NOTICE WARNING CONCERNING COPYRIGHT RESTRICTIONS:
The copyright law of the United States (title 17, U.S. Code) governs the making of photocopies or other reproductions of copyrighted material. Any copying of this document without permission of its author may be prohibited by law.

Technical Description of KADBASE

by

H.C. Howard and D.R. Rehak

EDRC-12-20-88

UNIVERSITY LIBRARIES
CARNEGIE-MELLON UNIVERSITY
PITTSBURGH, PENNSYLVANIA 15213

TECHNICAL DESCRIPTION OF KADBASE

Technical Report
to accompany

INTERFACING DATABASES AND KNOWLEDGE-BASED SYSTEMS
FOR STRUCTURAL ENGINEERING APPLICATIONS
(EDRC 12-06-86)

by

H. C. Howard and D. R. Rehak

Department of Civil Engineering
Carnegie-Mellon University
Pittsburgh, Pennsylvania
January 1988

Table of Contents

UNIVERSITY LIBRARIES
CARNEGIE-MELLON UNIVERSITY
PITTSBURGH, PENNSYLVANIA 15213

List of Figures

List of Tables

Chapter 1
Introduction to the Technical Report

This report is intended to supplement the information provided in the accompanying thesis, *INTERFACING DATABASES AND KNOWLEDGE BASED SYSTEMS FOR STRUCTURAL ENGINEERING APPLICATIONS* [Howard 86]. This document is not independent of the thesis; the reader is assumed to have some familiarity with the material in the thesis and to have it available for cross-reference.

The report is divided into the following chapters:

- **KADBASE Query Language** — defines the data manipulation language for the KADBASE user interface. KQL provides the basis for the message data structure used to represent queries and updates in KADBASE.
- Message **Data Structure** — describes the internal frame-based representation for requests and replies in KADBASE components. The chapter contains a complete set of frames for a sample SPEX query.
- **KADBASE Schema Description Details** — defines the frame-based schema description components that support semantic mapping in KADBASE. The chapter also describes the schema description utility that supports the input of schema descriptions using a problem-oriented language.
- **KADBASE Communications** — describes the KADBASE communication utilities and how they operate.
- **SPEX Supplement** — contains the complete schema definitions for SPEX, the AISC database, and the global schema as well as transcripts showing the operation of SPEX and the AISC KBDBI.
- **HICOST-II Supplement** — contains the complete schema definitions for HICOST-II and its three databases, the data in the databases before and after HICOST-II execution, and the rules used in the takeoff process.
- **JADE Supplement** — contains the complete schema definitions for JADE, the structural configuration database, the analysis results database, and the global schema.

Chapter 2
KADBASE Query Language

KQL (KADBASE Query Language) is the basis for the message frame structure described in Chapter 3. Its primary purpose is to define the syntax for the frame-based representation request. As a secondary function, it serves as the language for the KADBASE user interface.

KQL is primarily based on QUEL, the interactive, tuple-based, relational calculus query language for INGRES [Stonebraker 76, Woodfill 81]. Thus, KQL is a relational calculus language.[1] However, since it is intended to reference frame-based data structures with inheritance, it differs from QUEL in that it is *entity-based)* i.e., all the inherited attributes of an entity may be referenced as if they were directly available. KQL also contains an additional feature to facilitate queries with minimizing or maximizing criteria. KQL is strictly a data manipulation language; i.e., it does not contain capabilities to define or alter the organization of the KQL schemata, restrict access to data, define storage structures, or create indices. All KQL capabilities described in this chapter are implemented in the KADBASE user interface, the INGRES knowledge-based database interface, and the NDAM global subrequest knowledge module.

This chapter is divided into three sections to discuss the three KQL commands **(range, retrieve,** and **update),** the general definitions necessary to support the remainder of KQL, and a discussion of the use of optimizing criteria in KQL. The language is presented in a modified BNF using the following notational conventions:

Boldface **type** is used for reserved words in commands.
Angle brackets < > denote variables.
Square brackets [ ] indicate optional components.
Vertical bars | separate alternate definitions.

Wherever they occur, commas (,) and parentheses () are part of the language.

*In retrospect, some argument can be made for the use of relational algebra as the internal format for KADBASE requests. For example, algebra would facilitate recognition of join clauses for semantic translation, make subrequest decomposition easier, and simplify the implementation of the global subrequest processor. On the other hand, these objections do not outweigh the original advantages of a relational calculus formulation. The calculus focuses attention on *what* is requested rather than how to access it as in the procedural algebraic formulation.

2.1. KQL Commands

2.1.1. Range

Definition. KQL uses entity *range* statements that are identical in form to QUEL's relation range statements. An entity range statement declares an entity variable that references a specific entity type. Range statements are necessary to provide the capability to reference more than one instance of the same entity type in a single request. Entity variables retain their scope until redefined, except within aggregate expressions (see Section 2.2.3).

The form of the entity range statement is as follows:

<range> ::= **range of** <entity-variable> is <entity-name>

Examples. An example of the range statement is shown below.

range of beaml is beams

"Beaml" is the entity variable that references instances of the entity "beams". Further applications of this command are shown in the following sections.

2.1.2. Retrieve

Definition. The *retrieve* command is used to query KADBASE for sets of slot values. The set of slot names in the retrieve command is called the *target list.* The command may include boolean *qualifiers* that limit the sets of slot values returned and *optimizing criteria* to find the target-list that satisfies some minimizing or maximizing criteria (see Section 2.3). KADBASE responds to retrieve commands with a reply containing all satisficing sets of target list values.

The retrieve command is basically the same as its counterpart in QUEL with three exceptions:

- the target list must include the names of all slots that are to be returned—there is no option similar to the "all" in QUEL;
- the option to store the result of the query in a relation has been omitted; and
- the option to specify optimizing criteria has been added.

The retrieve command and its components are defined as follows:

```
<retrieve>  ::=  retrieve  [unique]  (<target-list>)
                          [where  <qualifier>]
                          [subject to   (<optimizing-criteria-list>)]

<target-list>  ::=  <target-item>  |
                    <target-list>, <target-item>

<target-item>  ::=  <slot-reference>  |

                    <slot-reference> = <expression>

<slot-reference>  ::=  <entity-variable>.<slot-name>

<optimizing-criterion-list>  ::=  <optimizing-criterion>  |
                                  <optimizing-criterion-list>, <optimizing-criterion>
<optimizing-criterion>  ::=  max (<arithmetic-expression>)   |
                             min  (<arithmetic-expression>)
```

Section 2.2 describes the specific syntax for expressions and qualifiers.

Examples. The first example is a simple query for a standard steel wide-flange section that has a depth less than or equal to 18 inches and a section modulus of at least 300 cubic inches.

```
range of  wf  is  wide-flange-sections
retrieve  (wf.designation, wf.depth, wf.section-modulus)
          where  wf.depth <= 18  and  wf.section-modulus >= 300
```

The second example shows an optimizing query to return the wide-flange section that has a section modulus greater than 300 inches and has the minimum weight among those sections having the smallest nominal depth (see Section 2.3 for a more detailed discussion of this query).

```
range of  wf  is  wide-flange-sections
retrieve  (wf.designation)
         where  wf.section-modulus >= 300
         subject to  (min(wf.nominal-depth), min(wf.weight))
```

The third example is a more complex query involving two entity variables. It represents a query for the set (presented pairwise) of wide-flange sections that have the same value for weight per linear foot.

```
range of  wfl   is  wide-flange-sections
range of  wf2   is  wide-flange-sections
retrieve  (wfl.designation, wf2.designation)
          where  wf1. weight = wf2.weight
```

2.1.3. Update

Definition. The *update* command is a combination of the append and replace commands in QUEL. The update command can be invoked in three different modes corresponding to the following three update options:

- *Replace* — update information for existing entities only (the same as QUEL's replace);
- *Append*—insert information for new entities only (similar to QUEL's append);
- *Combined* — update information for existing entities and insert information for new entities (a combination of the first two options).

The latter two options require the key attribute(s) for the entity must be specified in the target list. The third option (combined) is the default.

The form for the update command is as follows:

```
<update>  ::=  update  [<update-option>] (<target-list>)
                       [where <qualifier>]

<update-option>  ::=  replace | append | combined
```

The form for the update target list is the same as that for the retrieve target list, except that all slots specified in the target list must belong to the same entity (called the *update entity).*

Examples. The first example is a simple update where all the slot values are known. Note that this update will not create an entity if beam number 201 does not exist.

```
range of beam is beams
update replace (beam.length = 20, beam.designation = "W10xl2")
               where beam.number = 201
```

The second example is similar in appearance to the first, but it will change the database only if beam number 201 does not exist; in that case, it will add the data specified in the target list.

```
range of beam is beams
update append (beam.number = 201, beam.length = 20,
               beam.designation = MW10xl2")
```

The third example is a replacement with variables in the target-list. (This update will not create any entities.)

```
range of beam is beams
update replace (beam.length = 2 * beam.length)
               where beam.left-joint = 798
```

The fourth example is a combined update with variables in the target-list expressions. This request will create an adjacent joint entity (where the jointl and joint2 slots define the primary key) for each joint combination that does not already exist in the adjacent relation, and it will update the value of the distance slot for each joint combination that already exists.

```
range of jl is joints
range of j2 is joints
range of beam is beams
range of aj is adjacent-joints
update combined (aj.jointl =jl.number, aj.joint2=j2.number,
                 aj.distance = beam.length)
                where beam.left-joint = j 1 and beam.right-joint = j2
```

2.2. General Definitions

2.2.1. Qualifiers

The form for qualifiers in retrievals, inserts, and updates is taken directly from the *INGRES Reference Manual* [Woodfill 81]:

```
<qualifier> ::= <clause> |
                not <qualifier> |
                <qualifier> or <qualifier> |
                <qualifier> and <qualifier> |
                (<qualifier>)
```

2.2.2. Clauses

Clauses are comparisons of the values of expressions (arithmetic, string, or boolean) or boolean expressions (true, false, boolean attribute values, or boolean functions). Clauses may take the following forms:

```
<clause> ::= <arithmetic-expression> <comparison-operator> <arithmetic-expression> |
             <string-expression> <comparison-operator> <string-expression> |
             <boolean-expression> <comparison-operator> <boolean-expression> |
             <boolean-expression>
```

The first three forms may be abbreviated "<expression> <comparison-operator> <expression>". However, the longer form is used in the formal definition to emphasize that the expressions being compared must be compatible; e.g., strings can not be compared to numbers.

The KQL comparison operators are the same as those in QUEL, which in turn are based on those in the C programming language:

```
<comparison-operator> ::= <  |  <=  |  >  |  >=  I  ==  I  !=
```

2.2.3. Expressions

Expressions in clauses, updates, and insertions may be arithmetic, string, or boolean expressions.

```
<expression>  ::=  <arithmetic-expression>  |  <string-expression>  |
                   <boolean-expression>
```

Arithmetic Expressions. The KQL format for arithmetic expressions is based on the QUEL definition of *ajexpr* (arbitrary expression). Arithmetic expressions may be integer or real expressions; that distinction does not affect the following definition:

```
<arithmetic-expression>  ::=  <arithmetic-constant>  |
                              <arithmetic-slot-reference>  |
                              <arithmetic-function>  |
                              <arithmetic-aggregate>  |
                              (<arithmetic-expression>)  |
                              <unary-arithmetic-operator>  (<arithmetic-expression>)  |
                              <arithmetic-expression>   <binary-arithmetic-operator>
                                                        <arithmetic-expression>
```

An <arithmetic-slot-reference> is simply a <slot-reference> (see Section 2.1.2) in which the slot represents an arithmetic value in the local data structure. Type coercions are performed within arithmetic expressions as necessary.

Arithmetic Operators. The arithmetic operators are the same as in QUEL except that ** and A are both used to indicate exponentiation. The operators are shown in order of descending precedence:

```
<unary-arithmetic-operators>  ::=  +   |   -

<binary-arithmetic-operators>  ::=  **   |  A   |
                                    *    I  /   I
                                    +    |  -   |
```

Exponentiation is not valid for fractional powers of negative numbers.

Arithmetic Functions. The functions listed below are taken from those available under QUEL, supplemented by some useful C functions.

```
<arithmetic-function>  ::=  abs(<arithmetic-expression>)  |
                             atan(<arithmetic-expression>)  |
                             cos(<arithmetic-expression>)  |
                             exp(<arithmetic-expression>) |
                             log(<arithmetic-expression>)  |
                             mod(<arithmetic-expression>,  <arithmetic-expression>) |
                             sin(<arithmetic-expression>)  |
                             sqrt(<arithmetic-expression>)  |
                             float(<arithmetic-expression>)  |
                             fix(<arithmetic-expression>)
```

Arithmetic Aggregates. The KQL format for arithmetic aggregate expressions is identical to its QUEL counterpart:

```
<arithmetic-aggregate>  : :=  <arithmetic-aggregate-operator>(<arithmetic-expression>
                                                                [where <qualifier>])

<arithmetic-aggregate-operator>  ::=  count  |  sum  |  avg  |
                                      max  | min
```

As in QUEL, the scoping of entity variables in aggregate functions is independent of the scoping in the rest of the query.[2]

String and Character Expressions. String expressions are distinct from arithmetic expressions in KQL. Characters are strings of length one. String expressions have the following definition:

```
<string-expression>  ::=  tf<string-constant>tf  |
                          <string-slot-reference>  |
                          <string-function>  |

<string-constant>  ::= <character>  |
                       <string-constantxstring-constant>
```

A <string-slot-reference> is simply a <slot-reference> in which the slot represents a string value in the local data structure. The KQL definition for string constants is less robust than that in QUEL because KQL lacks wild-card matching.

Special string functions are defined to perform concatenation of strings, to extract substrings from strings, and to convert arithmetic expressions to strings.

[2] Originally, aggregates in KQL were intended to be *dependent* on the entity variable scoping in the rest of the query. However, this form proved to be difficult to translate into QUEL. Therefore, for programming expediency, the prototype version of KADBASE uses only independent aggregates.

```
<string-function>  ::=  concat(<string-expression> , <string-expression>) |
                         substring(<string-expression>,
                                   <integer-arithmetic-expression>,
                                   <integer-arithmetic-expression>)          |
                         ascii(<arithmetic-expression>)
```

Boolean Expressions. In KQL, boolean expressions have following definition:

```
<boolean-expression>  ::=  <boolean-slot-reference>  |
                           <boolean-aggregate>  |
                           true  |
                           false

<boolean-aggregate>  ::=  any(<slot-expression> [where <qualifier>])
```

A <boolean-slot-reference> is simply a <slot-reference> in which the slot represents a boolean value (true or false) in the local data structure. The boolean operator "any" is equivalent to a clause specifying the value returned by the "count" operator is greater than zero.

2.3. Optimizing Criteria in KQL

The use of a separate specification for optimizing criteria in queries arises from some basic observations about engineering queries, in particular single-object queries during the design process. The following sections describe the issues involved, present a KQL form including optimizing criteria, and discuss the implications of this new form.

2.3.1. Single-Object Queries: Examples and Discussion

Queries to a database for information about a single object take the general form: Find *<attributes of object>* where *<set of qualifiers>* are satisfied. As an example, consider a query to select a suitable beam cross-section from a database of standard steel wide-flange section. In relational query languages like QUEL [Stonebraker 76] and SQL [Date 81], the query would take the following forms:

QUEL

```
range of wf is wide-flange-sections
retrieve (wf.number)
        where wf.section-modulus > 300
        and wf.weight = MIN(wf.weight
                             where wf.section-modulus > 300)
```

SQL

```
SELECT   wf.designation
FROM     wide-flange-sections
WHERE    wf.section-modulus > 300
AND      wf.weight =
         (SELECT   MIN(wf.weight)
          FROM     wide-flange-sections
          WHERE    wf.section-modulus > 300)
```

Clearly, these queries contain two qualifiers: (1) the section modulus must be greater than the required section modulus, and (2) the weight must be the minimum of the set of sections that satisfy the first qualifier. The two qualifiers are excellent paradigms of two basic types of qualifiers:

- *Constraints* — those qualifiers that test the entity against an absolute standard; e.g., wf.section-modulus is greater than 300. Constraints determine a *satisficing* set of entities, i.e., those entities that satisfy the constraints without necessarily being optimum solutions.

- *Optimizing Criteria* — Those qualifiers that test the entities against a relative standard by comparing the entity to other entities of its class; e.g., wf.weight = min(wf.weight). Optimizing criteria are normally used to select a single object from the satisficing set determined by the constraints.

This division of qualifiers is not represented very clearly in either of the sample queries as both formulations require that the constraint be repeated in the aggregate operation. More complicated queries would further impair the distinction between these types of qualifiers in practical queries. In particular, the queries can become very complicated as the optimizing criteria are applied sequentially, i.e., the satisficing set is pared by the application of a first criteria, and the new smaller set is subjected to a second criteria.

2.3.2. KQL Form for Optimizing Criteria

KQL provides a query form that emphasizes the roles of the constraints and the hierarchy of optimizing criteria. The query consists of a specified entity type (or attributes of that entity), a set of constraints, and an ordered set of optimizing criteria. The constraints are applied first to determine the satisficing set, and the optimizing criteria are applied in order to narrow the set to the desired entity. As an example, consider the selection of a steel wide-flange section for a floor beam (taken from HI-RISE [Maher 85, Maher 84].) The constraint is the same as in the previous example—the beam section modulus must be greater than the require section modulus. The optimizing criteria are based on finding the minimum nominal depth and the minimum weight for the section with the minimum depth being the more important factor. The query formulations would be as follows:

QUEL

```
range of wf is wide-flange-sections
retrieve (wf.number)
        where wf.section-modulus > 300
        and wf.nominal-depth
            = MIN(wf.nominal-depth
                   where wf.section-modulus > 300)
        and wf.weight
            = MIN(wf. weight
                   where wf.nominal-depth = MIN(wf.nominal-depth
                                                 where wf.section-modulus
                                                        >= 300))
```

SQL

```
SELECT  wf.designation
FROM    wide-flange-sections
WHERE   wf.section-modulus > 300
AND     wf.nominal-depth =
        (SELECT  MIN(wf.nominal-depth)
         FROM    wide-flange-sections
         WHERE   wf.section-modulus > 300)
AND     wf. weight =
        (SELECT  MIN(wf.weight)
         FROM    wide-flange-sections
         WHERE   wf.section-modulus > 300
         AND     wf.nominal-depth =
                 (SELECT  MIN(wf.nominal-depth)
                  FROM    wide-flange-sections
                  WHERE   wf.section-modulus >= 300))
```

KQL

```
range of  wf  is  wide-flange-sections
retrieve (wf.designation)
        where  wf.section-modulus >= 300
        subject to  (min(wf.nominal-depth), min(wf.weight))
```

The intent of the query is much clearer in the KQL formulation than either the QUEL or SQL forms. The KQL form is also more amenable to query optimization because the structure of the query is not procedurally restrictive.

