Bilingual Generation of Job Descriptions from Quasi-Conceptual Forms*

David E. Caldwell
Tatiana Korelsky
CoGenTex, Inc.
ted@cogentex.qc.ca
tanya@cogentex.com

Abstract

The EXCLASS system (Expert Job Evaluation Assistant) is intended to provide intelligent support for job description and classification in the Canadian Public Service. The Job Description Module (JDM) of EXCLASS is used to create conceptual representations of job descriptions, which are used for job evaluation and bilingual generation of textual job descriptions. The design of these representations was subject to two opposing constraints: (1) that they be deep enough to resolve the ambiguities present in textual job descriptions, and (2) that they be close enough to surface linguistic forms that they can be conveniently manipulated by users with little specialized training. The close correspondence of concepts to surface words and phrases, as well as properties of the job description sublanguage, permit a simplified generator design, whereby phrases are prepackaged with a certain amount of linguistic structure, and combined according to a small set of mostly language-independent rules. Text planning, consisting mainly of grouping and ordering of conjoined phrases, is performed manually by the user, and composition of conceptual forms is supported by a "continuous text feedback" function.

1. Goals of EXCLASS

The EXCLASS system (described on a more general level in Korelsky & Caldwell 1993) is intended to provide intelligent support for the process of describing and evaluating jobs in the Canadian Public Service. The Job Description Module (JDM) of EXCLASS, developed by CoGenTex for the Canadian Treasury Board, provides resources for the user to compose conceptual representations of job descriptions. The JDM generates textual job descriptions in both English and French from these representations; a Job Evaluation Module (JEM) also reasons on them to produce a classification and rating of a job, according to the government's evolving Universal Classification Standard.

The first phase of the EXCLASS project resulted in a proof-of-concept prototype, based on a sample of some 30 job descriptions in the domain of procurement and asset management, in which the JDM and JEM are linked through a common graphical interface. The second phase, concluded in the spring of 1994, involved R&D in preparation for fielding and site testing of the system in a selected government department.

EXCLASS is intended to eventually be used by thousands of managers across Canada, thus decreasing reliance on classification experts, while at the same time increasing the standardization, objectivity and comparability of job classifications across diverse occupational and organizational groupings.

2. Functional Requirements

The principal task of the JDM is to produce an unambiguous conceptual representation of a job description, which is suitable for (1) automatic reasoning by the job evaluation component, (2) bilingual text generation, and (3) manipulation by users with little or no training in knowledge representation. It must also provide various tools to facilitate such manipulation, and it must do this on a 386-class PC under Microsoft Windows.

In the current standard format, public-service job descriptions consist of three basic types of statements, which describe a position in progressively greater detail: Client Service Results, Key Activities, and Substantiating Data. Substantiating Data is further classified into various Factors and Elements, e.g. Working Conditions: Environment, Risk to Health; Skill and Knowledge: Physical Demands, Communications. Figure 1 shows a sample of the job description format.

```
CLIENT-SERVICE RESULTS
• Procurement of military aircraft and airframes
    for the Department of National Defense.
KEY ACTIVITIES
• Issuing invitations to tenders and requests for
    proposals.
• Conducting negotiations with sole-source sup-
    pliers.
• Preparing and issuing contracts within own au-
    thority and recommending approval of con-
    tracts in excess of own authority.
SUBSTANTIATING DATA

...
Environment
• The work involves an office environment, re-
    sulting in frequent use of computers and oc-
    casional exposure to noise. Some travel is
    required.

...
```

Figure 1: Sample job description text.

* We are grateful to Ehud Reiter for his valuable comments on an earlier version of this paper, which greatly influenced its present form.

Results and Key Activities are expressed in point form; Results as nominal phrases, and Key Activities as gerunds. Substantiating Data statements are sometimes multi-sentential, but tend to follow fairly rigid templates.

A comprehensive analysis of user requirements for the JDM was conducted, during which it became clear that users favoured more explicit control over all aspects of the content of a job description, even if it came at the expense of convenience of composition. The idea of prepackaged templates as a basis for conceptual job descriptions—for example, classifications of Key Activities likely to be associated with department heads, middle management, clerical staff, etc.—met with some resistance, since it might prejudice the outcome of job evaluation. Users also expressed a desire for a convenient means of adding to the collection of concepts available, in the event that they did not find what they needed for a particular job description.

