. It is sometimes desirable to break out of this tree limitation, allowing the equivalent power of a system network, e.g., disjunctive and conjunctive entry conditions, and simultaneous systems. These can, however, be simulated:

> "There are no complex inquiry presentation conditions in the present version of the inquiry framework. In other words, there can be no conjunction or disjunction of branches before an inquiry is presented. Conjunction can be simulated by accumulating responses as the depth increases in the tree of inquiries. Disjunction can be simulated by using the same inquiry in more than one place." (Matthiessen 1987b, pp14-15).

3. WAG: Feature Selection Constraints

In the WAG system, the mapping between two strata is represented by associating higher-level constraints with features of the lower stratum. Each feature of a network is assigned a *selection constraint*. If the feature is to be chosen, then its selection constraint must be satisfied.

Selection constraint are similar to realisation statements -- they are both expressed using WAG's constraint language (see chapter 8 for more detail). Selection constraints, however, map upwards to a higher stratum, while realisation constraints map downwards, to a lower rank. Selection constraints also allow more logical complexity than realisations, involving conjunctions, disjunctions and negations of constraints. Some additional operators are also used.

3.1 Re-Representing The Process-Type Chooser

To allow comparison with Penman's mapping formalism, I will re-represent the Recipiency chooser from above in terms of feature selection constraints, as shown in Figure 6.6. The translation results in a complex semantic constraint on each feature in the system.

As noted above, Penman's resources do not supply a definition for *possession-onset-specification-q*. I assume this inquiry is a test of relevance, and have thus encoded it as:

```
(:or (:relevant Process.Beneficiary)
    (:relevant Process.Recipient) )
```

```
(defconstraint recipiency
  (or (:and (:exists Referent.Beneficiary)
            (:relevant Referent.Beneficiary)
            (:same Referent.Beneficiary Beneficiary.Referent))
      (:and (:exists Referent.Recipient)
            (:relevant Referent.Recipient)
            (:same Beneficiary.Referent Referent.Recipient)))

(defconstraint nonrecipiency
  (:and (:or (:not (:exists Referent.Beneficiary))
             (:not (:relevant Referent.Beneficiary)))
        (:or (:not (:exists Referent.Recipient))
             (:not (:relevant Referent.Recipient)))))
```

Figure 6.6: The Recipiency Chooser in WAG's Constraint Formalism

These feature selection constraints replace the chooser *and* the inquiries. The resultant form is more succinct than the equivalent chooser and inquiries, and represents the mapping from grammar to semantics in a single data-structure, rather than the two required by Penman.

3.2 The Constraint Language

To show how the selection-constraints function, I will tackle several issues in turn. The formalism will be discussed more fully in chapter 8 on feature logic. Only those components relevant to inter-stratal mapping are discussed here.

3.2.1 Roles and Role-Chains

Mapping constraints can refer to specific units, in either the present representation, or in other representations. The simplest way to refer to another unit is to use a role or role-chain. A simple role refers to an immediate role of the current unit. For instance, *Agent* refers to the Agent role of the present unit. Role-chains are used to refer to roles of roles of the current unit. For instance *Agent.Head* refers to the *Head* of the *Agent* of the present unit. Role chains can be of any length.

The first element in a role-chain can be a *global variable name*, rather than a role. The variable contains the unique identifier of some unit, and that unit is taken as the starting point of the role-chain. For instance, the global variable *Speech-Act* is defined to hold the unit-id of the current speech-act. Mapping rules can thus refer to this variable to access the speech-act. For instance, **Speech-Act*.Speaker* refers to the speaker of the current speech-act (Penman uses a dedicated role called *Onus* for this purpose).

Semantic Referents: Each grammatical unit is associated with an ideational unit - the *semantic referent*, or *semantic correlate*, of the grammatical unit. The grammatical unit encodes the meaning of its referent. Each grammatical unit thus has a role called *Referent*, which points to the referent. Sub-roles of the referent can be indicated using a role-chain, e.g., *Agent.Referent* refers to the *Referent* of the *Agent* of the present unit.