2.3.3. Implications For Constraint Processing

The engineering design process is an example of the classic search problem in artificial intelligence. The set of design constraints defines the solution space for the search. Within that solution space, the designer has a set of criteria that may be used to compare the relative merit of the individual solutions. The remaining component of the design process is a strategy for searching the solution space. When the designer is willing to limit the potential solution space to a finite domain (e.g., standard wide-flange steel sections), the solution space may be determined by testing each potential solution against the constraints. Likewise, the optimum solution may be selected by comparing each satisficing solution against the aggregate optimum value.

Database queries are an application of this technique of searching a limited space. The KQL query formulation emphasizes this view of queries by separating the constraints and optimizing criteria. When a designer is willing to accept the range of solutions represented by a library of solutions in the database, then he need only identify the applicable constraints and optimizing criteria, form them into a query, and send the query to the KADBASE. This simplification of the query process has some important implications for expert system queries to databases. With the general ability to query a database with an arbitrary list of constraints and optimizing criteria, the expert system rules can concentrate on the identification of the appropriate constraints and optimizing criteria without having to be concerned with details of query formulation and query handling.

Chapter 3
Message Data Structure

Internally, KADBASE components represent messages as trees of frames. The frame elements of specific messages are defined as *instances* of the template frames presented in this chapter, and therefore each is linked to the corresponding template frame through an *instance* relationship. The relationship *part-of-message* (with its inverse *message-subpart)* is used to link the message frame instances into a hierarchy. The root of the tree is an instance of the message frame. Its direct descendant is an instance of a KQL request frame **(retrieve** or **update)** or a KQL **reply** frame. Request target lists, qualifiers, and optimizing criteria are represented as subtrees composed of general purpose message **tree node** frames. Associated range commands are represented as instances of the **entity variable** frame. Each of these template frames is defined in detail in this chapter.

Updates and replies may have associated data files. For updates, the associated data file represents a virtual relation that the update target list and qualifier may reference as if it existed in the local data structure. For replies, the associated data file represents the reply to the corresponding retrieve request. In both cases, the data in the file is organized into tuples represented as lists of values.

This chapter has three sections: the first section describes the five special purpose frames (message, retrieve, update, reply, and entity variable), roughly paralleling the KQL chapter; the second section describes the general purpose message tree node; and the third section contains two complete sample messages from the SPEX demonstration.

In this chapter, frame names are typeset in **bold face,** and slot names are typeset in SMALL CAPITALS as shown below:

frame-name
SLOT1: slotl-value
SLOT2: slot2-value

3.1. Special Purpose Message Frames

3.1.1. Message Frame

The message frame acts as a header for all messages between KADBASE components. It contains only that information common to all types of messages. The frame name for a message instance consists of the three-letter component name of the sender concatenated with a local time stamp to produce a unique message identifier. The template for the message frame is shown below, followed by the slot definitions.

```
message
    TO:
    FROM:
    TIME:
    MESSAGE-TYPE:
    MESSAGE-SUBPART:
```

- TO — the KADBASE identifier of the system for which the message is intended; either an application, a DBMS, or the network manager (e.g., "nda" for the network data access manager).
- FROM — the KADBASE identifier of the system originating the message; either an application, a DBMS, or the network manager (e.g., "spx" for the SPEX application).
- TIME — the local time of the sender, expressed as a single value (e.g., the number of seconds since January 1, 1970: 519005610).
- MESSAGE-TYPE — the type of message tree attached to the MESSAGE-SUBPART; one of "retrieve", "update", or "reply".
- MESSAGE-SUBPART — the instance name of the message frame's immediate descendant in the message tree.

3.1.2. Retrieve Frame

The retrieve frame represents the retrieve command in KQL. It is linked to frames representing the qualifier, the optimizing criteria, the entity variables, the target list, and the retrieval meta-knowledge. Those frames are attached by the MESSAGE-SUBPART relationship, and indicated individually by specific slots in the retrieve frame. The template for the retrieve frame is shown below, followed by the slot definitions.

retrieve

```
    PART-OF-MESSAGE:
    ENTITY-VARIABLES:
    TARGET-LIST:
    QUALIFIER:
    CRITERIA:
    RETRIEVE-META-KNOWLEDGE:
    UNIQUE:
    MESSAGE-SUBPART:
```

- PART-OF-MESSAGE — the name of the message frame instance for this **retrieve.**
- ENTITY-VARIABLES — the list of the entity variable frames; this slot should never be empty. (This is really equivalent to including a list of the applicable range statements with the retrieve.)
- TARGET-LIST — the ordered list of the target items; this slot should never be empty. The sequence of the items indicates the order in which they are to appear in the reply.
- QUALIFIER — the root of the qualifier tree; if this slot is empty, then all instances of the target-list data are returned.
- CRITERIA — the ordered list of the optimizing criteria for the **retrieve;** if this slot is empty, the **retrieve** has no optimizing criteria. The order of the criteria determines their relative priority, with the first entry having the highest priority.
- RETRIEVE-META-KNOWLEDGE — the slot for attachment of retrieval meta-knowledge frames.[3]
- UNIQUE — boolean value indicating that the set of reply tuples should contain no duplicates.[4]
- MESSAGE-SUBPART — the instance names of the frames representing the various components of the retrieve, including the entity-variables, the target-list, the qualifier, the criteria, and the retrieve meta-knowledge.

3.1.3. Update Frame

The **update** frame represents the KQL update command. As with the **retrieve** frame, the **update** frame has descendants indicated by the MESSAGE-SUBPART relationship and by direct slot references.

As noted in the introduction to this chapter, updates may have an associated data file that acts

[3] **This implementation of KADBASE has no specific requirements for the use of meta-knowledge. However, to preserve compatibility with the original view of the system and to guard against the need for additional information about the retrieval, the retrieve frame contains an unused slot for meta-knowledge.**

[4] **Since the data structure for KADBASE is based on an entity-based, frame-based schema overlaying a relational system, *unique* should be the default. However, in practice, most relational database systems do not undertake to guarantee uniqueness of tuples unless specifically requested, and KADBASE adopts a similar approach.**

as a virtual relation. In that case, the entity-variable frame instance referenced in the DATA-FILE-ENTITY slot of the **update** frame is the entity variable that ranges over the tuples in that virtual relation. The subtrees for the update target list and qualifier may reference that entity variable in the same way as any other entity variable that is associated with an actual local data structure.

The template for the **update** frame is shown below, followed by the slot definitions.

```
update
    PART-OF-MESSAGE:
    UPDATE-ENTITY:
    ENTITY-VARIABLES:
    UPDATE-OPTION:
    TARGET-LIST:
    QUALIFIER:
    UPDATE-META-KNOWLEDGE:
    MESSAGE-SUBPART:
    NUMBER-OF-TUPLES:
    DATA-FILE-NAME:
    DATA-FILE-ENTITY:
```

- PART-OF-MESSAGE — the name of the message instance frame of which it is a part.
- UPDATE-ENTITY — the name of the entity variable frame which represents the entity that is being updated. This slot should never be empty.
- ENTITY-VARIABLES — the list of the entity variable frames; this slot should never be empty.
- UPDATE-OPTION — one of *replace, append,* or *combined', combined* is the default.
- TARGET-LIST — the list of the update target items; this slot should never be empty. All of the target list slots should belong to the update entity.
- QUALIFIER — the root of the qualifier tree; if this slot is empty, then the update has no qualifiers, and will update all applicable entities.
- UPDATE-META-KNOWLEDGE — the slot for the attachment of update meta-knowledge.[5]
- MESSAGE-SUBPART — the instance names of the frames representing the various components of the retrieve, including the entity-variables, the target list, the qualifier, and the update meta-knowledge.
- NUMBER-OF-TUPLES — the number of tuples in the associated data file, if any.
- DATA-FILE-NAME — the name of the associated data file containing the update tuples, if needed.
- DATA-FILE-ENTITY — the entity variable frame instance used to represent the tuples in the associated data file.

[5] *Sec* **footnote on page 15 about retrieve-meta-knowledge.**

3.1.4. Reply Frame

A **reply** frame must indicate the request message to which it is replying, the success of the request, a description of any errors in the event of failure, the number of tuples returned, and, of course, the requested data organized as tuples. If the reply is successful and the number of tuples is greater than zero, the tuples are contained in an associated data file whose name is stored in the DATA-FILE-NAME slot. The template for the **reply** frame is shown below, followed by the slot definitions.

```
reply
    PART-OF-MESSAGE:
    IN-REPLY-TO:
    SUCCESS:
    ERROR-CONDITION:
    NUMBER-OF-TUPLES:
    DATA-FILE-NAME:
```

- PART-OF-MESSAGE — the name of the message frame instance of which it is a part.
- IN-REPLY-TO — the name of the message frame instance that elicited the reply.
- SUCCESS — the status of the request represented as a boolean value; "true" indicates success of request.
- ERROR-CONDITION — the KADBASE-specific description of the error resulting from the request. For the prototype implementation, KADBASE traps only errors that arise from invalid commands to the INGRES DBMS; in the event of such an error, the INGRES error message is stored as a string in the ERROR-CONDITION slot.
- NUMBER-OF-TUPLES — the number of tuples in the associated data file.
- DATA-FILE-NAME — the name of the data file containing the reply tuples.

3.1.5. Entity Variable Frame

Entity variable frames represent the entity variables referenced in a request. In effect, they are implicit *range* statements attached to the request. They contain information on the entity type represented by the entity variable. Each entity reference within a retrieve or update tree is indicated by the name of the corresponding **entity-variable** frame. The template for the entity **variable** frame is shown below, followed by the slot definitions.

```
entity-variable
    PART-OF-MESSAGE:
    VARIABLE-NAME:
    ENTITY-TYPE:
```

- PART-OF-MESSAGE — the instance name of the retrieve, union, or update frame that includes the entity variable.

- VARIABLE-NAME — the string used to represent the entity variable (e.g., "wf' for the entity type "wide-flange-shapes").
- ENTITY-TYPE — the name of the entity in the terminology of the current schema (local schema, local frame-based schema, or global schema); the *Ifbs-entity* facet of the ENTITY-TYPE slot is used to link the **entity-variable** frame to the frame representing the entity in the local frame-based schema; likewise, the *gs-entity* facet is used to link the frame to the entity frame in the global schema.

3.2. General Purpose Message Tree Node Frames

Qualifiers, target lists, and criteria lists are represented by trees of **message tree node** frames. An instance of the **message tree node** frame can represent an operator, a literal value (arithmetic, string, or boolean), or a slot in the local data structure. Each type of **message tree node** frame is represented by a template frame. Message tree **nodes** are linked to a template frame via an *is-a* relationship. The three types of template frames are defined below.

- **Operator template frames** — The set of pre-defined operator template frames represents the operators defined in KQL (e.g., "+", "*", "min", etc.). The operators are organized into a shallow hierarchy of template frames, grouped according to similar properties (e.g., "binary arithmetic operators") as shown in Table 3-1. Each operator frame has (or can inherit) a TYPE slot and an ARGUMENT corresponding to the same slots in the **message tree node** frame. Binary operators have a special BINDING-POWER slot to indicate operator precedence for converting from the infix notation KQL to the prefix notation message data structure. The special symbol *value* used for the argument data types of the binary comparison operators ("==" and " !=") indicates that the two operands must be of the same data type.
- **Literal value template frames** — The set of literal value template frames represents the four types of literal values defined in KQL (i.e., "real", "integer", "string", and "boolean"). The value frames have a TYPE slot indicating the data types that value may assume, e.g., a literal integer value can be a "integer" or a "number". Table 3-2 shows the four types of literal values with the template frame names and the possible data type each can assume.
- **Slot reference template frame** — The slot reference template frame **(kad-slot-value)** is basically the same as the literal value template except that its **type** slot includes all possible data types ("boolean", "number", "integer", "real", "string"). KADBASE determines the data type according the usage of the slot reference.

Message tree node frames have the following uses in qualifiers, target lists, and optimizing criteria:

operator name	template frame name	binding power	type	argument
Assignment Operator				
Assignment	kad-op=	N/A	N/A	N/A
Unary arithmetic operators				
+	kad-unary-op+	N/A	number	number
—	kad-unary-op-	N/A	number	number
Binary arithmetic operators				
A	kad-op^{A}	5	number	number
*	kad-op*	4	number	number
/	kad-op/	4	number	number
+	kad-op+	3	number	number
—	kad-op-	3	number	number
Non-aggregate function operators				
absolute value	kad-op-abs	N/A	number	number
arctangent	kad-op-atan	N/A	real	number
cosine	kad-op-cos	N/A	real	number
exponentiation	kad-op-exp	N/A	real	number
logarithm	kad-op-log	N/A	real	number
modulo	kad-op-mod	N/A	integer	integer
sine	kad-op-sin	N/A	real	number
square root	kad-op-sqrt	N/A	real	number
number to ascii	kad-op-ascii	N/A	string	number
concatenate	kad-op-concat	N/A	string	string,string
substring	kad-op-substring	N/A	string	string,int,int
Aggregate function operators				
minimum	kad-op-min	N/A	number	slot,boolean
maximum	kad-op-max	N/A	number	slot,boolean
sum	kad-op-sum	N/A	number	slot,boolean
average	kad-op-avg	N/A	number	slot,boolean
count	kad-op-count	N/A	integer	slot,boolean
any	kad-op-any	N/A	boolean	slot,boolean
Binary comparison operators				
==	kad-op==	2	boolean	value,value
<	kad-op<	2	boolean	number,number
<=	kad-op<=	2	boolean	number,number
>	kad-op>	2	boolean	number,number
>=	kad-op>=	2	boolean	number,number
!=	kad-op!=	2	boolean	value,value
Unary logical operator				
not	kad-op-not	N/A	boolean	boolean
Binary logical operators				
and	kad-op-and	1	boolean	boolean,boolean
or	kad-op-or	0	boolean	boolean,boolean

Table 3-1: KADBASE Operator Template Frame Definitions

value type	template frame name	TYPE
real	**kad-real-value**	number, real
integer	**kad-integer-value**	number,integer
string	**kad-string-value**	string
boolean	**kad-boolean-value**	boolean

Table 3-2: KADBASE Literal Value Template Frame Definitions

- A *qualifier* consists of a hierarchy of **message tree node** instances representing a valid sequence of operators and values (expressed in prefix notation), as defined in the KQL qualifier syntax. For instance, the KQL qualifier "10 == wf.area " is represented as a tree of three **message tree node** instances, with the frame for the equality comparison operator being the root of the tree as shown below:

  nodel
  IS-A: kad-op==
  MESSAGE-SUBPART: node2 node3

  node2
  IS-A: kad-integer-value
  VALUE: 10
  PART-OF-MESSAGE: nodel

  node3
  IS-A: kad-slot-value
  ENTITY-REFERENCE: wf-entity-variable
  SLOT-REFERENCE: area
  PART-OF-MESSAGE: nodel

  where **wf-entity-variable** is entity variable frame linking the variable "wf to the entity type "wide-flange-shapes".

- A *target list* is an ordered set of **message tree node** instances. Each instance in the target list must be either a reference to a slot in the local data structure or an assignment operator ("=") with a slot reference as its first descendant and a hierarchy of **message tree node** instances representing an expression as its second descendant (see the definition of target list in Section 2.1.2)

- An *optimizing criteria list* is an ordered set of message **tree node** instances. Each frame instance in the list must be either a "min" or "max" operator. The order of the instances indicates the priority of the optimizing criteria.

The general template for the **message tree node** frame is shown below, followed by the slot definitions.

```
message tree node
    PART-OF-MESSAGE:
    MESSAGE-SUBPART:
    IS-A:
    TYPE:
    ARGUMENTS:
    VALUE:
    ENTITY-REFERENCE:
    SLOT-REFERENCE:
```

- PART-OF-MESSAGE — the name of the frame's parent node in tree.
- MESSAGE-SUBPART — the ordered list of descendants; only filled when the message tree **node** instance is an operator, in which case the descendants are the operands. The ordering of the descendants is dependent on the operator.
- IS-A — the name of the operator, literal value, or slot reference template frame to which the **message tree node** corresponds.
- TYPE — the list of the possible data types of the value or operator represented by the message tree **node;** one of number, integer, real, string, or boolean; may be inherited through the IS-A slot.
- ARGUMENTS — an ordered list of the data types of the arguments (operands) of the message **tree node** if it is an operator; may be inherited through the IS-A slot.
- VALUE — the numeric, string, or boolean value.
- ENTITY-REFERENCE — the instance name of the entity-variable frame.
- SLOT-REFERENCE — the name of a slot in the local schema; the *Ifbs-slot* facet of the SLOT-REFERENCE slot is used to link the message **tree node** to the frame representing the slot in the local frame-based schema; likewise, the *gs-slot* facet is used to link the frame to the slot frame in the global schema.

3.3. Sample Message Frames

This section contains two sample messages taken from the SPEX demonstration described in the thesis. The first message represents the SPEX query as it is sent from the SPEX KBSI to the AISC KBDBI. The second message represents the reply from the AISC KBDBI.

3.3.1. Sample SPEX Query

The query is shown graphically in Figure 3-1 (Figure 5-4 in the thesis). The frame representation of the query is shown below and on the following pages.