3. Functionality

The EXCLASS JDM comprises two modules: the Job Description Builder (JDB) and the Job Description Generator (JDG). The JDB supports composition and editing of conceptual representations, which take the form of trees of concepts drawn from a structured conceptual dictionary. The JDG produces text from these representations, by combining realization templates associated with each concept. The next three sections describe the conceptual dictionary, conceptual forms, and the structure of the generator.

3.1 Knowledge Representation

The dictionary of concepts used in the JDB to compose conceptual representations comprises several disjoint hierarchies of entities which figure in job descriptions. The current dictionary covers a sample of some 30 job descriptions in English and French, although the analysis on which it was based encompassed at least twice that number.

In order to determine just what the entities represented in the conceptual dictionary should be, we began with the following criteria, which derive from the functional requirements:

1. In order to provide a basis for suitable input to the Job Evaluation Module and the Job Description Generator, concepts should be free of the ambiguities observed in textual job descriptions. These ambiguities have three main sources:
    - multiple word senses;
    - attachment of dependent phrases;
    - scope of conjunction.
2. In order to allow managers, who have little or no training in knowledge representation, to work with conceptual representations at the most detailed level, concepts should introduce as little specialized notation as possible.

The first criterion calls for concepts which are abstracted from surface linguistic forms, while the second says that they should be close to surface forms, since that is what managers are accustomed to working with when they write job descriptions.

In order to satisfy these conflicting criteria, concepts were designed to correspond to surface words or phrases as closely as possible, while remaining free of ambiguities. Concepts corresponding to different senses of the same word are annotated with distinguishing labels—e.g. *negotiation [activity]* (as in *negotiating price and cost elements for multi-phase contracts*) vs. *negotiations [process]* (as in *conducting negotiations with sole-source suppliers*). Concepts corresponding to surface forms which take dependent phrases are associated with semantic roles (see below). And concepts contain only irreducible conjunctions (e.g. *The Banff National Park and region*).

With regard to the appropriate granularity of concepts, again there were conflicting criteria:

3. Concepts should be fine-grained enough to permit users to express the distinctions that are important to them.
4. Concepts should be coarse-grained enough that editing of conceptual representations is not more burdensome than editing text.

Again, the approach adopted was to make concepts just fine-grained enough to account for collocational patterns observed in the corpus (through analysis of concordances).

The conceptual dictionary is structured using a representation similar to KL-ONE (Woods & Schmolze, 1992). Concepts are arranged in hierarchies from most general to most specific, and associated with semantic roles and "structural conditions" on those roles. For example, the concept *negotiations [process]* is a child of ("a kind of") the concept *interactions*, and has roles for the action involved (e.g. *conducting, leading*), what is being negotiated (e.g. *contracts, agreements*), and who is being negotiated with (e.g. *suppliers, foreign government representatives*).

The structural conditions on a concept's roles are expressed partly in terms of a division of the set of concepts into subsets of different types:

- *Object* concepts (e.g. *resources, systems for secure storage, special inventory counts*), which can serve as roots of conceptual forms (see the next section).
- *Domain* concepts (e.g. *asset management, warehousing, custodial warehousing*), which correspond to occupational groupings.
- *Body* concepts (e.g. *Canadian Parks Service, industry sales representatives, other service providers*), which denote types of individuals or corporate entities.

- *Location* concepts (e.g. *Prairie Region, National Capital Region Supply Centre*).
- *Purpose* concepts (e.g. *to ensure adequate service, to ensure that all aspects of contracts have been completed*).
- *Action* concepts (e.g. *developing, maintaining, approving*).

Object concepts form a hierarchy descending from the most general concept of *service* (they are also referred to as "aspects of service"). There are separate hierarchies for domains, bodies, and locations; purposes and actions are not hierarchically structured. In general, it is object concepts that have roles, which are filled by concepts of appropriate other types. The structural conditions on roles taking values from one of the hierarchies list a default (most typical) value for the filler, as well as a most-general possible value. When values come from a non-structured set, such as actions, the structural conditions consist of a list of possible values.

The conceptual dictionary is also structured according to occupational domains. Concepts peculiar to certain domains are marked with features corresponding to those domains—for example, *contracts* is a procurement concept; *materiel handling equipment* is a warehousing concept.