3.2.2 Type Specification (Preselection)

The main operator used in inter-stratal mapping performs *type specification*. Systemics calls this operator *preselection*, but this term is a direction-biased metaphor, assuming some already-existing unit forces a choice in a unit yet to be constructed. I label the operator *:type* instead, since this is direction-neutral, merely a requirement that the specified role has the specified feature, e.g.,

```
(:type Referent addressee-oriented)
```

The feature specification can be logically complex, using a combination of conjunction, disjunction and negation of features, e.g.,

```
(:type Referent (:or not-addressee-oriented
                nondirected-action
                mental-inactive))
```

3.2.3 Identity Requirement

The *:same* operator requires that its two arguments - each a role or role-chain - are filled by the same unit. If they are not already the same, they are conflated. The standard Systemic name for this operator is *conflate*, which I have replaced because it is based on a process metaphor. There are two main uses of this operator in inter-stratal mapping:

1) **Setting the referent of constituents**, e.g., the following constraint asserts that the Referent of the Subject role should be the Actor of the clause's Referent (see figure 6.7):

```
(:same Subject.Referent Referent.Actor)
```

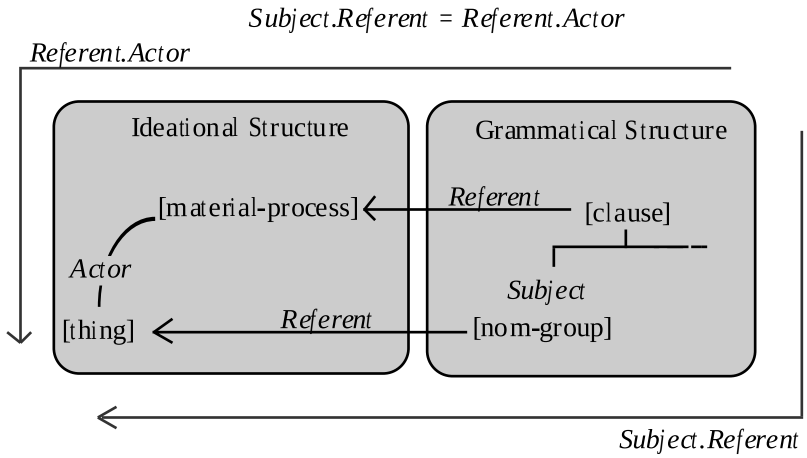

Figure 6.7: Referring to the Same Unit Through Different Role Chains

2) **Checking whether a role in one metafunction plays a particular role in another metafunction**: for instance, we may wish to check if a particular Ideational entity plays some part in the Interactional structure (e.g., *Speaker*, *Hearer*, *Required*), or Textual structure (e.g., *Theme*). The following rule checks whether the Referent of the grammatical-unit (an ideational unit) is the Speaker (see figure 6.8):

```
(:same Referent *Speech-Act*.Speaker)
```

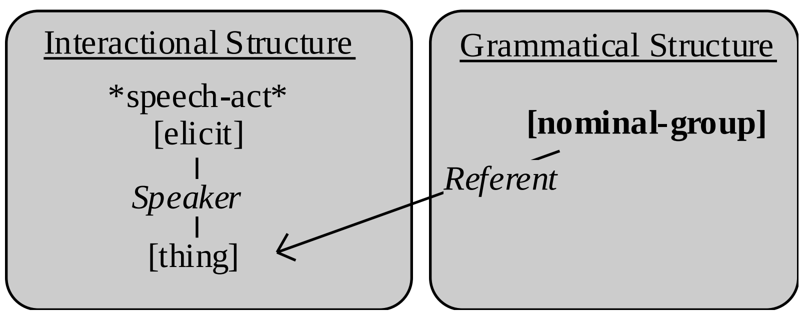

Figure 6.8: Checking Interactional Status of a Referent

3.2.4 Existence Requirement

An *:exists* constraint returns true if the nominated role is already specified, e.g.,

```
(:exists Referent.Location)
```

This is used to check if particular roles have been asserted. Note that we cannot just use the insert/require operator, since this would create the role if it didn't exist already.