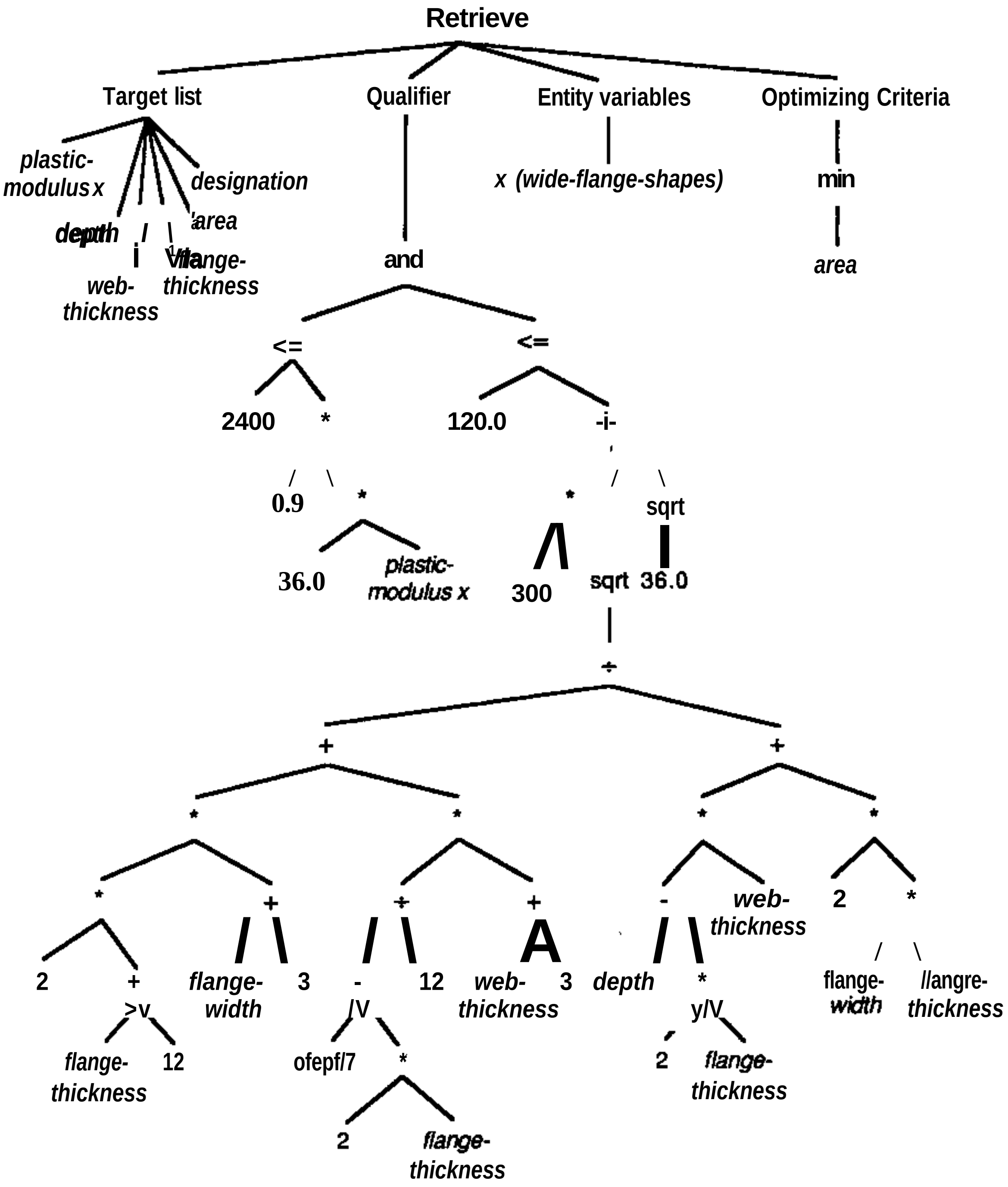

Figure 3-1: Graphical Representation of SPEX Query

```
spx519005610
     MESSAGE-SUBPART:  spx-retrieve-g00013
     MESSAGE-TYPE:  retrieve
     TIME:  519005610
     FROM:  spx
     TO:  ais
     INSTANCE:  message

spx-retrieve-g00013
     PART-OF-MESSAGE:  spx519005610
     CRITERIA:  spx-msg-tree-g00073
     QUALIFIER:  spx-msg-tree-g00022
     TARGET-LIST:  spx-msg-tree-g00021 spx-msg-tree-g00020
                   spx-msg-tree-g00019spx-msg-tree-g00018
                   spx-msg-tree-g00017  spx-msg-tree-g00016
                   spx-msg-tree-g00015
     ENTITY-VARIABLES:  spx-entity-g00014
     MESSAGE-SUBPART:  spx-entity-g00014 spx-msg-tree-g00021
                   spx-msg-tree-g00020spx-msg-tree-g00019
                   spx-msg-tree-g00018spx-msg-tree-g00017
                   spx-msg-tree-g00016spx-msg-tree-g00015
                   spx-msg-tree-g00022  spx-msg-tree-g00073
     INSTANCE:  retrieve

spx-msg-tree-g00073
     PART-OF-MESSAGE:  spx-retrieve-g00013
     MESSAGE-SUBPART:  spx-msg-tree-g00074
     IS-A:  kad-op-min
     INSTANCE:  message-tree-node

spx-msg-tree-g00074
     TYPE:  slot
     PART-OF-MESSAGE:  spx-msg-tree-g00073
     ENTITY-REFERENCE:  spx-entity-g00014
     SLOT-REFERENCE:  area
     ISA:  kad-slot-value
     INSTANCE:  message-tree-node

spx-msg-tree-g00022
     PART-OF-MESSAGE:  spx-retrieve-g00013
     MESSAGE-SUBPART:  spx-msg-tree-g00030 spx-msg-tree-g00023
     ISA:  kad-op-and
     INSTANCE:  message-tree-node

spx-msg-tree-g00023
     TYPE:  boolean
     PART-OF-MESSAGE:  spx-msg-tree-g00022
     MESSAGE-SUBPART:  spx-msg-tree-g00025 spx-msg-tree-g00024
     IS-A:  kad-op<=
     INSTANCE:  message-tree-node

spx-msg-tree-g00024
     TYPE:  number
     PART-OF-MESSAGE:  spx-msg-tree-g00023
     VALUE:  2400.0
     ISA:  kad-real-value
     INSTANCE:  message-tree-node

spx-msg-tree-g00025
     TYPE:  number
     PART-OF-MESSAGE:  spx-msg-tree-g00023
     MESSAGE-SUBPART:  spx-msg-tree-g00027 spx-msg-tree-g00026
     IS-A:  kad-op*
     INSTANCE:  message-tree-node
```

```
spx-msg-tree-g00026
    TYPE: number
    PART-OF-MESSAGE: spx-msg-tree-g00025
    VALUE: 0.9
    IS-A: kad-real-value
    INSTANCE: message-tree-node

spx-msg-tree-g00027
    TYPE: number
    PART-OF-MESSAGE: spx-msg-tree-g00025
    MESSAGE-SUBPART: spx-msg-tree-g00029 spx-msg-tree-g00028
    ISA: kad-op*
    INSTANCE: message-tree-node

spx-msg-tree-g00028
    TYPE: number
    PART-OF-MESSAGE: spx-msg-tree-g00027
    VALUE: 36.0
    ISA: kad-real-value
    INSTANCE: message-tree-node

spx-msg-tree-g00029
    TYPE: number
    PART-OF-MESSAGE: spx-msg-tree-g00027
    ENTITY-REFERENCE: spx-entity-g00014
    SLOT-REFERENCE: plastic-modulus-x
    IS-A: kad-slot-value
    INSTANCE: message-tree-node

spx-msg-tree-g00030
    TYPE: boolean
    PART-OF-MESSAGE: spx-msg-tree-g00022
    MESSAGE-SUBPART: spx-msg-tree-g00032 spx-msg-tree-g00031
    ISA: kad-op<=
    INSTANCE: message-tree-node

spx-msg-tree-g00031
    TYPE: number
    PART-OF-MESSAGE: spx-msg-tree-g00030
    VALUE: 120.0
    ISA: kad-real-value
    INSTANCE: message-tree-node

spx-msg-tree-g00032
    TYPE: number
    PART-OF-MESSAGE: spx-msg-tree-g00030
    MESSAGE-SUBPART: spx-msg-tree-g00071 spx-msg-tree-g00033
    ISA: kad-op/
    INSTANCE: message-tree-node

spx-msg-tree-g00033
    TYPE: number
    PART-OF-MESSAGE: spx-msg-tree-g00032
    MESSAGE-SUBPART: spx-msg-tree-g00035 spx-msg-tree-g00034
    ISA: kad-op*
    INSTANCE: message-tree-node

spx-msg-tree-g00034
    TYPE: number
    PART-OF-MESSAGE: spx-msg-tree-g00033
    VALUE: 300
    IS-A: kad-integer-value
    INSTANCE: message-tree-node
```

```
spx-msg-tree-g00035
    TYPE: number
    PART-OF-MESSAGE: spx-msg-tree-g00033
    MESSAGE-SUBPART: spx-msg-tree-g00036
    IS-A: kad-op-sqrt
    INSTANCE: message-tree-node

spx-msg-tree-g00036
    TYPE: number
    PART-OF-MESSAGE: spx-msg-tree-g00035
    MESSAGE-SUBPART: spx-msg-tree-g00058 spx-msg-tree-g00037
    ISA: kad-op/
    INSTANCE: message-tree-node

spx-msg-tree-g00037
    TYPE: number
    PART-OF-MESSAGE: spx-msg-tree-g00036
    MESSAGE-SUBPART: spx-msg-tree-g00047 spx-msg-tree-g00038
    ISA: kad-op+
    INSTANCE: message-tree-node

spx-msg-tree-g00038
    TYPE: number
    PART-OF-MESSAGE: spx-msg-tree-g00037
    MESSAGE-SUBPART: spx-msg-tree-g00044 spx-msg-tree-g00039
    ISA: kad-op*
    INSTANCE: message-tree-node

spx-msg-tree-g00039
    TYPE: number
    PART-OF-MESSAGE: spx-msg-tree-g00038
    MESSAGE-SUBPART: spx-msg-tree-g00041 spx-msg-tree-g00040
    ISA: kad-op*
    INSTANCE: message-tree-node

spx-msg-tree-g00040
    TYPE: number
    PART-OF-MESSAGE: spx-msg-tree-g00039
    VALUE: 2
    IS-A: kad-integer-value
    INSTANCE: message-tree-node

spx-msg-tree-g00041
    TYPE: number
    PART-OF-MESSAGE: spx-msg-tree-g00039
    MESSAGE-SUBPART: spx-msg-tree-g00043 spx-msg-tree-g00042
    ISA: kad-op/
    INSTANCE: message-tree-node

spx-msg-tree-g00042
    TYPE: number
    PART-OF-MESSAGE: spx-msg-tree-g00041
    ENTITY-REFERENCE: spx-entity-g00014
    SLOT-REFERENCE: flange-thickness
    IS-A: kad-slot-value
    INSTANCE: message-tree-node

spx-msg-tree-g00043
    TYPE: number
    PART-OF-MESSAGE: spx-msg-tree-g00041
    VALUE: 12
    IS-A: kad-integer-value
    INSTANCE: message-tree-node
```

spx-msg-tree-g00044
TYPE: number
PART-OF-MESSAGE: spx-msg-tree-g00038
MESSAGE-SUBPART: spx-msg-tree-g00046 spx-msg-tree-g00045
ISA: kad-op^A
INSTANCE: message-tree-node

spx-msg-tree-g00045
TYPE: number
PART-OF-MESSAGE: spx-msg-tree-g00044
ENTITY-REFERENCE: spx-entity-g00014
SLOT-REFERENCE: flange-width
IS-A: kad-slot-value
INSTANCE: message-tree-node

spx-msg-tree-g00046
TYPE: number
PART-OF-MESSAGE: spx-msg-tree-g00044
VALUE: 3
ISA: kad-integer-value
INSTANCE: message-tree-node

spx-msg-tree-g00047
TYPE: number
PART-OF-MESSAGE: spx-msg-tree-g00037
MESSAGE-SUBPART: spx-msg-tree-g00055 spx-msg-tree-g00048
ISA: kad-op*
INSTANCE: message-tree-node

spx-msg-tree-g00048
TYPE: number
PART-OF-MESSAGE: spx-msg-tree-g00047
MESSAGE-SUBPART: spx-msg-tree-g00054 spx-msg-tree-g00049
IS-A: kad-op/
INSTANCE: message-tree-node

spx-msg-tree-g00049
TYPE: number
PART-OF-MESSAGE: spx-msg-tree-g00048
MESSAGE-SUBPART: spx-msg-tree-g00051 spx-msg-tree-g00050
IS-A: kad-op-
INSTANCE: message-tree-node

spx-msg-tree-g00050
TYPE: number
PART-OF-MESSAGE: spx-msg-tree-g00049
ENTITY-REFERENCE: spx-entity-g00014
SLOT-REFERENCE: depth
ISA: kad-slot-value
INSTANCE: message-tree-node

spx-msg-tree-g00051
TYPE: number
PART-OF-MESSAGE: spx-msg-tree-g00049
MESSAGE-SUBPART: spx-msg-tree-g00053 spx-msg-tree-g00052
ISA: kad-op*
INSTANCE: message-tree-node

spx-msg-tree-g00052
TYPE: number
PART-OF-MESSAGE: spx-msg-tree-g00051
VALUE: 2
ISA: kad-integer-value
INSTANCE: message-tree-node

spx-msg-tree-g00053
TYPE: number
PART-OF-MESSAGE: spx-msg-tree-g00051
ENTITY-REFERENCE: spx-entity-g00014
SLOT-REFERENCE: flange-thickness
IS-A: kad-slot-value
INSTANCE: message-tree-node

spx-msg-tree-g00054
TYPE: number
PART-OF-MESSAGE: spx-msg-tree-g00048
VALUE: 12
IS-A: kad-integer-value
INSTANCE: message-tree-node

spx-msg-tree-g00055
TYPE: number
PART-OF-MESSAGE: spx-msg-tree-g00047
MESSAGE-SUBPART: spx-msg-tree-g00057 spx-msg-tree-g00056
IS-A: kad-op^{A}
INSTANCE: message-tree-node

spx-msg-tree-g00056
TYPE: number
PART-OF-MESSAGE: spx-msg-tree-g00055
ENTITY-REFERENCE: spx-entity-g00014
SLOT-REFERENCE: web-thickness
IS-A: kad-slot-value
INSTANCE: message-tree-node

spx-msg-tree-g00057
TYPE: number
PART-OF-MESSAGE: spx-msg-tree-g00055
VALUE: 3
IS-A: kad-integer-value
INSTANCE: message-tree-node

spx-msg-tree-g00058
TYPE: number
PART-OF-MESSAGE: spx-msg-tree-g00036
MESSAGE-SUBPART: spx-msg-tree-g00066 spx-msg-tree-g00059
IS-A: kad-op+
INSTANCE: message-tree-node

spx-msg-tree-g00059
TYPE: number
PART-OF-MESSAGE: spx-msg-tree-g00058
MESSAGE-SUBPART: spx-msg-tree-g00065 spx-msg-tree-g00060
ISA: kad-op*
INSTANCE: message-tree-node

spx-msg-tree-g00060
TYPE: number
PART-OF-MESSAGE: spx-msg-tree-g00059
MESSAGE-SUBPART: spx-msg-tree-g00062 spx-msg-tree-g00061
ISA: kad-op-
INSTANCE: message-tree-node

spx-msg-tree-g00061
TYPE: number
PART-OF-MESSAGE: spx-msg-tree-g00060
ENTITY-REFERENCE: spx-entity-g00014
SLOT-REFERENCE: depth
ISA: kad-slot-value
INSTANCE: message-tree-node

spx-msg-tree-g00062
TYPE: number
PART-OF-MESSAGE: spx-msg-tree-g00060
MESSAGE-SUBPART: spx-msg-tree-g00064 spx-msg-tree-g00063
IS-A: kad-op*
INSTANCE: message-tree-node

spx-msg-tree-g00063
TYPE: number
PART-OF-MESSAGE: spx-msg-tree-g00062
VALUE: 2
IS-A: kad-integer-value
INSTANCE: message-tree-node

spx-msg-tree-g00064
TYPE: number
PART-OF-MESSAGE: spx-msg-tree-g00062
ENTITY-REFERENCE: spx-entity-g00014
SLOT-REFERENCE: flange-thickness
IS-A: kad-slot-value
INSTANCE: message-tree-node

spx-msg-tree-g00065
TYPE: number
PART-OF-MESSAGE: spx-msg-tree-g00059
ENTITY-REFERENCE: spx-entity-g00014
SLOT-REFERENCE: web-thickness
ISA: kad-slot-value
INSTANCE: message-tree-node

spx-msg-tree-g00066
TYPE: number
PART-OF-MESSAGE: spx-msg-tree-g00058
MESSAGE-SUBPART: spx-msg-tree-g00068 spx-msg-tree-g00067
ISA: kad-op*
INSTANCE: message-tree-node

spx-msg-tree-g00067
TYPE: number
PART-OF-MESSAGE: spx-msg-tree-g00066
VALUE: 2
IS-A: kad-integer-value
INSTANCE: message-tree-node

spx-msg-tree-g00068
TYPE: number
PART-OF-MESSAGE: spx-msg-tree-g00066
MESSAGE-SUBPART: spx-msg-tree-g00070 spx-msg-tree-g00069
IS-A: kad-op*
INSTANCE: message-tree-node

spx-msg-tree-g00069
TYPE: number
PART-OF-MESSAGE: spx-msg-tree-g00068
ENTITY-REFERENCE: spx-entity-g00014
SLOT-REFERENCE: flange-width
ISA: kad-slot-value
INSTANCE: message-tree-node

```
spx-msg-tree-g00070
    TYPE: number
    PART-OF-MESSAGE: spx-msg-tree-g00068
    ENTITY-REFERENCE: spx-entity-g00014
    SLOT-REFERENCE: flange-thickness
    IS-A: kad-slot-value
    INSTANCE: message-tree-node

spx-msg-tree-g00071
    TYPE: number
    PART-OF-MESSAGE: spx-msg-tree-g00032
    MESSAGE-SUBPART: spx-msg-tree-g00072
    IS-A: kad-op-sqrt
    INSTANCE: message-tree-node

spx-msg-tree-g00072
    TYPE: number
    PART-OF-MESSAGE: spx-msg-tree-g00071
    VALUE: 36.0
    IS-A: kad-real-value
    INSTANCE: message-tree-node

spx-msg-tree-g00015
    PART-OF-MESSAGE: spx-retrieve-g00013
    ENTITY-REFERENCE: spx-entity-g00014
    SLOT-REFERENCE: plastic-modulus-x
    IS-A: kad-slot-value
    INSTANCE: message-tree-node

spx-msg-tree-gOOO 16
    PART-OF-MESSAGE: spx-retrieve-g00013
    ENTITY-REFERENCE: spx-entity-g00014
    SLOT-REFERENCE: depth
    IS-A: kad-slot-value
    INSTANCE: message-tree-node

spx-msg-tree-g00017
    PART-OF-MESSAGE: spx-retrieve-g00013
    ENTITY-REFERENCE: spx-entity-g00014
    SLOT-REFERENCE: web-thickness
    IS-A: kad-slot-value
    INSTANCE: message-tree-node

spx-msg-tree-g00018
    PART-OF-MESSAGE: spx-retrieve-g00013
    ENTITY-REFERENCE: spx-entity-g00014
    SLOT-REFERENCE: flange-width
    IS-A: kad-slot-value
    INSTANCE: message-tree-node

spx-msg-tree-g00019
    PART-OF-MESSAGE: spx-retrieve-g00013
    ENTITY-REFERENCE: spx-entity-g00014
    SLOT-REFERENCE: flange-thickness
    IS-A: kad-slot-value
    INSTANCE: message-tree-node

spx-msg-tree-g00020
    PART-OF-MESSAGE: spx-retrleve-g00013
    ENTITY-REFERENCE: spx-entity-g00014
    SLOT-REFERENCE: area
    IS-A: kad-slot-value
    INSTANCE: message-tree-node
```

spx-msg-tree-g00021
PART-OF-MESSAGE: **spx-retrieve-g00013**
ENTITY-REFERENCE: **spx-entity-g00014**
SLOT-REFERENCE: designation
IS-A: kad-slot-value
INSTANCE: message-tree-node

spx-entity-g00014
PART-OF-MESSAGE: spx-retrieve-g00013
ENTITY-TYPE: wide-flange-shape
VARIABLE-NAME: X
INSTANCE: **entity-**variable

3.3.2. Sample SPEX Reply

The frame representation of the reply from the AISC KBDBI to the SPEX query is shown below. It is followed by a listing of the contents of the reply data file.

a is519007098
MESSAGE-SUBPART: ais-reply-g00005
MESSAGE-TYPE: reply
TIME: 519007098
FROM: ais
TO: spx
INSTANCE: message

ais-reply-g00005
NUMBER-OF-TUPLES: 1
DATA-FILE-NAME: ais519007104
PART-OF-MESSAGE: ais519007098
SUCCESS: t
IN-REPLY-TO: spx519005610
INSTANCE: reply

The corresponding reply data file contains the following line.

```
(77.9 12.06 0.345 9.995 0.575 15.6 "W12x53")
```

Chapter 4
KADBASE Schema Description Details

This chapter presents a detailed definition of each frame-based schema description component. The schema definitions are represented using instances of the template frames described in the following sections. The frames representing a schema are organized into a hierarchy using the *part-of-schema* relationship and its inverse relationship *schema-subpart.* This chapter also describes the KADBASE schema definition utility, which uses a problem-oriented language to build the schema description frames for each component. The material in this chapter is intended to supplement the information found in Section 4.1 of the thesis.