The "aspects of service" hierarchy is based not just on "kind of" relations, but also "aspect of" relations—for example, *multi-phase contracts* are a "kind of" *contracts*, whereas *operational costs* are an "aspect of" *operations*. Inheritance of concept roles and attributes through "kind of" links is used as the basis of the concept acquisition interface (see the last section), although it is not used for retrieving concept data. The exact nature and implementation of inheritance on "aspect of" links is a topic for future research.

3.2 Conceptual Forms

In order to compose and edit representations of job descriptions, the user works with conceptual forms. A conceptual form is a tree of concepts, whose arcs correspond to semantic roles associated with concepts. Visually, concepts in trees are presented as frames with slots named for semantic roles, into which the user can insert other concepts. This was seen as the best way of giving users control over the most detailed aspects of conceptual representations, while keeping their visual presentation relatively simple.

An example of the conceptual form of a Key Activity is shown in Figure 2. The MAIN CONCEPT slot of the Key Activity frame takes one or more "aspect of service" concepts as values. The frame for a Result statement corresponds to the central concept *service*, with slots for NATURE OF SERVICE and CLIENT OF SERVICE.

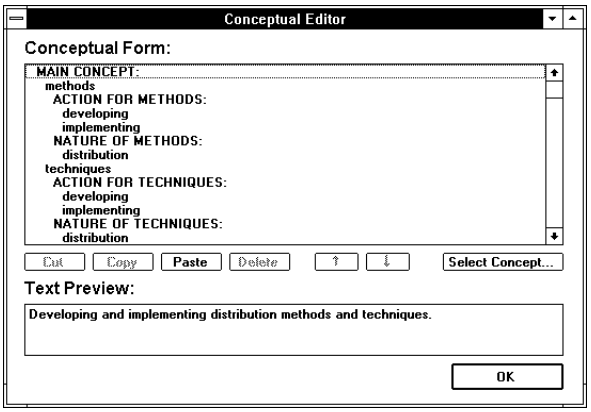

Figure 2: Example of a conceptual form.

The basic editing operation for constructing conceptual forms is to highlight a slot, then select a concept to go in that slot. For slots taking values from hierarchically-structured subsets of the vocabulary, such as objects or locations, the user can browse through the relevant hierarchy, subject to the conditions described earlier (Figure 3). The concept browser shows a "focused" concept, together with its parents and children; the user moves up or down by shifting the focus to a parent or child (a Find Concept function is also available). When values are from a non-structured subset (e.g. actions), selection is from a flat list of possible values.

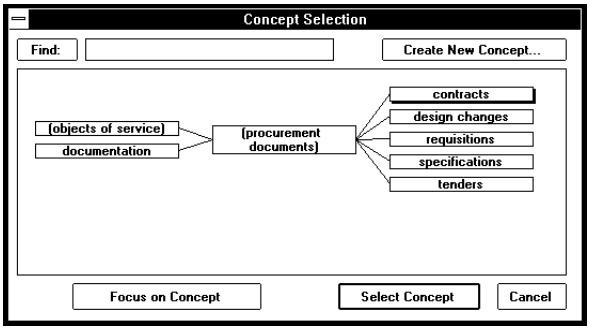

Figure 3: The concept browser.

Editing of existing conceptual forms is supported by cut, copy and paste functions, which operate on subtrees of conceptual forms. The same operations are defined for whole statements, so that users can move conceptual structures of any size within the same job description, or between different ones.

A notable feature of conceptual forms is that, contrary to usual linguistic practice, object concepts (which in general correspond to grammatical direct objects) are the roots, while action concepts are the dependents. The rationale behind this is that it is relatively straightforward to structure objects into a reasonably deep, exhaustive, and intuitive hierarchy, whereas this would be very difficult for actions. The set of actions can be im-

plicitly structured, however, by constructing lists of actions appropriate for use with any given object. The reason for structuring sets of concepts is to aid the user in composition, so that s/he only has to choose from a small number of alternative concepts at any one point. So the implicit structuring of actions according to whether they can occur with a given object is only useful if the user selects the object first, and then the actions.

Above the level of conceptual forms for individual statements of various types, there is currently no meaningful representation of a job description as a whole, except that the domains listed under NATURE OF SERVICE in Result statements are used to "trim" the concepts displayed in the browser when composing the rest of the job to only those relevant to those domains. How to represent links or enforce consistency between different statements—in particular between Results/Key Activities and Substantiating Data—is a topic of ongoing research by the developers, and discussion by potential users.