3.2.5 Complex Constraints

A constraint can be logically complex, using the following logical connectors:

:and All the constraints conjoined by this operator must be true, e.g.

```
(:and (:type Referent material-process)
      (:type Referent.Actor singular-thing) )
```

:or At least one of the constraints conjoined by this operator must be true, e.g.,

```
(:or (:same Referent.Actor *Speech-act*.Speaker)
     (:same Referent.Actor *Speech-act*.Hearer) )
```

:not the constraint must be false, e.g.,

```
(:not (:same Referent *Speech-Act*.Speaker))
```

4. Some Descriptions

I will provide examples of some selection conditions on lexico-grammatical features. We have already looked at an example of ideational constraint on the lexico-grammar (the *Recipiency* system), so I will provide examples of interactional and textual constraint.

4.1 Mapping Speech-Function and Grammar

In this example, I will show how the some clausal systems are constrained by the interactional structure. Figure 6.9 shows some systems involving the type of clause, its finiteness, and the Mood. The box under each of the features shows not its realisations, but the selection condition (semantic constraint) of the feature. The constraints are glossed in Table 6.1.

The constraints do not handle the full range of mappings between speech-function and lexico-grammar (although the resources can be extended to handle them). For instance, indirect speech-acts are not dealt with. It is also possible to ask yes-no questions using a declarative form, but selecting a rising tone, e.g., “John is coming?”. Other extensions to the resources are also possible.

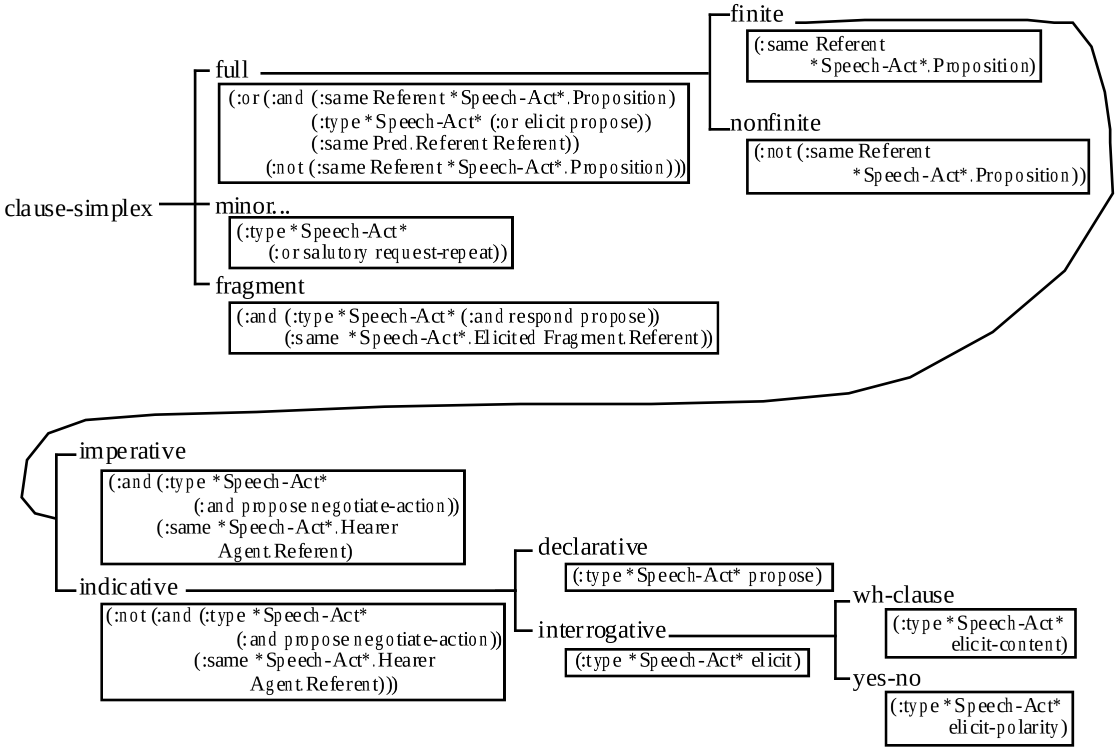

Figure 6.9: Interactional Constraints on Mood Systems

Feature	Constraint Gloss
full	the referent fills the Proposition role of the current speech-act, and the speech-act is eliciting or proposing. Alternatively, the process or relation being expressed is an embedded part of the speech-act's proposition.
minor	the clause is expressing a salutation (e.g., "hello", "thank you"), or is requesting the prior move be repeated (e.g., "Sorry?").
fragment	the clause is expressing a partial response to a question.
finite	the clause is expressing an eliciting or proposing speech-act.
nonfinite	the clause is expressing an embedded process or ideational relation.
imperative	the clause is expressing a proposal of action, where the intended performer of the action is the Hearer.
indicative	the clause is expressing not expressing a proposal of action, or the intended performer is not the Hearer.
declarative	the clause is expressing a proposing move.
interrogative	the clause is expressing an eliciting move.
wh-clause	the clause is expressing a move eliciting content.
yes-no	the clause is expressing a move eliciting polarity.