The local schema examples are based on the concrete footing entity definition for the HICOST-II building design database, described in Section 5.3.2.3 and the accompanying Figure 5.4 of the thesis. The global schema examples are based on the HICOST-II construction quantity and unit cost entity definitions. The following six sections describe the local frame-based schema (LFBS), the local frame-based mapping (LFBM), the local integration mapping (LIM), the global schema (GS), the global data source mapping (GDSM), and the global integration mapping (GIM). The seventh section describes the problem-oriented language for the schema definition utility.

4.1. Local Frame-Based Schema Definition

The local frame-based schema (LFBS) of a database or KBS context consists of the definitions of the entities and slots included in the local data structure. An entity may be represented in the underlying data structure by a single relation, a frame template, or a network node (depending on the underlying local data representation), or it may be a composite of several relations, frames, or nodes. For instance, all relations having the same primary key describe the same entity. The entities and slots in the local frame-based schema are defined as frames and linked into a hierarchy via the *part-of-schema* relationship. The LFBS frame is the top node of the hierarchy, and the entity defintion frames are its direct descendants. The following sections describe the information required for entity slot definitions.

4.1.1. LFBS Entity Definition

The following information is required to define an entity type in the LFBS. These items correspond to the slots in the frame that represents the entity.

- ENTITY NAME — the local schema name for the entity.[6] The entity name must be a *label*
- PRIMARY KEY — a single attribute or a list of attributes and relationships that serves to uniquely identify the entity (a designation, an object number, a frame instance name, etc.). Wherever possible, entities should have a single-attribute primary key.[7]
- CANDIDATE-KEYS — the single attributes or lists of attributes and relationships, each of which uniquely identifies an instance of the entity. The primary key is by definition a candidate key and therefore must be a value for this slot.
- PART-OF-SCHEMA — the link to the LFBS frame.
- SCHEMA-SUBPART — the slot frames that are associated with this entity.

In the building design database, the LFBS definition for the concrete footing entity is as follows:

```
concrete-footing
        ENTITY-NAME:  concrete-footing
        PRIMARY-KEY:  ID
        CANDIDATE-KEYS:  ID
        PART-OF-SCHEMA:  lfbs
```

4.1.2. LFBS Slot Definition

Slots represent attributes or relationships with other entities. The following general information is required to describe a slot irrespective of its type:

- SLOT-NAME — the name of the slot in the local data structure. The slot name must be a *label.*
- SLOT-TYPE — the role of the slot in the local data structure; either *attribute* or *relationship.*

[6] **If the entity corresponds to a several local data structures, then it may not be feasible or desirable to have the entity name correspond to a particular name in the local schema.**

[7] **As an implementation restriction, the current of KADBASE cannot perform some mappings for entities that do not have single-attribute primary keys. Translations between the LS and LFBS for entities that correspond to multiple local data structures are implemented only for the case where the entity has a single-attribute primary key that is present is all corresponding local data structures. Also, any entity which acts as the range in a relationship must have a single attribute primary key.**

A relationship can not be a primary key by itself because that would imply a one-to-one relationship which would be better expressed by grouping the attributes and relationships of the object into a single entity definition.

- LOCAL-CONSTRAINTS — a list of local constraints.[8] These constraints may apply to either attributes (e.g., slot value is less than 10) and relationships (e.g., the slot value must be a value for slot X in relation Y).

- DATA-TYPE — the local storage type; may be one of *integer, decimal, real, number* (may be either real or integer), *character* (treated as strings of length 1), *boolean,* or *string* (string may include an optional length specification, e.g., string[10]). A relationship slot is restricted to integers, characters, or strings. A special *list* type may be used in frame-based systems for inverse relationship slots, but usually it is restricted to local use and cannot be mapped across schemata.[9]

- PART-OF-SCHEMA — the link to the entity frame for the slot.

The following special information is required for slots that define attributes.

- UNITS — feet, inches, lbs/If, dollars, etc. Only those units that have been defined in the global set of unit descriptions are allowed.[10]

The following special information is required for slots that define relationships.

- RELATIONSHIP-TYPE — the name of the underlying relationship type (or its inverse); e.g., IS-A, PART-OF, INSTANCE, and their inverses.

- RANGE — the entity types that may be referenced by the relationship.

The following definitions for the ID, J2, and SS9 slots of concrete footing are examples of the LFBS slot definition frame.

ID

SLOT-NAME: ID
SLOT-TYPE: attribute
DATA-TYPE: integer
PART-OF-SCHEMA: **concrete-footing**

J2

SLOT-NAME: J2
SLOT-TYPE: attribute
DATA-TYPE: real
UNITS: inches
PART-OF-SCHEMA: **concrete-footing**

[8]Local constraints are not implemented in the current version of KADBASE.

[9]Additional special data types (e.g., a bitmap type to represent images) may be required.

[1]^he unit description may be regarded as an implicit domain mapping, but it is included here because units serve to characterize the quantities represented in addition to providing a means of mapping between those dimensioned quantities. In this implementation of KADBASE, unit descriptions are treated as literals; i.e., an arithmetic unit definition such as "ft * lb / s" are not treated as combination of feet (ft), pounds (lb), and seconds (s), but as a single literal unit. An excellent discussion of the proper way to define units may be found in [Manner 86]. The allowable list of units is stored in a global variable ^{M}kad-units.^{M}

SS9
  SLOT-NAME: SS9
  SLOT-TYPE: attribute
  DATA-TYPE: integer
  PART-OF-SCHEMA: concrete-footing

4.2. Local Frame-Based Mapping

The local frame-based mapping (LFBM) relates the entities and slots in the local frame-based schema to the underlying local schema. The need for this mapping information arises from the fact that entities in the LFBS may be represented by multiple relations in the local schema. If two relations share a common primary key, then both represent the same entity type, and the slots from both relations apply to that entity type.

The local frame-based mapping is required for two phases of the semantic mapping: from LFBS to local schema (LS) and from LS to LFBS. Accordingly, the mapping information needs to be readily accessible for either type of translation; i.e., given an entity or slot in the LFBS, identify the corresponding relation(s) or slot(s) in the LS, and vice versa. For the LFBS to LS translation, the information can be attached to the entity and slot definitions in the LFBS schema. For the LS to LFBS translation, a separate index is required to map the LS data structures into the LFBS entities and slots. Since both sets of mapping information are equivalent, one set can be generated from the other. Therefore, only one set needs to be defined.[11]

To provide the LFBS to LS translation, the following slot is added to each entity definition frame:

- LFBM — the names of the local data structures (relations, frames, objects, nodes, etc.) that define the entity in the local data representation.

Likewise, the following slot is added to each slot definition frame:

- LFBM — the local occurrences of the slot. Each local occurrence is represented as a pair consisting of the local data structure name and the local slot name.

The LS to LFBS translation is represented by an index that maps the LS data structures into the LFBS. The index is implemented as a frame. Each slot of the index frame corresponds to the name of an LS data structure (relation, frame, etc.) The value of the slot is the name of the corresponding entity in the LFBS. The other facets of the slot correspond to each attribute of the LS data structure.

[n] **The KADBASE schema definition utility uses the LFBS to LS mapping information to define the LFBM.**

To represent the LFBM for the building design database, the following slots are attached to the entity and slot frames previously defined.

```
concrete-footing
        LFBM:  cf__att  cf_spec


ID
        LFBM:  (cf_att.ID)  (cf_spec . ID)

J2
        LFBM:  (cf_att. J)


SS9
        LFBM:  (cf_spec. SS9)
```

4.3. Local Integration Mapping

The local integration mapping is essentially a terminology and domain mapping between the local frame-based schema and the global schema. The global names for the entities and slots must be indicated if the local names are different, and the local attribute domains must be mapped into the corresponding global domains if they differ.

As with the local frame-based mapping, the local integration mapping is required for two phases of the semantic mapping: from LFBS to GS and from GS to LFBS. Accordingly, the mapping information needs to be readily accessible for either type of translation; i.e., given an entity or slot in the LFBS, identify the corresponding entity or slot in the GS, and vice versa. For the LFBS to GS translation, the information can be attached to the entity and slot definitions in the LFBS schema. For the GS to LFBS translation, a separate index is required to map the GS data structures into the LFBS entities and slots. Since both sets of mapping information are equivalent, one set can be generated from the other. Therefore, only one set needs to be defined.[12]

To provide the LFBS to GS entity translation, the following slot is added to each entity definition frame:

- LIM — the name of the corresponding global entity, if different from the local name.

[12] The KADBASE schema definition utility uses the LFBS to *GS* mapping information to define the LIM.

The LFBS to GS translation for slots includes both name mappings for all slots and domain mappings for slots that act as attributes. These domain mappings include *tabular domain mappings,* which are used to represent one-to-one mappings between local attribute values and global values, and *functional domain mappings,* which are used when tabular mappings are inappropriate. Domain mapping functions need not be reversible; i.e., either a direct or an inverse mapping function may appear independently. The following LIM slots are attached to the LFBS slot frame.

- LIM — the global name for the slot, if different from the local name.
- DIRECT-MAPPING-TABLE — a table that represents a one-to-one mapping from the local domain values to the global domain values. Only one of the two mapping tables needs to be defined explicitly as the other may be generated directly. In the KADBASE prototype, a mapping table is represented by Lisp "assoc" list of dotted pairs corresponding to each pair of values in die one-to-one mapping.
- INVERSE-MAPPING-TABLE — a table that represents a one-to-one mapping from the global domain values to the local domain values.
- DIRECT-MAPPING-FUNCTION — a function that takes a local value of the attribute and produces an equivalent value in the corresponding domain of the global system. For KADBASE, the mapping functions are defined in Lisp.
- INVERSE-MAPPING-FUNCTION — a function that takes a global value of the attribute and produces an equivalent value in the corresponding domain of the local system.
- MAPPED-TYPE — the data type of the attribute value after direct tabular or functional mapping, i.e., the *global type.*

The GS to LFBS translation in the LIM is represented by an index which maps the GS data structures into the LFBS in the same fashion as the index in the LFBM maps the LS into the LFBS. The index is implemented as a frame. Each slot of the index frame corresponds to the name of a GS entity. The value of the slot is the name of the corresponding entity in the LFBS. The other facets of the slot correspond to each GS slot for that entity that is present in the LFBS.

The following shows the LIM additions to the previously-defined sample slot definitions for the concrete footing entity.

```
ID
          LIM:  component-id


J2
          LIM:  long-bar-spacing
          DIRECT-MAPPING-FUNCTION:  (lambda x (fix))
          MAPPED-TYPE:  integer


SS9
          LIM:  quality-of-concrete
          DIRECT-MAPPING-TABLE:     ((0 . "3000 psi standard mix")
                                     (1. "deduct for 2500 psi")
                                     (2. "add for 3500 psi")
                                     (3. "add for 3750 psi")
                                     (4. "add for 4000 psi"))
          INVERSE-MAPPING-TABLE:    (("3000 psi standard mix" . 0)
                                     ("deduct for 2500 psi" . 1)
                                     ("add for 3500 psi" . 2)
                                     ("add for 3750 psi" . 3)
                                     ("add for 4000 psi" . 4))
          MAPPED-TYPE:  string
```

4.4. Global Schema

The global schema is formed by taking the union of the local frame-based schemata using the global terminology. All entity types and all slots are represented in the global schema. Since LFBSs may differ with respect to terminology (the names for common frames and slots) and slot domains (data types and dimensions), the establishment of the global schema involves the selection of a single set of global names and domains. This selection is performed by the *global schema administrator,* who is responsible for the consistency and completeness of the global schema. The KADBASE prototype does not perform automated schema integration. The concluding chapter of the thesis describes some of the considerations relevant to automated schema integration.

As in the LFBS, the entities and slots in the global schema are defined as frames and linked into a hierarchy via the *part-of-schema* relationship. The GS frame is the top node of the hierarchy; the global entity definition frames are its direct descendants.

4.4.1. GS Entity Definition

The description of entities in the global schema shown in the following sections parallels the LFBS entity definition in Section 4.1.1

- GLOBAL-ENTITY-NAME — the global name for the entity. The entity name must be a *label.*
- PRIMARY-KEY— a single attribute or a list of attributes and relationships that serves to uniquely identify the entity.[13]
- CANDIDATE-KEYS — the single attributes or lists of attributes and relationships, each of which uniquely identifies an instance of the entity. The primary key is by definition a candidate key, and therefore must be a value for this slot.
- PART-OF-SCHEMA — the link to the GS frame.
- SCHEMA-SUBPART — the global slots associated with this entity.

The construction quantity and unit-cost entities in the HICOST-II global schema are defined as follows.

construction-quantity

GLOBAL-ENTITY-NAME: construction-quantity
PRIMARY-KEY: (component-id quantity-id)
CANDIDATE-KEYS: (component-id quantity-id)
PART-OF-SCHEMA: GS

unit-cost

GLOBAL-ENTITY-NAME: unit-cost
PRIMARY-KEY: quantity-id
CANDIDATE-KEYS: quantity-id
PART-OF-SCHEMA: GS

4.4.2. GS Slot Definition

Slots represent attributes or relationships with other entities. The following general information is required to describe a slot irrespective of its type:

- GLOBAL-SLOT-NAME — the name of the slot in the global schema. The slot name must be a *label.*
- SLOT-TYPE — the role of the slot in the global data structure; either *attribute* or *relationship.*
- DATA-TYPE — global data type; may be one of *integer, decimal, real, number* (may

[13] The primary key may be superfluous at the global level because each LFBS may use a different element from the list of candidate keys as a primary key. However, this implementation of KADB ASE uses primary keys (in fact, single attribute primary keys) for convenience whenever possible.

be either real or integer), *character* (treated as strings of length 1), *boolean,* or *string* (string may include an optional length specification, e.g., string[10]). A relationship slot is restricted to integers, characters, or strings.

The following special information is required for slots that define attributes.

- UNITS — feet, inches, lbs/If, dollars, etc. Only those units that have been defined in the global set of unit descriptions are allowed.[14]

The following special information is required for slots that define relationships.

- RELATIONSHIP-TYPE — the name of the underlying relationship type (or its inverse); e.g., *is-a, part-of, sub-part, instance,* or name of the special relationship definition, if applicable.
- RANGE — the entity types that may be referenced by the relationship.

The following frames define the COMPONENT-ID and QUANTITY-ID slots for the construction quantity entity and the QUANTITY-ID and UNIT-COST slots for the unit cost entity.

component-id

GLOBAL-SLOT-NAME: component-id
SLOT-TYPE: relationship
DATA-TYPE: symbol
RELATIONSHIP-TYPE: is-a
RANGE: concrete-footing
PART-OF-SCHEMA: construction-quantity

quantity-id

GLOBAL-SLOT-NAME: quantity-id
SLOT-TYPE: relationship
DATA-TYPE: symbol
RELATIONSHIP-TYPE: is-a
RANGE: unit-cost
PART-OF-SCHEMA: construction-quantity

quantity-id

GLOBAL-SLOT-NAME: quantity-id
SLOT-TYPE: attribute
DATA-TYPE: integer
PART-OF-SCHEMA: unit-cost

unit-cost

GLOBAL-SLOT-NAME: unit-cost
SLOT-TYPE: attribute
DATA-TYPE: real
UNITS: dollars
PART-OF-SCHEMA: unit-cost

[14]A suggested alternative to the use of specific units in the global schema is the assignment of *unit types* to slots at the global level; e.g., length, area, volume, time, energy, grad-student-years, etc. However, differences between the units in local schemata must still be resolved. It is better to resolve them on a local to global basis rather than a local to local to local to local to ..., ad infinitum.

See footnote on page 33 for discussion of implementation of units in KADBASE.

4.5. Global Data Source Mapping

The global data source mapping (GDSM) relates each entity and slot in the global schema to the list of LFBSs in which they can be found. The following slot is added to the entity definition frame to provide this mapping:

- SOURCE — the names of the local schemata where the entity type occurs.

The following slot is added to the slot definition frame:

- SOURCE — the names of the local schemata where the slot occurs for the indicated entity type.

The previously-defined entity and slot frames for the global schema have the following SOURCE slots.

construction-quantity
```
    PART-OF-SCHEMA:  GS
    SOURCE:  (hicpmd)
```

unit-cost
```
    PART-OF-SCHEMA:  GS
    SOURCE:  (ucd)
```

component-id
```
    PART-OF-SCHEMA:  construction-quantity
    SOURCE:  (hicpmd)
```

quantity-id
```
    PART-OF-SCHEMA:  construction-quantity
    SOURCE:  (hicpmd)
```

quantity-id
```
    PART-OF-SCHEMA:  unit-cost
    SOURCE:  (ucd)
```

unit-cost
```
    PART-OF-SCHEMA:  unit-cost
    SOURCE:
```

The abbreviations *hie, pmd,* and *ucd* represent HICOST-II, the project management database, and the unit cost database respectively. Note that the UNIT-COST slot of the **unit-cost** entity does not have any sources.

4.6. Global Integration Mapping

The global integration mapping (GIM) consists of constraints that define mathematical relationships between the attributes of an entity at the global level. The constraint mappings are intended to relate attributes from different data sources. Each constraint represents one global attribute as an arithmetic expression of other attributes. To avoid problems with entity variable scoping, the slots within a constraint may come from only one entity type. A combination of constraints and inheritance through relationships is used to represent multi-entity constraints.

A constraint is defined as a separate frame in the global integration mapping, and it is linked to the constrained slot by means of the CONSTRAINT slot in the global schema slot frame. The following slots are required for each constraint frame.

- CONSTRAINED-SLOT — the name of the slot that is being constrained; i.e., the slot that is alone on one side of the equality expression for the constraint.
- CONSTRAINT-SLOTS — the names of slots that are part of the constraint, excluding the constrained slot; i.e., the *ingredients* of the constraint expression.
- CONSTRAINT-EXPRESSION — the constraint represented as a tree of frames in terms of the KADBASE message data structure (see Chapter 3).

The HICOST-II global schema uses the following constraint to represent the combination of the material and installation unit costs into a single unit cost value for the unit cost entity.