3.3 Linguistic Realization

Given the close correspondence between conceptual forms and surface linguistic forms, we decided to re-examine our initial assumption that the Job Description Generator would be implemented by adapting CoGenTex's existing text-generation shell.

Versions of this generator, based on Meaning-Text Theory (Mel'č uk & Pertsov, 1987), have been used in other applications, including the generation of bilingual weather forecasts (Goldberg et al., to appear) and statistical reports (Iordanskaja et al., 1992). In order to produce text suitable to these applications, the generator starts with deep conceptual representations, successively deriving deep-syntactic, surface-syntactic, morphological, and surface representations. It also incorporates sophisticated mechanisms for text planning and paraphrase.

For several reasons, the existing generator was considered unsuitable for this application. The main rationale was that, since concepts already resembled pieces of surface text, those pieces should not be reconstructed by the generator unless this was necessary to produce text of acceptable quality. If the words and phrases corresponding to concepts could be given just enough linguistic structure that a simplified generator could combine them more or less directly to produce text, then it would be a waste of effort to decompose them to the level of detail on which the existing generator operated, only to regenerate aspects of surface form that were already present in concept labels.

Another factor favouring a simplified generator design was the decision, following the design of the conceptual forms, to include a "continuous text feedback" function in the JDM interface. Again, since users were unaccustomed to working with conceptual representations, it would be useful if they could confirm their choices on the conceptual level with textual feedback from the JDG. The JDM's conceptual editor (Figure 2 above) incorporates a text preview area, which is updated every time a change is made to the conceptual form. It also has the feature of displaying text even for incomplete conceptual forms. The existing generator did not have the level of real-time performance demanded by this feature (on a 386-PC platform), or the ability to generate incomplete phrases.

A simplified generator design was facilitated by certain linguistic properties of job descriptions:

- When statements are not simple clauses, they follow fairly rigid templates. All conjunctions except *and* and *or* can be treated as parts of concepts (e.g. the purpose concept *to ensure that all aspects of contracts have been completed*) .
- Referring expressions are always either generic or proper noun phrases (no pronouns or definite/indefinite distinctions).
- There is very little morphology to deal with—there is no agreement, due to the lack of subjects, and the fact that adjectives and articles can always be treated as part of the same concept as the noun they modify.

Given these facts, all the generator has to do is select different alternatives for realization of concepts in some cases, concatenate phrases, and perform ellipsis resulting from conjunctions. Text planning is performed manually by users—they can order clauses in a Key Activity, or actions for an object, in the same way that they order Key Activities in a job description.

The generator is in the spirit of a Montague-style categorial grammar (Dowty et al., 1981), except that operations of function application and composition, rather than operating on semantic objects in parallel with the concatenation of surface elements, operate in effect on the surface elements themselves. In order to illustrate its operation, consider the conceptual form in Figure 4, which is realized as *Supervising performance of routine and special assignments to ensure adequate service*:

```
key_activity
    MAIN CONCEPT:
        activities of others
            ACTION FOR ACTIVITIES OF OTHERS:
                supervising
            MAIN CONCEPT OF ACTIVITIES OF OTHERS:
                routine assignments
                    ACTION FOR ROUTINE ASSIGNMENTS:
                        performing
                            PURPOSE OF PERFORMANCE:
                                ensure adequate service
                special assignments
                    ACTION FOR SPECIAL ASSIGNMENTS:
                        performing
                            PURPOSE OF PERFORMANCE:
                                ensure adequate service
```

Figure 4: Conceptual form for a complex Key Activity statement

a.