Table 6.1: Speech-Functional Constraints on Clausal Systems.

These constraints all contain a reference to **Speech-act**, which is interpreted as the current speech-act (the one being generated or analysed). The type of the speech-act constrains the availability of the various clause features. To facilitate interpretation of the constraints, a fragment of the speech-act network is repeated here, in figure 6.10.

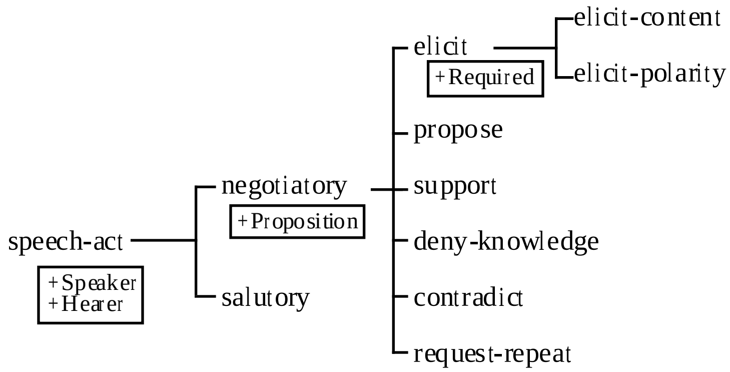

Figure 6.10: Speech-Act Systems which Constrain Mood

It is not only the speech-act *type* which constrains the Mood systems. Often, the constraints need to refer to roles of the speech-act, for instance, to see if the Speaker is also the Actor of the proposition (part of the constraint on an imperative). Another constraint checks to see if this clause is the main clause of the speech-act (expressing the Propositional role of the speech-act). It might instead be realising some sub-unit of the proposition, an embedded clause.

4.2 Mapping Information Status and Grammar

In chapter 5, I discussed the *sharedness* and *recoverability* of ideational entities:

sharedness: is the entity is known to both the speaker and hearer?

recoverability: is the entity part of the immediate discourse context, either through being mentioned earlier in the discourse, or through being in the immediate physical environment of the text production (including the speaker and listener)?

The way in which an entity is expressed is largely determined by these textual concerns. I will briefly explore how they constrain nominal-group structure. Figure 6.11 shows some systems of the WAG nominal group network. The selection condition for each feature is also shown, and glossed in table 6.2. Nominal-group systems not conditioned by information status are not shown, including those concerning pre-modifiers (Numerator, Epithet and Classifier), post-modifiers (Qualifier, Genitive-Marker), pronominal-type, person, case or number selection.

This discussion does not present a complete model of reference, it is only intended to demonstrate the inter-stratal mapping formalism. Noted lacks of this description are:

- **Substitution and Ellipsis:** (cf. Halliday & Hasan 1976; Halliday 1985; Martin 1992) Substitution is not considered at all in the WAG system at present, while only limited ellipsis is handled.

- **Generic and Specific Reference**: Martin (1992) makes a distinction between *specific* references -- references to a particular entity, collection of entities or mass -- and *generic* references -- references to a class of entities. In Martin (1992)'s words: "Generic reference is selected when the whole of some experiential class of participants is at stake rather than a specific manifestation of that class." (p103). Taking the generic/specific nature of a mention into account further constrains the nominal group availability (each type allows different referential strategies). While this distinction is handled in the WAG system, I have ignored this distinction in this example, to simplify discussion.

The constraints make use of three operators discussed earlier in this chapter: *:same*, *:type* and *:exists.* These operators are used in the following way:

:same to test whether the referent of the nominal-group is the speaker or hearer; or perhaps the wh- element.