UNIT-COST == MATLCOST + INSTCOST

That constraint is represented by the constraint frame and the five **message tree node** frames shown below.

unit-cost-constraint

CONSTRAINED-SLOT: unit-cost
CONSTRAINT-SLOTS: matlcost instcost
CONSTRAINT-EXPRESSION: (+ matlcost instcost)

nodel

IS-A: kad-op==
PART-OF-MESSAGE: **unit-cost-constraint**
MESSAGE-SUBPART: **node2 node3**

node2

IS-A: kad-slot-value
PART-OF-MESSAGE: **nodel**
SLOT-REFERENCE: unit-cost

node3

IS-A: kad-op+
PART-OF-MESSAGE: **nodel**
MESSAGE-SUBPART: **node4 node5**

node4

IS-A: kad-slot-value
PART-OF-MESSAGE: **node3**
SLOT-REFERENCE: matlcost

nodeS

IS-A: kad-slot-value
PART-OF-MESSAGE: node3
SLOT-REFERENCE: instcost

4.7. Schema Definition Utility

The schema definition utility builds the frames that define the schema description components using input in the form of a problem-oriented language (POL). The complete definition of the POL is given below and on the following pages. The formalisms used in the definition of the POL in this section are the same as used those in the definition of KQL (see Chapter 2). The keywords used in the language are basically the same as those used for the slot names in the schema description frames. For convenience, the POL permits the LFBS, LFBM, and LIM to be defined together or separately; likewise, the GS, GDSM, and GIM may be defined simultaneously. This feature speeds the language processing and groups the information logically according to entities and slots instead of mapping usage.

<LFBS description := **lfbs** <lfbs name>
<lfbs entities>
[<relationship range declarations>][15]

<lfbs entities> := <lfbs entity description> |
<lfbs entities> <lfbs entity description>

<lfbs entity description> := **entity** <entity name>
lfbm <local data structure name>
lim <global entity name>
<lfbs slots>
primary-key <slotlist>
candidate-key <slot lists>

<slotlist> ::= <slotname> | <slot list> <slot name>

<slotlists> ::= <slotlist> | <slot listsxslot lists>

<lfbs slots> := <lfbs slot description> |
<lfbs slots> <lfbs slot description>

<lfbs slot description> := **slot** <slot name>
slot-type <slot type>
data-type <datatype>
[**units** <unit name>]
[**relationship-type** <relationship type>]
lfbm <lfbm slots>
[**lim** <global slot name>]
[**mapped-type** <data type>]
[**mapping-table** (<mapping table>)]
[**direct-function** <lisp expression>]
[**inverse-function** <lisp expression>]

<slottype> ::= **attribute | relationship**

<datatype> ::= integer | real | decimal | number |
character | string[[<integer>]] | boolean

<lfbm slots> ::= (<local data structure name> . <local attribute name>) |
<lfbm slots> (<local data structure name> . <local attribute name>)

<mapping table> ::= (<local value> . <global value>) |
<mapping table> (<local value> . <global value>)

<relationship type> ::= **is-a** | **part-of | instance |**
is-a+inv | subpart | instance-finv

<relationship range declarations> := <entity range declaration> |
<relationship range declarations>
<entity range declaration>

<entity range declaration> := entity <entity name> <slot range declaration> |
<entity range declaration> <slot range declaration>

[15] The relationship range declarations are listed last because all the entities involved in the relationship must be defined prior to the declaration since the schema definition utility checks the validity of the entities involved.

<slot range declaration> := **slot** <slot name> <range declaration>
<slot range declaration> <range declaration>

<range declaration> := **range** <entity name>

<GS description> := **global-schema**
<GS entities>
[<entity constraint declarations:^[16]
[<relationship range declarations>]

<GS entities> ::= <GS entity description> |
<GS entities> <GS entity description>

<GS entity description> := **entity** <entity name>
<GS slot descriptions>
primary-key <slotlist>
candidate-key <slot lists>

<GS slots> ::= <GS slot description> |
<GS slots> <GS slot description>

<GS slot description> := slot <slot name>
slot-type <slottype>
data-type <datatype>
[**units** <unit name>]
[relationship-type <relationship-type>]
[**sources** (<data sources>)]

<data sources> ::= <data source name> |
<data sources> <data source name>

<entity constraint declaration> := **entity** <entity name> <slot constraint declaration> |
<entity constraint declaration>
<slot constraint declaration>

<slot constraint declaration> := **slot** <slot name> <constraint declaration>
<slot constraint declaration> <constraint declaration>

<constraint declaration> := **constraint** (<clause[17]>)

To demonstrate the use of the schema definition language, the definitions for the local and global schema examples used in this chapter are presented on the following pages. The complete versions of these schemata are found in the HICOST-II supplement in Chapter 7.

[16] The constraint declarations are last because all the slots involved in the constraint must be defined prior to the declaration since the schema definition utility checks the validity of tho slots involved.

[17] See Chapter 2 for definition of *clause.*

```
lfbs sample-lfbs
entity concrete-footing
lfbm cf__att
lfbm cf__spec
lim concrete-footing
   slot ID
      slot-type attribute
      data-type integer
      lfbm (cf__att . ID) (cf__spec . ID)
      lim component-id
   slot J2
      slot-type attribute
      data-type real
      units inches
      lfbm (cf__att . J)
      lim long-bar-spacing
      direct-mapping-function lambda x fix
      mapped-type  integer
   slot SS9
      slot-type attribute
      data-type integer
      lfbm (cf__spec . SS9)
      lim quality-of-concrete
      direct-mapping-table 0 . "3000 psi standard mix"
                           1 . "deduct for 2500 psi"
                           2 . "add for 3500 psi"
                           3 . "add for 3750 psi"
                           4 . "add for 4000 psi"
      inverse-mapping-table "3000 psi standard mix" . 0
                            "deduct for 2500 psi" . 1
                            "add for 3500 psi" . 2
                            "add for 3750 psi" . 3
                            "add for 4000 psi" . 4
primary-key ID
candidate-keys ID
```

```
global-schema
entity construction-quantity
source (hie pmd)
   slot component-id
      slot-type relationship
      data-type symbol
      relationship-type is-a
      range concrete-footing
      source (hie pmd)
   slot quantity-id
      slot-type relationship
      data-type symbol
      relationship-type is-a
      range unit-cost
      source (hie pmd)
primary-key  component-id quantity-id
candidate-keys  component-id quantity-id

entity unit-cost
source (ucd)
   slot quantity-id
      slot-type attribute
      data-type integer
      source (ucd)
   slot unit-cost
      slot-type attribute
      data-type real
      units dollars
          constraint (unit-cost.unit-cost —•• unit-cost.matlcost +
                                             unit-cost.instcost)
primary-key  quantity-id
candidate-keys  quantity-id
```

Chapter 5
KADBASE Communications

The KADBASE communications utilities provide the means to transfer messages (expressed as frames in the form described in Chapter 3) between KADBASE components. The utilities are intended to hide the implementation-specific details of communications protocols from the communicating components. Therefore, the top level functions are very simple:

- *K2i&-message-name{component-name)* — returns a unique message name formed from the requesting component's name[18] and a local time stamp.
- *K.ad'Send-message(message-frame)* — sends the message represented by the message frame instance to the component indicated in the TO slot of the message frame.
- *Kad-check-message(component-name)* — checks to see if a message has arrived for the indicated component. The function returns the name of the message frame if found.
- *Kad-wait-for-message(component-name)* — waits until a message arrives for the indicated component. The function returns the name of the message frame when it arrives.

These four functions are all that a KADBASE component needs to know about communications. The following sections describe the basic implementation details ofKADBASE communications.

5.1. Message Files

KADBASE messages are transmitted as ASCII files, containing the information necessary to reassemble the message frame and all its descendants in another component. The sending component is responsible for creating the ASCII message file from the message frames, and the receiving component is responsible for recreating the message frames from the information in the message file. The name of the file is the same as the name of the top-level message frame. The form of the ASCII message file is implementation-dependent. Update and reply messages may have auxiliary files that contains data to augment the frame representation of the message.

In the Lisp-Framekit environment, messages files represent message frames using the Framekit functions for building frames. An outgoing message is written to a file in the Framekit format, and an incoming message is read using the Lisp "load" function.

[1] Components are identified by unique three-character names; e.g., "kui" for KADBASE User Interface, "nda" for Network Data Access Manager, etc.

5.2. Sending Messages

To send a message, a component first requests a unique message name using the **kad-message-name** function. Then, the component creates a message frame instance having that name; fills the MESSAGE-TYPE, TIME, FROM, and TO slots; and attaches the request or reply frame via the MESSAGE-SUBPART relationship slot. The sending component need not know anything about the addressee other than its three-character KADBASE component name. Finally, the component calls **kad-send-message** with the message frame as the argument.

Messages for KADBASE components are placed in special message file directories by **kad-send-message.** A separate directory is used for each component; e.g., ^{M}/../cive/usr/hch/kad/kui/message/in/^{M} for the KADBASE user interface ("kui"). The message directories constitute the absolute *addresses* of the components and are stored in a global frame for access by the function **kad-send-message.** The message file transfer is implemented in one of the following two ways, depending on the location of the sending and receiving components in the computer network:

- *Intra-machine transfers* are accomplished by using operating system utilities (e.g., UNIX "cp") to copy the message file to the indicated directory.
- *Inter-machine transfers* are accomplished by using network utilities (e.g., TCP-IP "ftp") to copy the message to the indicated directory on the indicated machine.

5.3. Receiving Messages

The arrival of a message is indicated by the presence of a file in the message directory. The function **kad-wait-for-message** waits for a message to appear in the component's message directory, loads the message frame and its descendants into memory, removes the file, and returns the name of the message frame. The function **kad-get-message** performs the same set of actions if a message file is present when the function is invoked. If no message file is present, **kad-get-message** performs no actions and returns a "null" value.

Chapter 6
SPEX Supplement

This chapter is intended to supplement the description of the SPEX-KADBASE implementation found in Section 5.1 of the thesis. The supplement contains the schema description information for the SPEX KBS context and the AISC database (including the LFBS, LFBM, and LIM for each component) as well as the GS, GDSM, and GIM. This supplement does not reproduce the data contained in the AISC database as that data is readily available in the *AISC Manual of Steel Construction* [AISC 80]. The chapter concludes with traces of the components (SPEX and the AISC KBDBI) at runtime.

6.1. Schemata

The schema descriptions are expressed in the problem-oriented language of the schema description utility (described in Section 4.7). Each of the component description sections contains the local frame-based schema (LFBS), the local frame-based mapping (LFBM), and the local integration mapping (LIM) grouped into a single set of definitions. The final subsection presents the global schema (GS), the global data source mapping (GDSM), and the global integration mapping (GIM) in the same fashion.

6.1.1. SPEX Context Schemata

```
lfbs spx

entity w__shape
lfbm w__shape
lim wide-flange-shape
  slot designation
    slot-type attribute
    data-type string
    lfbm (w__shape . designation)
  slot nom__depth
    slot-type attribute
    data-type real
    lfbm (w_shape . nom_depth)
    lim nominal-depth
  slot weight
    slot-type attribute
    data-type real
    lfbm (w__shape . weight)
  slot area
    slot-type attribute
    data-type real
    lfbm (w_shape . area)
```

```
slot d
  slot-type attribute
  data-type real
  lfbm (w_shape . d)
  lim depth
slot tw
  slot-type attribute
  data-type real
  lfbm (w_shape . tw)
  lim web-thickness
slot bf
  slot-type attribute
  data-type real
  lfbm (w_shape . bf)
  lim flange-width
slot tf
  slot-type attribute
  data-type real
  lfbm (w__shape . tf)
  lim flange-thickness
slot k
  slot-type attribute
  data-type real
  lfbm (w_shape . k)
slot bf2tf
  slot-type attribute
  data-type real
  lfbm (w_shape . bf2tf)
slot fyp
  slot-type attribute
  data-type real
  lfbm (w_shape . fyp)
slot dtw
  slot-type attribute
  data-type real
  lfbm (w_shape . dtw)
slot fyppp
  slot-type attribute
  data-type real
  lfbm (w__shape . fyppp)
slot rt
  slot-type attribute
  data-type real
  lfbm (w__shape . rt)
slot daf
  slot-type attribute
  data-type real
  lfbm (w__shape . daf)
slot ri
  slot-type attribute
  data-type real
  lfbm (w_shape . ri)
slot ra
  slot-type attribute
  data-type real
  lfbm (w__shape . ra)
slot nt
  slot-type attribute
  data-type real
  lfbm (w_shape . nt)
slot ix
  slot-type attribute
  data-type real
  lfbm (w_shape . Ix)
  lim moment-of-inertia-x
slot sx
```

```
    slot-type attribute
    data-type real
    lfbm (w_shape . Sxc)
    lfbm (w_shape . Sxt)
    lim section-modulus-x
  slot rx
    slot-type attribute
    data-type real
    lfbm (w_shape . rx)
    lim radius-of-gyration-x
  slot iy
    slot-type attribute
    data-type real
    lfbm (w_shape . Iy)
    lim moment-of-inertia-y
  slot sy
    slot-type attribute
    data-type real
    lfbm (w_shape . Syt)
    lfbm (w_shape . Syc)
    lim section-modulus-y
  slot ry
    slot-type attribute
    data-type real
    lfbm (w_shape . ry)
    lim radius-of-gyration-y
  slot j
    slot-type attribute
    data-type real
    lfbm (w_shape . j)
    lim torsional-constant
  slot zx
    slot-type attribute
    data-type real
    lfbm (w_shape . Zx)
    lim plastic-modulus-x
  slot zy
    slot-type attribute
    data-type real
    lfbm (w_shape . Zy)
    lim plastic-modulus-x
candidate-key designation
primary-key designation
;
```

6.1.2. AISC Database Schemata

```
lfbs ais

entity w__shape
lfbm w_shape
lim wide-flange-shape
  slot designation
    slot-type attribute
    data-type stringl2
    lfbm (w_shape . designation)
  slot nom_depth
    slot-type attribute
    data-type real
    lfbm (w_shape . nom_depth)
    lim nominal-depth
  slot weight
  slot-type attribute
  data-type real
  lfbm (w_shape . weight)
slot area
  slot-type attribute
  data-type real
  lfbm (w__shape . area)
slot depth
  slot-type attribute
  data-type real
  lfbm (w__shape . depth)
slot tw
  slot-type attribute
  data-type real
  lfbm (w__shape . tw)
  lim web-thickness
slot bf
  slot-type attribute
  data-type real
  lfbm (w_shape . bf)
  lim flange-width
slot tf
  slot-type attribute
  data-type real
  lfbm (w_shape . tf)
  lim flange-thickness
slot k
  slot-type attribute
  data-type real
  lfbm (w_shape . k)
slot bf2tf
  slot-type attribute
  data-type real
  lfbm (w_shape . bf2tf)
slot fyp
  slot-type attribute
  data-type real
  lfbm (w__shape . fyp)
slot dtw
  slot-type attribute
  data-type real
  lfbm (w_shape . dtw)
slot fyppp
  slot-type attribute
  data-type real
  lfbm (w_shape . fyppp)
slot rt
  slot-type attribute
  data-type real
  lfbm (w__shape . rt)
slot daf
  slot-type attribute
  data-type real
  lfbm (w__shape . daf)
slot ri
  slot-type attribute
  data-type real
  lfbm (w_shape . ri)
slot ra
  slot-type attribute
  data-type real
  lfbm (w_shape . ra)
slot nt
  slot-type attribute
  data-type real
  lfbm (w_shape . nt)
  slot ix
    slot-type attribute
    data-type real
    lfbm (w_shape . ix)
    lim moment-of-inertia-x
  slot sx
    slot-type attribute
    data-type real
    lfbm (w_shape . sx)
    lim section-modulus-x
  slot rx
    slot-type attribute
    data-type real
    lfbm (w_shape . rx)
    lim radius-of-gyration-x
  slot iy
    slot-type attribute
    data-type real
    lfbm (w_shape . iy)
    lim moment-of-inertia-y
  slot sy
    slot-type attribute
    data-type real
    lfbm (w__shape . sy)
    lim section-modulus-y
  slot ry
    slot-type attribute
    data-type real
    lfbm (w__shape . ry)
    lim radius-of-gyration-y
  slot j
    slot-type attribute
    data-type real
    lfbm (w_shape . j)
    lim torsional-constant
  slot zx
    slot-type attribute
    data-type real
    lfbm (w_shape . zx)
    lim plastic-modulus-x
  slot zy
    slot-type attribute
    data-type real
    lfbm (w__shape . zy)
    lim plastic-modulus-x
candidate-key designation
primary-key designation
;
```

6.1.3. Global Schemata

```
global-schema

entity wide-flange-shape
sources (spx ais)
  slot designation
    slot-type attribute
    data-type string
    sources (spx ais)
  slot nominal-depth
    slot-type attribute
    data-type real
  sources (spx ais)
slot weight
  slot-type attribute
  data-type real
  sources (spx ais)
slot area
  slot-type attribute
  data-type real
  sources (spx ais)
slot depth
  slot-type attribute
  data-type real
  sources (spx ais)
slot web-thickness
  slot-type attribute
  data-type real
  sources (spx ais)
slot flange-width
  slot-type attribute
  data-type real
  sources (spx ais)
slot flange-thickness
  slot-type attribute
  data-type real
  sources (spx ais)
slot k
  slot-type attribute
  data-type real
  sources (spx ais)
slot bf2tf
  slot-type attribute
  data-type real
  sources (spx ais)
slot fyp
  slot-type attribute
  data-type real
  sources (spx ais)
slot dtw
  slot-type attribute
  data-type real
  sources (spx ais)
slot fyppp
  slot-type attribute
  data-type real
  sources (spx ais)
slot rt
  slot-type attribute
  data-type real
  sources (spx ais)
slot daf
  slot-type attribute
  data-type real
  sources (spx ais)
slot ri
  slot-type attribute
  data-type real
  sources (spx ais)
slot ra
  slot-type attribute
  data-type real
  sources (spx ais)
slot nt
  slot-type attribute
  data-type real
  sources (spx ais)
slot moment-of-inertia-x
```

```
    slot-type attribute
    data-type real
    sources (spx ais)
  slot section-modulus-x
    slot-type attribute
    data-type real
    sources (spx ais)
  slot radius-of-gyration-x
    slot-type attribute
    data-type real
    sources (spx ais)
  slot moment-of-inertia-y
    slot-type attribute
    data-type real
    sources (spx ais)
  slot section-modulus-y
    slot-type attribute
    data-type real
    sources (spx ais)
  slot radius-of-gyration-y
    slot-type attribute
    data-type real
    sources (spx ais)
  slot torsional-constant
    slot-type attribute
    data-type real
    sources (spx ais)
  slot plastic-modulus-x
    slot-type attribute
    data-type real
    sources (spx ais)
  slot plastic-modulus-x
    slot-type attribute
    data-type real
    sources (spx ais)
candidate-key designation
primary-key designation
;
```

6.2. Program Traces

This section contains complete listings of the output for SPEX and the AISC KBDBI corresponding to the example described in Section 5.1.3 of the thesis. The message frames corresponding to this example are found in Section 3.3 of this report.