key_activity	→	$\langle MAIN{:}gerund \rangle$
activities of others	→	$\langle ACTION{:}gerund \rangle * \langle MAIN{:}nominal \rangle$
supervising (gerundive form)	→	$"supervising"$
routine assignments (gerundive form)	→	$\langle ACTION{:}gerund \rangle * ("routine" * "assignments")$
performing (nominal form)	→	$\lambda x((" performance" * (" of" * x)) * \langle PURPOSE{:}\_ \rangle)$
ensure adequate service	→	$" to" * ("ensure" * (" adequate" * " service"))$
special assignments (gerundive form)	→	$\langle ACTION{:}gerund \rangle * (" special" * " assignments")$

b.
$$" supervising" * \left| \begin{array}{l} \left(\lambda x((" performance" * (" of" * x)) * ("to" * (" ensure" * (" adequate" * " service")))) * (" routine" * " assignments") \right) \\ \& \\ \left(\lambda x((" performance" * (" of" * x)) * ("to" * (" ensure" * (" adequate" * " service")))) * (" special" * " assignments") \right) \end{array} \right|$$

c.
$$" supervising" * \left| \begin{array}{l} ((" performance" * (" of" * (" routine" * " assignments"))) * ("to" * ("ensure" * (" adequate" * " service")))) \\ \& \\ ((" performance" * (" of" * (" special" * " assignments"))) * ("to" * ("ensure" * (" adequate" * " service")))) \end{array} \right|$$

d.
$$" supervising" * ((" performance" * (" of" * ((" routine" \& " special") * " assignments"))) * ("to" * ("ensure" * (" adequate" * " service"$$

Figure 5: Steps in the derivation of a Key Activity statement

Each concept in the dictionary is associated with one or more realization templates, which are complex expressions built up from surface words or phrases, certain operators, and variables corresponding to the concept's slots[1]. The relevant English templates for the concepts in Figure 4 are shown in Figure 5a.

Expressions of the form <SLOT:type> specify how the contents of a slot are to be realized—i.e., using which of the available templates. For example, a Key Activity frame is realized by realizing the contents of its MAIN CONCEPT slot as a gerund. The *activities of others* frame, which essentially represents a Key Activity embedded within another, is realized by concatenating the gerundive form of its action with the nominal realization of the embedded frame. The first step the generator performs is to instantiate these expressions to the correct forms, and conjoin multiple fillers of a single slot with the & (*and*) operator, resulting in the form in Figure 5b. The next step is to reduce lambda expressions, which gives 5c. Ellipsis is then performed, giving the form in 5d. Finally, occurrences of the & operator are lexicalized as either commas or *and*, as appropriate.

The operators used in realization templates, other than λ and &, serve to represent structure which is consulted by the rules for lambda reduction and ellipsis. Lambda reduction of an expression $\lambda x(A) * B$ gives a copy of A in which all occurrences of x (usually one) are replaced with B. This is used for a "wrap" effect in cases where actions have dependents, as well as in nominalizations—in these cases the dependence of actions on objects in conceptual forms cannot be undone simply by concatenating the action to the left of the object. The lambda notation is also used to specify connecting phrases (usually prepositions) which are associated with the slots of certain concepts, and introduced by the generator—for example, in realizing the phrase *negotiations with contractors*, the preposition *with* is introduced by concatenating the connecting-phrase expression $\lambda x(" with" * x)$ associated with the NEGOTIATIONS WITH WHOM slot to the left of the slot's realization, "*contractors*". When slots are empty, the connecting phrase is omitted—this is mainly what accounts for the generator's ability to produce incomplete phrases (in some cases, conceptual forms with empty slots can produce acceptable phrases).

The basic rules for ellipsis are $(A * B) \& (A * C) \Rightarrow A * (B \& C)$ and $(A * C) \& (B * C) \Rightarrow (A \& B) * C$. There are other rules which optimize conjunctions to some degree by reordering conjuncts, but the overall approach is to let users control the order manually. An operator # is used in place of $*$ to block ellipsis, and an operator $\backslash$ handles cases in French where an *OR* is introduced during ellipsis, according to the rules $(A \backslash B) \& (A \backslash C) \Rightarrow A * (B / C)$ and $(A \backslash C) \& (B \backslash C) \Rightarrow (A \& B) / C$ (the / operator is lexicalized as "*ou*"). For example,

$$\begin{array}{l} ((" les" \# " contrats") \backslash (" à" \# (" fournisseur" * " unique"))) \\ \& \\ ((" les" \# " contrats") \backslash (" à" \# (" étapes" * " multiples"))) \end{array}$$

is realized as *les contrats à fournisseur unique ou à ètapes multiples*, and not as *les contrats à fournisseur unique et à ètapes multiples* or *les contrats à fournis-*

[1] There are rules in some cases for deriving variant templates for a concept from a basic template. For example, the gerundive (basic) template for an object concept in general has the form <*ACTION:gerund*>* …; the nominal form is derived from this simply by specifying the nominal form of the action.

seur unique ou ètapes multiples.