:type to test whether the thing being expressed is either a *substance-thing* (e.g., *water*), or a *plural-thing* (e.g., *cats*). Under these constraints, a common noun can be mentioned without a deictic.

:exists to test whether a Name role is defined for the thing being expressed. Expression using a proper noun is only appropriate in such cases.

Two textual operators were introduced in chapter 5, but have not so far been discussed in this chapter:

:recoverable testing whether the referent is a recoverable item, i.e., has been mentioned previously in the discourse, or is part of the immediate context of the interaction.

:shared-entity testing whether the referent is part of the shared knowledge of the participants, and thus a candidate for identifiable reference.

The Nominal-Group systems are not totally constrained by the constraints shown here. In some instances, they may allow two or more paths through the network, alternative reference strategies. For instance, after a participant has been introduced to the discourse, *pronominal*, *common* or *proper-name* mention is appropriate. The actual choice within these available options is decided on the basis of textual or contextual systems not so far considered, e.g., stylistic preference.

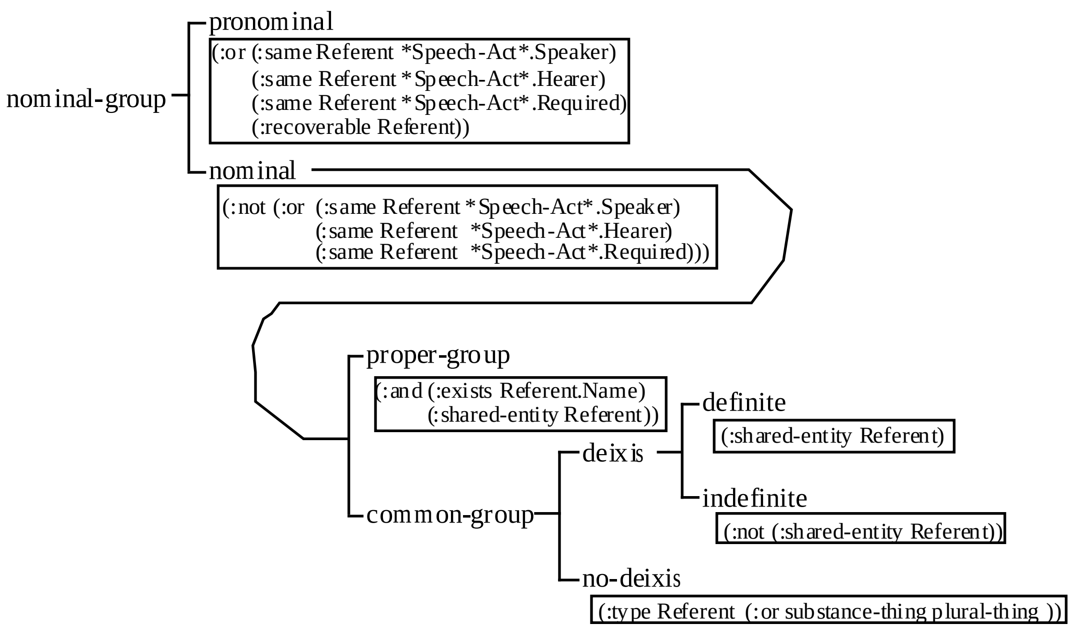

Figure 6.11: Nominal Group Systems with Textual Constraint

Feature	**Constraint Gloss**	**Examples**
pronominal	allowed if the referent is: • the speaker; • the hearer; • the wh- element in an elicitation; • recoverable from the discourse context.	I, me you where, who, what she, he, it, them
nominal	allowed <u>unless</u> the referent is: • the speaker; • the hearer; • the wh- element in an elicitation; Note that recoverable entities may be realised nominally as well as pronominally.	John, the cat, cats, some cats
proper-group	allowed if the referent has a Name field, and is part of the shared-information.	John, Mary
common-group	allowed in all situations that *nominal* is allowed.	the boy, boys
no-deixis	allowed if the referent is not recoverable, and either plural or a substance-thing.	Cars, People, Water, Wheat
deixis	allowed if *common-group* is allowed.	the person, a person, some people
definite	allowed if the referent is a shared-entity (note: the selection constraint needs extension to allow entities *related* to shared entities to be expressed using definite reference when the relationship is expressed).	the woman, the boys, that boy.
indefinite	allow <u>unless</u> the referent is identifiable.	a woman, some boys.