6.2.1. SPEX Trace

The listing starting below and continuing on the following pages shows the SPEX terminal output. The listing begins with the loading of the constituent Lisp code files, proceeds through the query formulation and processing by the SPEX KB SI, and concludes with the SPEX post-processing.

```
CMULisp, 5-16-86, from Franz Lisp, Opus 38

1.(load 'spex)
```

```
 [load spex.l]
 [fasl /usrceO/jhg/framekit/rulekit.o]
 [load /usrceO/jhg/framekit/general.1]
 [load /usrceO/jhg/framekit/macros.1]
 [load /usrceO/jhg/framekit/machacks.1]
 [fasl /usrceO/jhg/framekit/bigframe.o]
 [load /usr/jhg/thesis/spex/cmuenv.1]
 [fasl /usr/lisp/lisplib/cmumacs.o]
 [fasl /usr/lisp/lisplib/cmufncs.o]
 [fasl /usr/lisp/lisplib/cmutpl.o]
 [fasl /usr/lisp/lisplib/cmufile.o]
 [fasl /usr/jhg/thesis/spex/auxfens.o]
 [load /usr/jhg/thesis/spex/frame-fun.l]
 [load /usr/jhg/thesis/spex/class-st.1]
 [load /usr/jhg/thesis/spex/bboard.1]
 [fasl /usr/jhg/thesis/spex/syscont.o]
 [load /usr/jhg/thesis/spex/trks.1]
 [fasl /usr/jhg/thesis/spex/scanner.o]
 [load /usr/jhg/thesis/spex/dfks.l]
 [load /usr/jhg/thesis/spex/dfks-mod.1]
 [load /usr/jhg/thesis/spex/rulebd.1]
 [load /usr/jhg/thesis/spex/cgks.1]
 [load /usr/jhg/thesis/spex/prks.1]
 [load /usr/jhg/thesis/spex/csgks.1]
 [load /usr/jhg/thesis/spex/csg-mod.l]
 [load /usr/jhg/thesis/spex/deriv.1]
 [load /usr/jhg/thesis/spex/cssks.1]
 [load /usr/jhg/thesis/spex/cssks-mod.1]

Loading OPT as a foreign function
/usr/misc/.lisp/lisplib/nld -N -x -A /usr/misc/bin/cmulisp -T 131a00
   /usr/jhg/thesis/spex/opt.o -e _opt_ -b /tmp/Li8356.0 -1F77 -1177 -lc
 [load /usr/jhg/thesis/spex/evks.1]
 [load /usr/jhg/thesis/spex/dte.1]
 [load /usr/jhg/thesis/spex/post-pro.1]
 [load /usr/jhg/thesis/spex/debug-funs.1]
 [load /usr/jhg/thesis/spex/kadbase.1]
 [load /usr/hch/kad/kad-spex/spex-kbsi-load.l]
 [fasl /usr/hch/src/lisp/util.o]
 [fasl /usr/hch/src/lisp/fkutil.o]
 [fasl /usr/kbes/bin/bigframe.o]
 [fasl /usr/hch/kad/kadutil.o]
 [fasl /usr/hch/kad/kui/parse/kui-scanner.o]
 [fasl /usr/hch/kad/comm/schema/message.o]
 [fasl /usr/hch/kad/comm/kadcomm.o]
/usr/misc/.lisp/lisplib/nld -N -x -A /tmp/Li8356.0 -T 17d600
   /usr/hch/kad/comm/kadcommutil.o -e __kadfilecopy -o /tmp/Li8356.1  -lc
 [load /usr/hch/kad/comm/kad-address.1]
 [fasl /usr/hch/kad/kui/kui.o]
 [fasl /usr/hch/kad/kui/parse/parseschema.o]
 [fasl /usr/hch/kad/kui/parse/transutil.o]
 [fasl /usr/hch/kad/kui/parse/winston.o]
 [fasl /usr/hch/kad/map/schema.o]
 [fasl /usr/hch/kad/map/semutil.o]
 [fasl /usr/hch/kad/map/semslot.o]
 [fasl /usr/hch/kad/map/semreply.o]
 [load /usr/hch/kad/kad-spex/spex.l.schema]
 [load /usr/hch/kad/db/aisc/aisc-gs.1.schema]
 [fasl /usr/hch/kad/kad-spex/spex-kbsi.o]
rm: /usr/hch/kad/kad-spex/message/in/* nonexistent
 [load c-rules.l]
.......**.....*....*.....*......*...*............*.*..........*..*.****.

*                        WELCOME TO SPEX                             *
*    As an interactive user of SPEX, you are first required to input   *
*   the task specification.  Your input will be parsed until you type  *
*   the any of these keywords "end", "stop", "bye", "quit", "halt".    *
*   If you desire help on the input syntax, type the keyword "help".   *
*                                                                      .
•••••••••••••••a*************************************************************

SPEX> input init5b.dat [load /usr/jhg/thesis/spex/lrfd.1]
[load /usr/jhg/thesis/spex/lrfd-ct.1]
[load /usr/jhg/thesis/spex/dl-bt.l]
SPEX> quit

The following message was just posted : DESIGN_FOCUS_COMPLETE

1 <Enter> do-all-actions ((read-message 'DESIGN_FOCUS_COMPLETE) (prks) (pp
   requirement-check__list) (pp requirement-satisfy-list))
|2 <Enter> do-all-actions ((prks) (pp requirement-check__list) (pp
   requirement-satisfy-list))


The following message was just posted : REQUIREMENTS_RETRIEVED

| 3 <Enter> do-all-actions ((pp requirement-check__list) (pp requirement-
   satisfy-list))

| |4 <Enter> do-all-actions ((pp requirement-satisfy-list))

(defprop requirement-satisfy-list
         (requirement-satisfy-list (set (value lrfd-LFB-average)))
         frame:)

| I 5 <Enter> do-all-actions nil
| I 5 <EXIT> do-all-actions t
| 14 <EXIT> do-all-actions t
| 3 <EXIT> do-all-actions t
|2 <EXIT> do-all-actions t
1 <EXIT> do-all-actions t
1 <Enter> do-all-actions ((read-message 'REQUIREMENTS_RETRIEVED) (csgks)
   (debug-outputl))
12 <Enter> do-all-actions ((csgks) (debug-outputl))


The following message was just posted : CONSTRAINT_SET_GENERATED

I 3 <Enter> do-all-actions ((debug-outputl))




(defprop constraint-frames
         (constraint-frames
          (set
           (value constraint-ct-def
                  constraint-h-def
                  constraint-Af-def
                  constraint-Ix-def
                  constraint-Sxt-def
                  constraint-Mr-action
                  constraint-Mp-action
                  constraint-Cb-action
                  constraint-Mn-flbl-action
                  constraint-LFB-AVERAGE-C3
                  constraint-Fr-action
                  constraint-LFB-AVERAGE-c2
                  constraint-LFB-AVERAGE-cl)))
         frame:)

(defprop constraint-ct-def
         (constraint-ct-def  (is-a (value constraint-frame))
                             (definition (value yes))
                             (data-item (value ct))
                             (forced-action-of (value constraint-Sxt-def))
                             (member-of (value constraint-frames))
                             (expression (value (ct - (d // 2))))
                             (expr-type (value numeric)))
         frame:)

(defprop constraint-h-def
         (constraint-h-def  (is-a (value constraint-frame))
                            (definition (value yes))
                            (data-item (value h))
                            (forced-action-of (value constraint-Ix-def))
                            (member-of (value constraint-frames))
                            (expression (value (h - (d - (2 * tf)))))
                            (expr-type (value numeric)))
         frame:)

(defprop constraint-Af-def
         (constraint-Af-def  (is-a (value constraint-frame))
                             (definition (value yes))
                             (data-item (value Af))
                             (forced-action-of (value constraint-Ix-def))
                             (member-of (value constraint-frames))
                             (expression (value (Af - (bf * tf))))
                             (expr-type (value numeric)))
         frame:)

(defprop constraint-Ix-def
         (constraint-Ix-def  (is-a (value constraint-frame))
                             (definition (value yes) )
                             (data-item (value Ix))
                             (forced-action-of (value constraint-Sxt-def))
                             (member-of (value constraint-frames))
                             (expression
                              (value
                               (Ix =
                                   (Af *
                                       (((h + tf) // 2) ** 2)
                                      +
                                       (tw // 12)
                                      •
                                       (h ** 3)))))
                             (expr-type (value numeric))
                             (forced-actions
                              (value constraint-h-def constraint-Af-def)))
         frame:)

(defprop constraint-Sxt-def
         (constraint-Sxt-def  (is-a (value constraint-frame))
                              (definition (value yes))
                              (data-item (value Sxt))
                              (forced-action-of (value constraint-Mr-action))
                              (member-of (value constraint-frames))
                              (expression (value (Sxt = (Ix // ct))))
                              (expr-type (value numeric))
                              (forced-actions
                              (value  constraint-ct-def  constraint-Ix-def)))
          frame:)


(defprop  constraint-Mr-action
          (constraint-Mr-action  (is-a  (value  constraint-frame))
                                 (definition  (value  yes))
                                 (data-item  (value  Mr))
                                 (forced-action-of
                                  (value  constraint-Mn-flbl-action))
                                 (member-of  (value  constraint-frames))
                                 (expression  (value  (Mr  =  (Sxt  *  (36.0  -
  Fr)))))
                                 (expr-type  (value  numeric))
                                 (forced-actions  (value  constraint-Sxt-def)))
          frame:)


(defprop  constraint-Mp-action
          (constraint-Mp-action  (is-a  (value  constraint-frame))
                                 (definition  (value  yes))
                                 (data-item  (value  Mp))
                                 (forced-action-of
                                  (value  constraint-Mn-flbl-action))
                                 (member-of  (value  constraint-frames))
                                 (expression  (value  (Mp  =  (36.0  *  Zx))))
                                 (expr-type  (value  numeric)))
          frame:)


(defprop  constraint-Cb-action
          (constraint-Cb-action  (is-a  (value  constraint-frame))
                                 (definition  (value  yes))
                                 (data-item  (value  Cb))
                                 (forced-action-of
                                  (value  constraint-Mn-flbl-action))
                                 (member-of  (value  constraint-frames))
                                 (expression  (value  (Cb  =  (1.0))))
                                 (expr-type  (value  numeric)))
          frame:)


(defprop  constraint-Mn-flbl-action
          (constraint-Mn-flbl-action  (is-a  (value  constraint-frame))
                                      (definition  (value  yes))
                                      (data-item  (value  Mn-flbl))
                                      (forced-action-of
                                       (value  constraint-LFB-AVERAGE-c3))
                                      (member-of  (value  constraint-frames))
                                      (expression
                                       (value
                                        (Mn-flbl  =
                                                 (0.9  *
                                                      Cb
                                                      •

;  <««  start  back  on  the  left  <««
      (Mp -
          (Mp - Mr)
          •
          <((bf // (2 * tf)) - (65 // sqrt (36.0)))
           //
           ((147 // sqrt (36.0 - Fr)) - (65 // sqrt (36.0)))))
/  >»»  continue  on  the  right  »»>
))))
                                      (expr-type  (value  numeric))
```

```
                                     (forced-actions
                                      (value constraint-Mr-action
                                             constraint-Mp-action
                                             constraint-Cb-action)))
         frame:)


(defprop constraint-LFB-AVERAGE-c3
         (constraint-LFB-AVERAGE-c3  (is-a (value constraint-frame))
                                     (forced-constraint-of
                                      (value lrfd-LFB-average))
                                     (expression (value (2400.0 <« Mn-flbl)))
                                     (type (value performance))
                                     (expr-type (value numeric))
                                     (member-of (value constraint-frames))
                                     (forced-actions
                                      (value constraint-Mn-flbl-action)))
         frame:)


(defprop constraint-Fr-action
         (constraint-Fr-action  (is-a (value constraint-frame))
                                (definition (value yes))
                                (data-item (value Fr))
                                (forced-action-of
                                 (value constraint-LFB-AVERAGE-c2))
                                (member-of (value constraint-frames))
                                (expression (value (Fr = (10.0))))
                                (expr-type (value numeric)))
         frame:)


(defprop constraint-LFB-AVERAGE-c2
         (constraint-LFB-AVERAGE-c2  (is-a (value constraint-frame))
                                     (forced-constraint-of
                                      (value lrfd-LFB-average))
                                     (expression
                                      (value
                                       ( (bf // (2 * tf))
                                        <=
                                        147
                                        //
                                        sqrt
                                         (36.0 - Fr))))
                                     (type (value applicability))
                                     (expr-type (value numeric))
                                     (member-of (value constraint-frames))
                                     (forced-actions
                                      (value constraint-Fr-action)))
         frame:)


(defprop constraint-LFB-AVERAGE-cl
         (constraint-LFB-AVERAGE-cl  (is-a (value constraint-frame))
                                     (forced-constraint-of
                                      (value lrfd-LFB-average))
                                     (expression
                                      (value
                                       ((bf // (2 * tf)) > 65 // sqrt
   (36.0))))
                                     (type (value applicability))
                                     (expr-type (value numeric))
                                     (member-of (value constraint-frames)))
         frame:)

I .14 <Enter> do-all-actions nil
```

```
|  |4 <EXIT>  do-all-actions  t
I  3 <EXIT>  do-all-actions  t
|2 <EXIT>  do-all-actions  t
1 <EXIT>  do-all-actions  t
1 <Enter> do-all-actions  ((read-message 'CONSTRAINT__SET_GENERATED)  (post-
   message  'SATISFY_CONSTRAINT_SET))
|2 <Enter> do-all-actions  ((post-message  'SATISFY_CONSTRAINT_SET))


The following message was just posted : SATISFY__CONSTRAINT_SET

|  3 <Enter> do-all-actions nil
j  3 <EXIT>  do-all-actions  t
|2 <EXIT>  do-all-actions  t
1 <EXIT>  do-all-actions  t
1 <Enter> do-all-actions  ((read-message 'SATISFY_CONSTRAINT_SET)  (cssks)
   (debug-outputl)  (debug-output2))
12 <Enter> do-all-actions  ((cssks)  (debug-outputl)  (debug-output2))


The query frame is  :


(defprop query-frame
         (query-frame  (reply
                        (if-added  (map-database-resuits-to-data-items
   '.frame)))
                       (target  (value  (Zx tw d bf tf area designation)))
                       (object  (value w__shape) )
                       (constraints
                        (value
                         ( (2400.0 <=
                                  (0.9 *
                                       (1.0)
                                       •
                                       (-(36.0 * Zx)
                                        (-(36.0 * Zx)
                                         ((((bf * tf)
                                            •
                                            (<<(d -  (2 * tf))  + tf)  // 2)
                                             **
                                            +2)
                                            (tw // 12)
                                            •
                                            ((d -  (2 * tf))  ** 3))
                                           //
                                           (d // 2))
                                          *
                                          (36.0 -  (10.0))))
                                        (((bf //  (2 * tf))
                                          -
                                          (65 //  sqrt  (36.0)))
                                         //
                                         ((147 //  sqrt  (36.0 -  (10.0)))
                                          -
                                          (65 //  sqrt  (36.0)))))))
                          ( (bf //  (2 * tf))  <= 147 // sqrt  (36.0 -  (10.0)))
                          ((bf //  (2 * tf))  > 65 // sqrt  (36.0)))))
                      (optimization  (value  (area))))
         frame:)
```