Grammatical differences between French and English are dealt with by assigning different structures, sometimes using different operators, to the English and French templates for a given concept, but there are also cases where the lexicalization of a concept depends on another concept in the context—for example, *performing special assignments* translates as *executer les affectations spéciales*, whereas *performing post-contract cleanup* translates as *assurer le suivi des contrats*. These cases are modelled using the MTT notion of *lexical functions*—in this example, the values in English and French of the **Oper$_1$** function ("verb denoting the most typical action of the first actant") are *performing* and *executer* for the concept *special assignments/affectations spéciales*, and *performing* and *assurer* for the concept *post-contract cleanup/suivi des contrats*. Lexical functions are implemented in the conceptual dictionary as "virtual" concepts, with pointers to actual concepts for each language. Users can switch the language in which conceptual forms are displayed (independently of the language in which text is generated), and when they do so, the appropriate actual concepts are displayed, with no explicit indication of the underlying virtual concept. This means, for example, that a user could copy the concept *assurer* from the ACTION slot of *suivi des contrats*, and paste it into the ACTION slot of *affectations spéciales*, whereupon its label would change to *executer*.

The generator design described in this section has several advantages for this type of application:

- It takes full advantage of the similarity of concepts to surface linguistic forms, which was dictated by the functional requirements. Phrases are generated as chunks wherever possible, while still being assigned enough linguistic structure to produce adequate text.
- Given the large volumes of concepts anticipated, maintenance of realization templates will presumably be simplified if they do not refer to lexical entries in a main dictionary, and if a constrained grammatical formalism is employed.
- Incomplete phrases can be generated straightforwardly, in order to support the text preview function.

4. Research Topics

The main concern for deployment of EXCLASS on a large scale is how to deal with the large volumes of concepts which will be required. A concept acquisition interface has been designed to support expansion of the dictionary.

The acquisition interface is invoked from the concept browser, when the user has determined that the desired concept is not already available. The user selects a concept from the browser to be the parent of the new concept in the relevant hierarchy. The attributes of the new concept (label, slot types and possible values, realization templates) can then be edited, starting with default values. The defaults are inherited from the parent concept, on the assumption that the new concept is a "kind of" the parent. The nature of inheritance through "aspect of" links is a topic for future research.

Another topic of research is how to possibly enrich representations of a job as a whole, as well as of individual concepts. The JEM developers are experimenting with comparisons of job descriptions based on fuzzy distance measures, which in turn are based on the positions of individual concepts in the hierarchy. Action concepts are difficult to compare, since they are currently unstructured. Adding some sort of structure, such as ranking the possible actions for a given object, could facilitate job comparison, as well as treating linguistic phenomena such as “asymmetric” conjunction (*developing and implementing methods* vs. **implementing and developing methods*).

Finally, research is being conducted on different usage modes for the JDM interface—in particular, an “expert” mode in which the user could enter the text of simple (non-conjoined) statements and have it parsed to some extent (using an elaborated "find" function) into a conceptual form, rather than performing repetitive point-and-click operations.

References

David Dowty, Robert Wall, and Stanley Peters. *Introduction to Montague Semantics*. Dordrecht: Reidel, 1981.

Eli Golberg, Richard Kittredge, and Norbert Driedger. A new approach to the synthesis of weather forecast text. To appear in *IEEE Expert*. (Special Track on Processing Natural Language)

L. Iordanskaja, M. Kim, R. Kittredge, B. Lavoie, and A. Polguère. Generation of extended bilingual statistical reports. In *Proceedings of the 14th International Conference on Computational Linguistics (COLING-1992)*, volume 3, pages 1019-1023, 1992.

Tatiana Korelsky and David Caldwell. Concept-based composition of job descriptions with automatic text generation. In *Proceedings of the Canadian DND Combined Workshop in Advanced Technologies*, Ottawa, November 1993.

Igor Mel'č uk and Nikolaj Pertsov. *Surface Syntax of English: A Formal Model within the Meaning-Text Framework*. Amsterdam: John Benjamins, 1987.

William Woods and James Schmolze. The KL-ONE family. *Computers and Mathematics with Applications*, volume 23, no. 2-5, pages 133-177, 1992.