Table 6.2: Textual Constraints on Nominal Group Systems

5. Some Extensions

This section introduces two potential extensions to the semantico-grammatical mapping approach described above.

5.1 Contextual Resolution

Lexico-grammatical form is not totally constrained by the semantics: alternative realisations of a given semantic configuration are often possible. In a Systemic-Functional model, we need to consider the *context* of the text to determine which of the alternatives is appropriate. According to Halliday (1973):

> "Where there are alternatives, such that a given semantic option is realized EITHER by this or that set of features in the grammar, these are often determined by the environment. For instance, the grammar of personal medicine - the language used to describe one's ailments - is not the same in the doctor's consulting room as it is in a family or neighborhood context. ... The environment is generally the higher-level environment, either the immediate paradigmatic environment - that is, the other options that are being selected at the same time - or the social context as a whole." (p94).

There has been some work towards the contextual constraint on the lexico-grammar, for instance, Bateman & Paris (1989a, 1989b); Paris & Bateman (1990) and Cross (1991). Each of these studies has shown how context can be used to further constrain the interstratal mapping. However, these studies have so-far involved only small-scale application.

5.2 Mapping Mediated by an Inter-Stratal Network

In recent years, there has been some movement to introducing a resource between the semantics and the lexico-grammar, to organise the mapping between these strata. Matthiessen (1987b) pushes for a network organisation of this resource:

> “there are clear indications that inquiries could profitably be organised into an independent, global network of inquiries rather than the chooser packages that are localised with respect to grammatical systems.” (p17).

Bateman (1988) takes a first step towards such a network, introducing choosers which are not linked to any one system, but able to determine feature selections in a number of systems. Matthiessen & Bateman (1991) proposes a further step, discussing the organisation of inquiries in networks rather than trees -- allowing complex entry conditions, and simultaneous systems. This proposal is, however, still theoretical.

Cross (1992) reports on her *Horace* system, which includes an interstratal network, called the *semantic-lexicogrammar preselections resource* :

> "The semantic-lexicogrammar preselections resource provides the interface between the semantics of the text and the lexicogrammar. Traversal of the system network of the semantic-lexicogrammar preselections resource determines the passes and the preselections in the lexicogrammar and associates mapped rhetorical functions with grammatical functions." (Cross 1992, p17).

Wanner (1994; in preparation) also has proposed intermediate networks, which are used for mapping between the semantic representation and the grammar. Distinct networks are provided to handle the mapping of each metafunction (experiential, logical, interactional and textual). For instance, the experiential mapping network determines which of the ideational roles are to be expressed, maps them onto grammatical units, and determines the type of grammatical item to use.

The main advantage of representing the inter-stratal mapping information in an inter-stratal network is mainly that it reduces redundancy. When constraints are attached to individual features, or, as in the Penman case, to systems, often the same constraint is called upon in several places. During generation, the same complex decisions need to be made over and over again (see Bateman (1988) for a good example). By separating the mapping decisions off into a separate resource, representing each decision once only, we reduce the redundancy. The grammar is given access to the decision made in the inter-stratal resource. This resource can be organised in terms of a system network -- each feature of the network representing a mapping decision. A path through the inter-stratal network provides a selection expression, summarising the mapping decisions. The selection conditions in grammatical systems can refer to this selection expression, rather than having to re-make the decision each time. The selection conditions are thus simplified.

5.2.1 An Experimental Inter-Stratal Network

As an experiment, I implemented an inter-stratal network along the lines of Wanner's network -- a level at which inter-stratal mapping decisions are made. These decisions concern mainly which ideational roles to express (content selection), and what grammatical form they are to be expressed in (e.g., should a process be expressed as a clause, or as a nominalisation). Note that this inter-stratal network represents an optional extension on the WAG resource model -- the discussion in the rest of this thesis assumes that it is not used, that lexico-grammatical features are constrained directly from the micro-semantics.

One pass through the inter-stratal network is made for each element of the ideational structure. The pass determines how the elements of the semantic unit are mapped onto the elements of the grammatical unit.