Calling KADBASE/Ingres Database

```
 I 1 <Enter> spex-kbsi (query-frame)
 I |1 <Enter> spex-kbsi-syntax-query (query-frame)
 I | 1 <Enter> spex-Jcbsi-syntax-and (((2400.0 <= (0.9 * & * & ) ) ( (bf // &) <=
   147 // sqrt ...) ((bf // &) > 65 // sqrt ...)))
 I I |2 <Enter> spex-kbsi-syntax-and ( ( ( (bf // &) <= 147 // sqrt ...) ((bf //
   &) > 65 // sqrt ...)))
 | | | 3 <Enter> spex-kbsi-syntax-and ((((bf // &) > 65 // sqrt ...)))
 | | | 3 <EXIT>  spex-kbsi-syntax-and  (((bf // (2 * tf)) > 65 // sqrt ...))
 |  |  |2 <EXIT>  spex-kbsi-syntax-and  ( ( (bf // (2 * tf)) <« 147 // sqrt ...)
   and ((bf // (2 * tf)) > 65 // sqrt ...))
 |  | 1 <EXIT>  spex-kbsi-syntax-and  ((2400.0 <- (0.9 *  (1.0)  * ( & - & * & ) ) )
   and ((bf // (2 * tf)) <- 147 // sqrt ...) and ((bf // (2 * tf)) > 65 //
   sqrt ...))
 |  |1 <EXIT>  spex-kbsi-syntax-query  spx-retrieve-g00013
 |  |1 <Enter> spex-kbsi-send (spx-retrieve-g00013)
KAD-MESSAGE-NAME:  The message name is spx520368388
KAD-SEND-MESSAGE:  Starting to send message spx520368388
KAD-SEND-FILE: sending spx520368388  to
   /../ce/usr/hch/kad/db/aisc/message/in/ .
KAD-SEND-FILE: sending spx520368388  to /../ce/usr/hch/kad/log/message/in/ .
KAD-SEND-MESSAGE:  Message sent.
 |  |1 <EXIT>  spex-kbsi-send  t
 I  |1 <Enter> spex-kbsi-receive nil
KAD-WAIT-FOR-MESSAGE:  The system name is spx
[load /../ce/usr/hch/kad/kad-spex/message/in/ais520368635]
KAD-LOAD-MESSAGE:  The message name is ais520368635
 |  |1 <EXIT>  spex-kbsi-receive  ais-reply-g00005
KAD-GS-LFBS-REPLY:  Translating reply.
 |  |1 <Enter> spex-kbsi-syntax-reply (ais-reply-g00005 query-frame)
 |  |1 <EXIT>  spex-kbsi-syntax-reply  ((311.0 1.06 14.38 12.67 1.735 ...))
KAD-DELETE-MESSAGE:  Deleting message ais520368635  for system spx
rm: /../ce/usr/hch/kad/kad-spex/message/in/ais520368639 nonexistent
rm: override protection 644 for /usr/hch/kad/kad-
   spex/message/in/ais520368635? y
 | 1 <EXIT>  spex-kbsi  t


The following message was just posted : SET_SOLUTION__SUCCESSFUL

 I 3 <Enter> do-all-actions ((debug-outputl) (debug-output2))
```

```
 (defprop constraint-frames
          (constraint-frames
           (set
            (value constraint-ct-def
                   constraint-h-def
                   constraint-Af-def
                   constraint-Ix-def
                   constraint-Sxt-def
                   constraint-Mr-action
                   constraint-Mp-action
                   constraint-Cb-action
                   constraint-Mn-flbl-action
                   constraint-LFB-AVERAGE-c3
                   constraint-Fr-action
                   constraint-LFB-AVERAGE-c2
                   constraint-LFB-AVERAGE-cl)))
          frame:)
```

```
(defprop  constraint-ct-def
          (constraint-ct-def  (is-a (value constraint-frame))
                              (definition (value yes))
                              (data-item (value ct))
                              (forced-action-of (value constraint-Sxt-def))
                              (member-of (value constraint-frames))
                              (expression (value (ct - (d // 2))))
                              (expr-type (value numeric))
                              (epsilon (value 0.1438))
                              (value (value 0.0))
                              (status (value binding)))
          frame:)


(defprop  constraint-h-def
          (constraint-h-def  (is-a (value constraint-frame))
                             (definition (value yes))
                             (data-item (value h))
                             (forced-action-of (value constraint-Ix-def))
                             (member-of (value constraint-frames))
                             (expression (value (h - (d - (2 * tf)))))
                             (expr-type (value numeric))
                             (epsilon (value 0.2182))
                             (value (value 0.0))
                             (status (value binding)))
          frame:)


(defprop  constraint-Af-def
          (constraint-Af-def  (is-a (value constraint-frame))
                              (definition (value yes))
                              (data-item (value Af))
                              (forced-action-of (value constraint-Ix-def))
                              (member-of (value constraint-frames))
                              (expression (value (Af - (bf * tf))))
                              (expr-type (value numeric))
                              (epsilon (value 0.439649))
                              (value (value 0.0))
                              (status (value binding)))
          frame:)


(defprop  constraint-Ix-def
          (constraint-Ix-def  (is-a (value constraint-frame))
                              (definition (value yes))
                              (data-item (value Ix))
                              (forced-action-of (value constraint-Sxt-def))
                              (member-of (value constraint-frames))
                              (expression
                               (value
                                (Ix =
                                    (Af *
                                        <<(h + tf) // 2) ** 2)
                                        +
                                        (tw // 12)
                                        •
                                        (h ** 3)))))
                              (expr-type (value numeric))
                              (forced-actions
                               (value constraint-h-def constraint-Af-def))
                              (epsilon (value 19.86871914923958))
                              (value (value 0.0))
                              (status (value binding)))
          frame:)
```

```
(defprop constraint-Sxt-def
         (constraint-Sxt-def  (is-a (value constraint-frame))
                              (definition (value yes))
                              (data-item (value Sxt))
                              (forced-action-of (value constraint-Mr-action))
                              (member-of (value constraint-frames))
                              (expression (value (Sxt - (Ix // ct))))
                              (expr-type (value numeric))
                              (forced-actions
                               (value constraint-ct-def constraint-Ix-def))
                              (epsilon (value 2.76338235733513))
                              (value (value 0.0))
                              (status (value binding)))
         frame:)


(defprop constraint-Mr-action
         (constraint-Mr-action  (is-a (value constraint-frame))
                                (definition (value yes))
                                (data-item (value Mr))
                                (forced-action-of
                                 (value constraint-Mn-flbl-action))
                                (member-of (value constraint-frames))
                                (expression (value (Mr = (Sxt * (36.0 -
  Fr)))))
                                (expr-type (value numeric))
                                (forced-actions (value constraint-Sxt-def))
                                (epsilon (value 71.84794129071338))
                                (value (value 0.0))
                                (status (value binding)))
         frame:)


(defprop constraint-Mp-action
         (constraint-Mp-action  (is-a (value constraint-frame))
                                (definition (value yes))
                                (data-item (value Mp))
                                (forced-action-of
                                 (value constraint-Mn-flbl-action))
                                (member-of (value constraint-frames))
                                (expression (value (Mp = (36.0 * Zx))))
                                (expr-type (value numeric))
                                (epsilon (value 223.92))
                                (value (value 0.0))
                                (status (value binding)))
         frame:)


(defprop constraint-Cb-action
         (constraint-Cb-action  (is-a (value constraint-frame))
                                (definition (value yes))
                                (data-item (value Cb))
                                (forced-action-of
                                 (value constraint-Mn-flbl-action))
                                (member-of (value constraint-frames))
                                (expression (value (Cb = (1.0))))
                                (expr-type (value numeric))
                                (epsilon (value 0.02))
                                (value (value 0.0))
                                (status (value binding)))
         frame:)


(defprop constraint-Mn-flbl-action
         (constraint-Mn-flbl-action  (is-a (value constraint-frame))
                                     (definition (value yes))
                                     (data-item (value Mn-flbl))
                                     (forced-action-of
                                      (value constraint-LFB-AVERAGE-c3))
                                     (member-of (value constraint-frames))
                                     (expression
                                      (value
                                       (Mn-flbl -
                                                (0.9 *
                                                     Cb
                                                     •

; <«« start back on the left <««
     (Mp -
         (Mp - Mr)
         •
         (((bf // (2 * tf)) - (65 // sqrt (36.0)))
          //
          ((147 // sqrt (36.0 - Fr)) - (65 // sqrt (36.0)))))
; >»» continue on the right »»>
))))
                                     (expr-type (value numeric))
                                     (forced-actions
                                      (value constraint-Mr-action
                                             constraint-Mp-action
                                             constraint-Cb-action))
                                     (epsilon (value 256.1502855686401))
                                     (value (value 0.0) )
                                     (status (value binding)))
         frame:)


(defprop constraint-LFB-AVERAGE-c3
         (constraint-LFB-AVERAGE-c3  (is-a (value constraint-frame))
                                     (forced-constraint-of
                                      (value lrfd-LFB-average))
                                     (expression (value (2400.0 <= Mn-flbl)))
                                     (type (value performance))
                                     (expr-type (value numeric))
                                     (member-of (value constraint-frames))
                                     (forced-actions
                                      (value constraint-Mn-flbl-action))
                                     (epsilon (value 48.0))
                                     (value (value -10407.514278432))
                                     (status (value satisfied)))
         frame:)


(defprop constraint-Fr-action
         (constraint-Fr-action  (is-a (value constraint-frame))
                                (definition (value yes))
                                (data-item (value Fr))
                                (forced-action-of
                                 (value constraint-LFB-AVERAGE-c2))
                                (member-of (value constraint-frames))
                                (expression (value (Fr = (10.0))))
                                (expr-type (value numeric))
                                (epsilon (value 0.2))
                                (value (value 0.0))
                                (status (value binding)))
         frame:)


(defprop constraint-LFB-AVERAGE-c2
         (constraint-LFB-AVERAGE-c2  (is-a (value constraint-frame))
                                     (forced-constraint-of
                                      (value lrfd-LFB-average))
                                     (expression
                                      (value
                                       <(bf // (2 * tf))
                                        <=
                                        147
                                        //
                                        sqrt
                                         (36.0 - Fr))))
                                     (type (value applicability))
                                     (expr-type (value numeric))
                                     (member-of (value constraint-frames))
                                     (forced-actions
                                      (value constraint-Fr-action))
                                     (epsilon (value 0.07302593659942363))
                                     (value (value -25.17777503534187))
                                     (status (value satisfied)))
         frame:)


(defprop constraint-LFB-AVERAGE-cl
         (constraint-LFB-AVERAGE-cl  (is-a (value constraint-frame))
                                     (forced-constraint-of
                                      (value lrfd-LFB-average))
                                     (expression
                                      (value
                                       ((bf // (2 * tf)) > 65 // sqrt
   (36.0))))
                                     (type (value applicability))
                                     (expr-type (value numeric))
                                     (member-of (value constraint-frames))
                                     (epsilon (value 0.07302593659942363))
                                     (value (value 7.182036503362152))
                                     (status (value violated)))
         frame:)

I |4 <Enter> do-all-actions ((debug-output2))

Data Item: d * 14.38
Data Item: tw = 1.06
Data Item: Zx = 311.0
Data Item: bf = 12.67
Data Item: tf - 1.735
Data Item: ct = 7.19
Data Item: h - 10.91
Data Item: Af = 21.98245
Data Item: Ix = 993.4359574619792
Data Item: Sxt - 138.1691178667565
Data Item: Mr - 3592.397064535669
Data Item: Mp - 11196.0
Data Item: Cb = 1.0
Data Item: Mn-flbl = 12807.514278432
Data Item: Fr = 10.0| | 5 <Enter> do-all-actions nil
I | 5 <EXIT> do-all-actions t
I |4 <EXIT> do-all-actions t
j 3 <EXIT> do-all-actions t
|2 <EXIT> do-all-actions t
1 <EXIT> do-all-actions t
1 <Enter> do-all-actions ((read-message 'SET_SOLUTION_SUCCESSFUL) (post-
   message 'FINAL_SOLUTION_FOUND))
|2 <Enter> do-all-actions ((post-message 'FINAL_SOLUTION_FOUND))


The following message was just posted : FINAL_SOLUTION_FOUND

| 3 <Enter> do-all-actions nil
I 3 <EXIT> do-all-actions t
```

```
 |2 <EXIT>  do-all-actions  t
1 <EXIT>  do-all-actions  t
1 <Enter> do-all-actions ((read-message 'FINAL_SOLUTION_FOUND) (terpri)
   (terpri) (princ '|Task was successful.|) (terpri) ...)
 |2 <Enter> do-all-actions ((terpri) (terpri) (princ '|Task was successful.|)
   (terpri) (terpri))

 |  3 <Enter> do-all-actions ((terpri) (princ '|Task was successful.|)
   (terpri) (terpri))

 |  |4 <Enter> do-all-actions ((princ '|Task was successful.|) (terpri)
   (terpri))
Task was successful.|  | 5 <Enter> do-all-actions ((terpri) (terpri))

 |  |  |6 <Enter> do-all-actions ( (terpri))

 |  |  |  7 <Enter> do-all-actions nil
 I  I  I  7 <EXIT>  do-all-actions  t
 |  |  I 6 <EXIT>  do-all-actions  t
 I  |  5 <EXIT>  do-all-actions  t
 |  |4 <EXIT>  do-all-actions  t
 |  3 <EXIT>  do-all-actions  t
 |2 <EXIT>  do-all-actions  t
1 <EXIT>  do-all-actions  t


       _____SPEX Post Processor ____

SPEX> quit

_____Goodbye ____
```

6.2.2. AISC KBDBI Trace

The first listing shows the output of the AISC KBDBI as it processes the query from the SPEX KBSI. The second listing shows the INGRES output in response to the translated query.

AISC KBDBI Output

```
16.KAD-WAIT-F0R-MESSAGE:  The system name is ais
[load /usr/hch/kad/db/aisc/message/in/spx519005610]
KAD-LOAD-MESSAGE:  The message name is spx519005610
QUEL-RETRIEVE:  the retrieve string is
range of x is w_shapes
retrieve into kbdbitempO (x.zx,x.depth,x.tw,x.bf,x.tf,x.area,x.designation)
where 2400.000102 <= 0.900000 * 36.000002 * x.zx and 120.000003 <« 300 *
   sqrt((2 * x.tf / 12 * x.bf ** 3 + (x.depth - 2 * x.tf) / 12 * x.tw ** 3)
   / ((x.depth - 2 * x.tf) * x.tw + 2 * x.bf * x.tf)) / sqrt(36.000002)
range of tempo is kbdbitempO
retrieve into kbdbitempl (tempo.all)
where tempO.area = min(tempO.area)
print kbdbitempl
destroy kbdbitempO
destroy kbdbitempl
```

```
KAD-MESSAGE-NAME:  The message name is ais519006238
KAD-LFBS-GS-REPLY:  Translating reply.
KAD-SEND-MESSAGE:  Starting to send message.
KAD-MESSAGE-NAME:  The message name is ais519006243
KAD-SEND-FILE: sending file  to directory  on address-machine .
KAD-SEND-FILE: sending file  to directory  on address-machine .
KAD-SEND-MESSAGE:  Message sent.
KAD-DELETE-MESSAGE:  Deleting message spx519005610  for system nil
nil
17.Bye
```

INGRES Output

```
* * * * * * * * * Executing . . .

kbdbitempl relation

|zx        |depth     |tw        |bf        |tf        |area      |designation |
|------------------------------------------------------------------------------|
|   77.900|   12.060|    0.345|    9.995|    0.575|   15.600|W12x53       |
|------------------------------------------------------------------------------|
```

Chapter 7
HICOST-II Supplement

This chapter is intended to supplement the description of the HICOST-II implementation found in Section 5.2 of the thesis. The supplement contains the following information for HICOST-II, the building design database (BDD), the project management database (PMD), and the unit cost database (UCD).

- KADBASE schema descriptions for the HICOST-II KBS context, the HICOST global schema, and the three databases (including the LFBS, LFBM, and LIM for each component) as well as the GS, GDSM, and GIM.
- The data in the databases before and after the execution of HICOST-II.
- The component takeoff rules for three HICOST-II component types: concrete footings, retaining walls, and structural metal.

7.1. Schemata

The following five sections describe the schemata for HICOST-II, its three databases, and the global schema. The descriptions are expressed in the problem-oriented language of the schema description utility (described in Section 4.7). Each of the four component description sections contains the local frame-based schema (LFBS), the local frame-based mapping (LFBM), and the local integration mapping (LIM) grouped into a single set of definitions. The final section presents the global schema (GS), the global data source mapping (GDSM), and the global integration mapping (GIM) in the same fashion.

7.1.1. HICOST-II Context Schemata

```
lfbs hie

entity building
lfbm building
  slot instance-name
    slot-type attribute
    data-type symbol
    lfbm (building . instance-name)
  slot building-designation
    slot-type attribute
    data-type string
    lfbm (building . building-designation)
candidate-key instance-name
candidate-key building-designation
primary-key building-designation

entity construction-quantity
lfbm construction-quantity
```

```
  slot instance-name
    slot-type attribute
    data-type symbol
    lfbm (construction-quantity . instance-name)
  slot quantity-id
    slot-type attribute
    data-type symbol
    lfbm (construction-quantity . quantity-id)
  slot quantity-for
    slot-type relationship
    data-type symbol
    lfbm (construction-quantity . quantity-for)
    lim component-id
    relationship-type part-of
  slot units
    slot-type attribute
    data-type symbol
    lfbm (construction-quantity . units)
  slot amount
    slot-type attribute
    data-type number
    lfbm (construction-quantity . amount)
  slot unit-cost
    slot-type attribute
    data-type number
    lfbm (construction-quantity . unit-cost)
  slot cost
    slot-type attribute
    data-type number
    lfbm (construction-quantity . cost)
candidate-key instance-name
candidate-key (quantity-for quantity-id)
primary-key (quantity-for quantity-id)

entity concrete-footing
lfbm concrete-footing
  slot instance-name
    slot-type attribute
    data-type symbol
    lfbm (concrete-footing . instance-name)
  slot component-id
    slot-type attribute
    data-type symbol
    lfbm (concrete-footing . component-id)
  slot quantities
    slot-type relationship
    data-type symbol
    lfbm (concrete-footing . quantities)
    relationship-type sub-part
  slot number
    slot-type attribute
    data-type number
    lfbm (concrete-footing . number)
  slot length
    slot-type attribute
    data-type number
    lfbm (concrete-footing . length)
    units feet
  slot width
    slot-type attribute
    data-type number
    lfbm (concrete-footing . width)
    units feet
  slot footing-thickness
    slot-type attribute
    data-type number
    lfbm (concrete-footing . footing-thickness)
    units feet
  slot height-of-stepped-footing
    slot-type attribute
    data-type number
    lfbm (concrete-footing . height-of-stepped-footing)
  units feet
slot grade-elevation
  slot-type attribute
  data-type number
  lfbm (concrete-footing . grade-elevation)
  units feet
slot top-elevation
  slot-type attribute
  data-type number
  lfbm (concrete-footing . top-elevation)
  units feet
slot wall-thickness
  slot-type attribute
  data-type number
  lfbm (concrete-footing . wall-thickness)
  units feet
slot rock-elevation
  slot-type attribute
  data-type number
  lfbm (concrete-footing . rock-elevation)
  units feet
slot long-bar-size
  slot-type attribute
  data-type number
  lfbm (concrete-footing . long-bar-size)
slot long-bar-spacing
  slot-type attribute
  data-type number
  lfbm (concrete-footing . long-bar-spacing)
  units inches
slot long-bar-pieces
  slot-type attribute
  data-type number
  lfbm (concrete-footing . long-bar-pieces)
slot long-bar-lbs-lnft
  slot-type attribute
  data-type number
  lfbm (concrete-footing . long-bar-lbs-lnft)
  units lbs/lnft
slot long-bar-lbs-ftg
  slot-type attribute
  data-type number
  lfbm (concrete-footing . long-bar-lbs-ftg)
  units lbs
slot short-bar-size
  slot-type attribute
  data-type number
  lfbm (concrete-footing . short-bar-size)
slot short-bar-spacing
  slot-type attribute
  data-type number
  lfbm (concrete-footing . short-bar-spacing)
  units inches
slot short-bar-pieces
  slot-type attribute
  data-type number
  lfbm (concrete-footing . short-bar-pieces)
slot quality-of-concrete
  slot-type attribute
  data-type symbol
  lfbm (concrete-footing . quality-of-concrete)
slot concrete-placing-method-select ion
  slot-type attribute
  data-type symbol
  lfbm (concrete-footing . concrete-placing-method-selection)
slot footing-forms-selection
  slot-type attribute
  data-type symbol
  lfbm (concrete-footing . foot ing-forms-select ion)
slot earth-excavation-select ion
  slot-type attribute
  data-type symbol
```

```
    lfbm (concrete-footing . earth-excavation-selection)
  slot rock-excavation-selection
    slot-type attribute
    data-type symbol
    lfbm (concrete-footing . rock-excavation-selection)
  slot trench-bracing-selection
    slot-type attribute
    data-type symbol
    lfbm (concrete-footing . trench-bracing-selection)
  slot backfill-selection
    slot-type attribute
    data-type symbol
    lfbm (concrete-footing . backfill-selection)
  slot footing-keyway-selection
    slot-type attribute
    data-type symbol
    lfbm (concrete-footing . footing-keyway-selection)
  slot reinforeing-steel-select ion
    slot-type attribute
    data-type symbol
    lfbm (concrete-footing . reinforcing-steel-selection)
  slot anchor-boIts-and-base-plates-selection
    slot-type attribute
    data-type symbol
    lfbm (concrete-footing . anchor-boIts-and-base-plates-select ion)
candidate-key instance-name
candidate-key component-id
primary-key component-id

entity retaining-wall
lfbm retaining-wali
  slot instance-name
    slot-type attribute
    data-type symbol
    lfbm (retaining-wall . instance-name)
  slot component-id
    slot-type attribute
    data-type symbol
    lfbm (retaining-wall . compenent-id)
  slot quantities
    slot-type relationship
    data-type symbol
    lfbm (retaining-wall . quantities)
    relationship-type sub-part
  slot number
    slot-type attribute
    data-type number
    lfbm (retaining-wall . number)
  slot length
    slot-type attribute
    data-type number
    lfbm (retaining-wall . length)
    units feet
  slot height
    slot-type attribute
    data-type number
    lfbm (retaining-wall . height)
    units feet
  slot top-thickness
    slot-type attribute
    data-type number
    lfbm (retaining-wall . top-thickness)
    units feet
  slot base-thickness
    slot-type attribute
    data-type number
    lfbm (retaining-wall . base-thickness)
    units feet
  slot width
    slot-type attribute
    data-type number
    lfbm (retaining-wall . width)
    units feet
  slot footing-thickness
    slot-type attribute
    data-type number
    lfbm (retaining-wall . footing-thickness)
    units feet
  slot stepped-footing-height
    slot-type attribute
    data-type number
    lfbm (retaining-wall . stepped-footing-height)
    units feet
  slot footing-toe-width
    slot-type attribute
    data-type number
    lfbm (retaining-wall . footing-toe-width)
    units feet
  slot footing-depth
    slot-type attribute
    data-type number
    lfbm (retaining-wall . footing-depth)
    units feet
  slot reinforcing-steel
    slot-type attribute
    data-type number
    lfbm (retaining-wall . reinforcing-steel)
    units lbs/cuyd
  slot quality-of-concrete
    slot-type attribute
    data-type symbol
    lfbm (retaining-wall . quality-of-concrete)
  slot concrete-placement-selection
    slot-type attribute
    data-type symbol
    lfbm (retaining-wall . concrete-placement-selection)
  slot footing-form-and-finish-select ion
    slot-type attribute
    data-type symbol
    lfbm (retaining-wall . footing-form-and-finish-selection)
  slot wall-forms-selection
    slot-type attribute
    data-type symbol
    lfbm (retaining-wall . wall-forms-selection)
  slot excavation-selection
    slot-type attribute
    data-type symbol
    lfbm (retaining-wall . excavation-selection)
  slot trench-backfill-selection
    slot-type attribute
    data-type symbol
    lfbm (retaining-wall . trench-backfill-selection)
  slot retained-backfill-selection
    slot-type attribute
    data-type symbol
    lfbm (retaining-wall . retained-backfill-select ion)
  slot reinforcing-steel-selection
    slot-type attribute
    data-type symbol
    lfbm (retaining-wall . reinforcing-steel-selection)
  slot footing-key-and-constr-joints-selection
    slot-type attribute
    data-type symbol
    lfbm (retaining-wall . footing-key-and-constr-joints-selection)
  slot miscellaneous-selection
    slot-type attribute
    data-type symbol
    lfbm (retaining-wall . miscellaneous-selection)
candidate-key instance-name
candidate-key component-id
primary-key component-id

entity structural-metal
lfbm structural-metal
```

```
slot instance-name
  slot-type attribute
  data-type symbol
  Ifbm (structural-metal . instance-name)
slot component-id
  slot-type attribute
  data-type symbol
  Ifbm (structural-metal . component-id)
slot quantities
  slot-type relationship
  data-type symbol
  Ifbm (structural-metal . quantities)
  relationship-type sub-part
slot number
  slot-type attribute
  data-type number
  Ifbm (structural-metal . number)
slot pieces-per-location
  slot-type attribute
  data-type number
  Ifbm (structural-metal . pieces-per-location)
slot first-dimension
  slot-type attribute
  data-type number
  Ifbm (structural-metal . first-dimension)
  units inches
slot second-dimension
  slot-type attribute
  data-type number
  Ifbm (structural-metal . second-dimension)
  units inches
slot thickness
  slot-type attribute
  data-type number
  Ifbm (structural-metal . thickness)
  units inches
slot length
  slot-type attribute
  data-type number
  Ifbm (structural-metal . length)
  units feet
slot weight
  slot-type attribute
  data-type number
  Ifbm (structural-metal . weight)
  units lbs/lnft
slot fireproofing-thickness
  slot-type attribute
  data-type number
  Ifbm (structural-metal . fireproofing-thickness)
  units inches
slot anchor-bolt-diameter
  slot-type attribute
  data-type number
  Ifbm (structural-metal . anchor-bolt-diameter)
  units inches
slot anchor-bolt-length
  slot-type attribute
  data-type number
  Ifbm (structural-metal . anchor-bolt-length)
  units feet
slot type-of-metal-selection
  slot-type attribute
  data-type number
  Ifbm (structural-metal . type-of-metal-selection)
slot type-of-member-selection
  slot-type attribute
  data-type symbol
  Ifbm (structural-metal . type-of-member-selection)
slot erection-method-selection
  slot-type attribute
  data-type symbol
```

```
    lfbm (structural-metal . erection-method-selection)
  slot surface-treatment-selection
    slot-type attribute
    data-type symbol
    lfbm (structural-metal . surface-treatment-selection)
  slot spray-on-fireproofing-selection
    slot-type attribute
    data-type symbol
    lfbm (structural-metal . spray-on-fireproofing-selection)
  slot anchor-bolts-selection
    slot-type attribute
    data-type symbol
    lfbm (structural-metal . anchor-bolts-selection)
  slot element-number-selection
    slot-type attribute
    data-type symbol
    lfbm (structural-metal . element-number-selection)
candidate-key instance-name
candidate-key component-id
primary-key component-id

entity construction-quantity
  slot quantity-for
    range concrete-footing
    range retaining-wall
    range structural-metal
entity concrete-footing
  slot quantities
    range construction-quantity
entity retaining-wall
  slot quantities
    range construction-quantity
entity structural-metal
  slot quantities
    range construction-quantity

;
```