There is also structure at the inter-stratal level, what I call a *text-structure* (following Meteer 1990). Figure 6.12 shows an example of such a text-structure, and how it relates to the semantic structure. The diagram is divided into three boxes, representing

- **the ideation-base**: the information to be expressed in the text (only part of the whole ideation-base is shown);
- **the interactional-structure of the current move**;
- **the text-structure**: the inter-stratal mapping structure.

Limited textual information is also represented in the diagram: in the ideation-base, non-relevant information is marked in gray, or absent. The text-structure could be interpreted as the rhetorical-structure of the sentence being expressed (a process as the nucleus, with *Agency* and *Origination* satellites). Each unit in the text-structure has a *Content* role, which points to the ideational unit for which it is controlling the mapping. Each textual-unit is also be related to a grammatical unit, but this is not shown here.

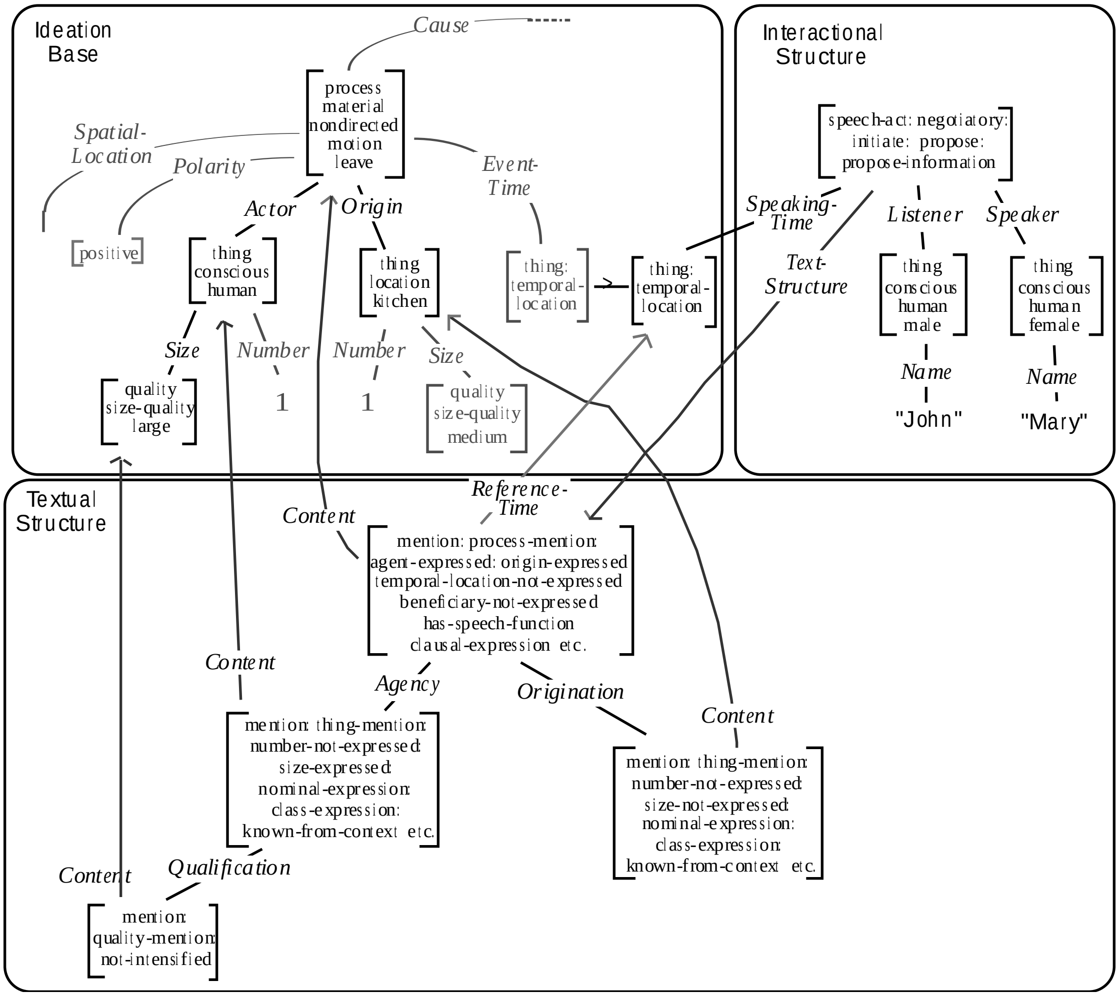

Figure 6.12: Text Structure Realising Semantic Structure

Figure 6.13 shows one of the systems of the network, where the decision is made as to whether a Spatial-location role should be expressed. Each of the features of the system has a semantic constraint which must be met before the feature can be chosen. The features of this interstratal network can each be seen as a single label for a complex structural constraint.

```
(defsystem
  :name spatial-location-expression
  :entry-condition (:or clausal-expression nominal-expression))
  :features ((spatial-location-expressed
              :selection-constraint
                (:and (:type content (:or located-process located-thing))
                      (:relevant Content.Location))
              :realisation (:and (:require Locator)
                                 (:type Locator thing-mention)))

             (spatial-location-not-expressed
              :selection-constraint
                (:or (:type content (:or unlocated-process unlocated-thing))
                     (:not (:relevant Content.Location))))))
```

Figure 6.13: A System of the Text-Structure Network

One advantage of making expression decisions on a level separated from the grammar is that we can generalise decisions -- making decisions which will effect both clauses and groups. The system in figure 6.13 -- determining whether a Spatial-Location is to be expressed -- will have consequences both in a clause (the presence of a Circumstance role), and in a group (the presence of a Qualifier role).

Using this approach, the selection constraints on grammatical features refer to the text-structure, rather than directly to the semantics. The text-structure thus acts as an intermediary, a summarisation of the ideational, interactional and textual information relevant to the construction of the lexico-grammatical form.

This discussion of inter-stratal networks has been phrased in terms of sentence generation. However, given a declarative representation of the network and constraints, the resource could also be used to aid in the semantic analysis of sentences.

6. Summary of Inter-stratal Mapping

This chapter has summarised various approaches to inter-stratal mapping, in particular, contrasting the following approaches:

- Mapping With Semantic Structure vs. Mapping without Structure
- Feature-Based vs. System-Based mapping:
- Semantic-Driven vs. Grammar-Driven vs. Strata-Neutral Mapping:

I have also described the WAG mapping formalism in detail, and demonstrated how it can be used to map between grammar and semantics. The main difference from prior mapping approaches are:

- Constraint on lower stratum, not higher;
- Constraints on features, not on systems;
- Syntagmatic mapping, not just paradigmatic: preselection is thus only one mechanism used for mapping:

  - Conflation (identity checking/asserting)

  - Insertion (role-requirement)
- Uses role-chains, not just immediate roles

This formalism has been illustrated using examples drawn from three areas of the grammar:

- **Ideational Constraint**: showing how ideational units are mapped onto grammatical units;
- **Interactional Constraint**: showing how the speech-functional systems constrain the various clausal systems;
- **Textual Constraint**: showing how information status constrain the options in the nominal-group.

Inter-stratal networks, which can be used to organise the mapping between grammar and semantics, were introduced and discussed, and an example given of such a network implemented in the WAG system.

6.1 Evaluation of Feature-Based Constraints

The Feature-based constraint system used in WAG (following Kasper's unpublished design) has various advantages over Penman's system-based constraints:

- **Conciseness**: Feature-based constraints are far smaller than the choosers and associated inquiries.
- **Directness**: The grammatical features are linked directly to the semantic structure, rather than, as in Penman, mediated by several intermediate steps:

  Penman: system -> chooser -> inquiry -> semantics;

  WAG: feature <-> semantics.
- **Clarity**: The feature constraints, being smaller and directly related to the semantics, are easier to interpret. For instance, in the Recipiency example above, we can see clearly the appropriate 'semantics' of the *recipiency/non-recipiency* choice.
- **Ease of Modification**: The WAG constraints show all of the constraint in one place, while constraints in the Penman system are scattered across several data-structures. Modification of the mapping resource is thus easier in WAG.
- **Declarativeness**: Because they are totally declarative, and direction-neutral, the WAG constraints can be transferred between platforms, and are re-usable for different processes.
- **Bidirectionality**: WAG's constraint system can be (and is) used for both generation and analysis.

This ends Part A, outlining of the resource model. Part B will show how these resources are used in computational processing.