7.1.2. Building Design Database Schemata

```
lfbs bdd

entity building
  slot designation
    slot-type attribute
    data-type string20
    lim building-designation
candidate-key designation
primary-key designation

entity concrete-footing
lfbm cf_att
lfbm cf_spec
  slot ID
    slot-type attribute
    data-type integer
    lfbm (cf_att . ID)
    lfbm (cf_spec . ID)
    lim component-id
  slot A
    slot-type attribute
    data-type integer
    lfbm (cf_att . A)
    lim number
  slot B
    slot-type attribute
  data-type real
  lfbm (cf_att . B)
  lim length
  units feet
slot C
  slot-type attribute
  data-type real
  lfbm (cf_att . C)
  lim width
  units feet
slot Dl
  slot-type attribute
  data-type real
  lfbm (cf_att . D)
  lim footing-thickness
  units feet
slot D2
  slot-type attribute
  data-type real
  lfbm (cf_att . D)
  lim height-of-stepped-footing
  units feet
slot E
  slot-type attribute
  data-type real
  lfbm (cf_att . E)
  lim grade-elevation
  units feet
slot F
  slot-type attribute
  data-type real
  lfbm (cf_att . F)
  lim top-elevation
  units feet
slot G
  slot-type attribute
  data-type real
  lfbm (cf_att . G)
  lim wall-thickness
  units feet
slot H
  slot-type attribute
  data-type real
  lfbm (cf_att . H)
  lim rock-elevation
  units feet
slot Jl
  slot-type attribute
  data-type real
  lfbm (cf_att . J)
  lim long-bar-pieces
  mapped-type integer
  direct-function (lambda (x) (fix x))
slot J2
  slot-type attribute
  data-type real
  lfbm (cf_att . J)
  lim long-bar-spacing
  mapped-type integer
  direct-function (lambda (x) (fix x))
  units inches
slot J3
  slot-type attribute
  data-type real
  lfbm (cf_att . J)
  lim long-bar-size
  mapped-type integer
  direct-function
    (lambda (x) (fix (add 0.1 (product (diff x (fix x) ) 100))))
slot J4
  slot-type attribute
  data-type real
  lfbm (cf_att . J)
  lim long-bar-lbs-lnft
  units ibs/lnft
slot J5
  slot-type attribute
  data-type real
  lfbm (cf_att . J)
  lim long-bar-lbs-ftg
  units lbs
slot Kl
  slot-type attribute
  data-type real
  lfbm (cf_att . K)
  lim short-bar-pieces
  mapped-type integer
  direct-function (lambda (x) (fix x) )
slot K2
  slot-type attribute
  data-type real
  lfbm (cf_att . K)
  lim short-bar-spacing
  units inches
  mapped-type integer
  direct-function (lambda (x) (fix x))
slot K3
  slot-type attribute
  data-type real
  lfbm (cf_att . K)
  lim short-bar-size
  mapped-type integer
  direct-function
    (lambda (x) (fix (add 0.1 (product (diff x (fix x) ) 100)))]
slot SS9
  slot-type attribute
  data-type integer
  lfbm (cf_spec . SS9)
  lim quality-of-concrete
  mapped-type string
  mapping-table
    ((0 . "3000 psi standard mix")
     (1 . "deduct for 2500 psi")
     (2 . "add for 3500 psi")
     (3 . "add for 3750 psi")
     (4 . "add for 4000 psi"))
slot SS8
  slot-type attribute
  data-type integer
  lfbm (cf_spec . SS8)
  lim concrete-placing-method-selection
  mapped-type string
  mapping-table
    ( (0 . "none required")
     (1 . "wall ftg pour concrete direct")
     (2 . "wall ftg pour concrete w/carts")
     (3 . "wall ftg pump concrete")
     (4 . "wall ftg pour concrete w/crane")
     (5 . "column ftg pour concrete direct")
     (6 . "column ftg pour concrete w/carts")
     (7 . "column ftg pump concrete")
     (8 . "column ftg pour concrete w/crane")
     (9 . "stepped wall ftg concrete"))
slot SSI
  slot-type attribute
  data-type integer
  lfbm (cf_spec . SS7)
  lim footing-forms-selection
  mapped-type string
  mapping-table
    ((0 . "none required")
     (1 . "form wall ftg earth")
     (2 . "form wall ftg rock")
     (3 . "form column ftg earth")
     (4 . "form column ftg rock")
     (5 . "stepped ftg forms")
     (6 . "pilaster wall ftg")
     (7 . "curved wall ftg")
     (8 . "pile cap forms")
     (9 . "wall ftg bulkhds."))
slot SS6
  slot-type attribute
  data-type integer
  lfbm (cf_spec . SS6)
  lim earth-excavation-selection
  mapped-type string
  mapping-table
    ((0 . "none required")
     (1 . "mach. to top of rock")
     (2 . "hand excavation only")
     (3 . "machine w/ hand dress-up"))
slot SS5
  slot-type attribute
  data-type integer
  lfbm (cf_spec . SS5)
  lim rock-excavation-selection
  mapped-type string
  mapping-table
    ((0 . "none required")
     (1 . "machine w/ hand dress-up")
     (2 . "hand excavation only")
     (3 . "blasting w/machine & hand dress-up"))
slot SS4
  slot-type attribute
  data-type integer
  lfbm (cf_spec . SS4)
  lim trench-bracing-selection
  mapped-type string
  mapping-table
    ( (0 . "none required")
     (1 . "jacks only wall ftg")
     (2 . "open plank wall ftg")
     (3 . "solid wood wall ftg")
     (4 . "solid wood col. ftg")
     (5 . "h-cols & wd plank col. ftg"))
slot SS3
  slot-type attribute
  data-type integer
  lfbm (cf_spec . SS3)
  lim backfill-selection
  mapped-type string
  mapping-table
    {(0 . "none required")
     (1 . "earth mach. w/ hand tamp")
     (2 . "earth borrow mach. w/ hand tamp")
     (3 . "earth hand only")
     (4 . "granular borrow mach. w/ hand tamp")
     (5 . "gravel/stone borrow mach. only"))
slot SS2
  slot-type attribute
  data-type integer
  lfbm (cf_spec . SS2)
  lim footing-keyway-selection
  mapped-type string
  mapping-table
    ((0 . "none required")
     (1 . "2x4 ftg key")
     (2 . "split key w/waterstop")
     (3 . "4x4 key or 2x8 key")
     (4 . "4x6 key or 3x8 key")
     (5 . "4x12 key"))
slot SSI
  slot-type attribute
  data-type integer
  lfbm (cf_spec . SSI)
  lim reinforcing-steel-selection
```

```
    mapped-type string
    mapping-table
      ((0 . "none required")
       (1 . "column footings lbs/cuyd calc")
       (2 . "wall footing 50 lfb/cuyd")
       (3 . "bars by pieces both ways")
       (4 . "bars by spacing both ways")
       (5 . "long bar by piece short bar by spacing")
       (6 . "long bar by spacing short bar by pieces")
       (7 . "lbs/lnft wall ftg with j dim")
       (8 . "lbs/footing column footing with j dim"))
  slot SSO
    slot-type attribute
    data-type integer
    lfbm (cf_spec . SSO)
    lim anchor-bolts-and-base-plates-selection
    mapped-type string
    mapping-table
      ({0 . "none required")
       (1 . "set 2ab")
       (2 . "set 4ab")
       (3 . "set 2ab grout base pit")
       (4 . "set 4ab grout base pit")
       (5 . "set 2ab set and grout base pit")
       (6 . "set 4ab set and grout base pit")
       (7 . "set & grout base plate"))
candidate-key ID
primary-key ID

entity retaining-wall
lfbm rw_att
lfbm rw_spec
  slot ID
    slot-type attribute
    data-type integer
    lfbm (rw_att . ID)
    lfbm (rw_spec . ID)
    lim component-id
  slot A
    slot-type attribute
    data-type integer
    lfbm (rw_att . A)
    lim number
  slot B
    slot-type attribute
    data-type real
    lfbm (rw_att . B)
    lim length
    units feet
  slot C
    slot-type attribute
    data-type real
    lfbm (rw_att . C)
    lim height
    units feet
  slot D
    slot-type attribute
    data-type real
    lfbm (rw_att . D)
    lim top-thickness
    units feet
  slot E
    slot-type attribute
    data-type real
    lfbm (rw_att . E)
    lim base-thickness
    units feet
  slot F
    slot-type attribute
    data-type real
    lfbm (rw_att . F)
    lim width
  units feet
slot Gl
  slot-type attribute
  data-type real
  lfbm (rw_att . G)
  lim footing-thickness
  units feet
slot G2
  slot-type attribute
  data-type real
  lfbm (rw_att . G)
  lim stepped-footing-height
  units feet
slot H
  slot-type attribute
  data-type real
  lfbm (rw_att . H)
  lim footing-toe-width
  units feet
slot J
  slot-type attribute
  data-type real
  lfbm (rw_att . J)
  lim footing-depth
  units feet
slot K
  slot-type attribute
  data-type real
  lfbm (rw_att . K)
  lim reinforcing-steel
  units lbs/cuyd
slot SS9
  slot-type attribute
  data-type integer
  lfbm (rw_spec . SS9)
  lim quality-of-concrete
  mapped-type string
  mapping-table
    ((0 . "3000 psi standard mix")
     (1 . "deduct for 2500 psi")
     (2 . "add for 3500 psi")
     (3 . "add for 4000 psi"))
slot SS8
  slot-type attribute
  data-type integer
  lfbm (rw_spec . SS8)
  lim concrete-placement-selection
  mapped-type string
  mapping-table
    ((0 . "none required")
     (1 . "ftg & wall direct")
     (2 . "ftg direct wall w/carts")
     (3 . "ftg direct wall - pump")
     (4 . "ftg direct wall w/crane")
     (5 . "ftg & wall w/carts")
     (6 . "ftg w/carts wall w/pump")
     (7 . "ftg w/carts wall w/crane")
     (8 . "ftg & wall w/pump")
     (9 . "stepped wall ftg direct"))
slot SS7
  slot-type attribute
  data-type integer
  lfbm (rw_spec . SS7)
  lim footing-form-and-finish-selection
  mapped-type string
  mapping-table
    ((0 . "none required")
     (1 . "in earth")
     (2 . "in rock")
     (3 . "stepped wall ftg forms")
     (4 . "ftg bulkhead"))
slot SS6
```

```
  slot-type attribute
  data-type integer
  lfbm (rw_spec . SS6)
  lim wall-forms-selection
  mapped-type string
  mapping-table
     ((0 . "none required")
      (1 . "both sides vertical")
      (2 . "one side battered")
      (3 . "add ledge")
      (4 . "add ledge & brick facing"))
slot SS5
  slot-type attribute
  data-type integer
  lfbm (rw_spec . SS5)
  lim excavation-selection
  mapped-type string
  mapping-table
    ((0 . "none required")
     (1 . "earth machine w/hand dress")
     (2 . "earth hand")
     (3 . "rock machine w/hand dress")
     (4 . "rock hand"))
slot SS4
  slot-type attribute
  data-type integer
  lfbm (rw_spec . SS4)
  lim trench-backfill-selection
  mapped-type string
  mapping-table
    ((0 . "none required")
     (1 . "earth adjacent mach placed hand tamp")
     (2 . "earth adjacent hand placed hand tamp")
     (3 . "earth hauled mach placed hand tamp")
     (4 . "earth hauled hand placed hand tamp"))
slot SS3
  slot-type attribute
  data-type integer
  lfbm (rw_spec . SS3)
  lim retained-backfill-selection
  mapped-type string
  mapping-table
    ((0 . "none required")
     (1 . "earth mach placed mach compact")
     (2 . "earth mach placed hand tamp")
     (3 . "earth borrow mach placed mach compact")
     (4 . "earth borrow mach placed hand tamp")
     (5 . "gravel mach placed")
     (6 . "bank run mach placed w/hand tamp"))
slot SS2
  slot-type attribute
  data-type integer
  lfbm (rw_spec . SS2)
  lim reinforcing-steel-selection
  mapped-type string
  mapping-table
    ((0 . "none required")
     (1 . "index crsi level bf- no surcharge")
     (2 . "index crsi sloped bf- no surcharge")
     (3 . "index crsi level bf- hwy surchg (light)")
     (4 . "index crsi level bf- railwy surch (heavy)")
     (5 . "Ibs/cuyd"))
slot SSI
  slot-type attribute
  data-type integer
  lfbm (rw_spec . SSI)
  lim footing-key-and-constr-joints-selection
  mapped-type string
  mapping-table
    ((0 . "none required")
     (1 . "2x4 key in ftg")
     (2 . "2x6 key in ftg")
```

```
        (3 . "2x8 key in ftg")
        (4 . "3x8 key in ftg11)
        (5 . "2x6 ftg key & one wall constr jt")
        (6 . "2x6 ftg key & wall constr jt  30 in. oc")
        (7 . "one wall constr jt")
        (8 . "wall constr jt 30 in. oc"))
  slot SSO
    slot-type attribute
    data-type integer
    lfbm (rw_spec . SSO)
    lim miscellaneous-selection
    mapped-type string
    mapping-table
      ((0 . "none required")
        (1 . "1 in. diam weep holes 36 in. o.c")
        (2 . "1-1/2 in. diam weep holes 36 in. o.c.11)
        (3 . "2 in. diam weep holes 36 in. o.c")
        (4 . "3 in. diam weep holes 36 in. o.c")
        (5 . "rub exposed wall sfc & finish top")
        (6 . "ftg waterstop ")
        (7 . "wall waterstop")
        (8 . "wall waterstop  30 in o.c"))
candidate-key ID
primary-key ID

entity structural-metal
lfbm sm_att
lfbm sm_spec
  slot ID
    slot-type attribute
    data-type integer
    lfbm (sm_att . ID)
    lfbm (sm_spec . ID)
    lim component-id
  slot A
    slot-type attribute
    data-type integer
    lfbm (sm_att . A)
    lim number
  slot B
    slot-type attribute
    data-type integer
    lfbm (sm_att . B)
    lim pieces-per-location
  slot C
    slot-type attribute
    data-type real
    lfbm (sm_att . C)
    lim first-dimension
    units inches
  slot D
    slot-type attribute
    data-type real
    lfbm (sm_att . D)
    lim second-dimension
    units inches
  slot E
    slot-type attribute
    data-type real
    lfbm (sm_att . E)
    lim thickness
    units inches
  slot F
    slot-type attribute
    data-type real
    lfbm (sm_att . F)
    lim length
    units feet
  slot G
    slot-type attribute
    data-type real
    lfbm (sm_att . G)
  lim weight
  units lbs/lnft
slot H
  slot-type attribute
  data-type real
  lfbm (sm_att . H)
  lim fireproofing-thickness
  units inches
slot J
  slot-type attribute
  data-type real
  lfbm (sm_att . J)
  lim anchor-bolt-diameter
  units inches
slot K
  slot-type attribute
  data-type real
  lfbm (sm_att . K)
  lim anchor-bolt-length
  units feet
slot SS9
  slot-type attribute
  data-type integer
  lfbm (sm_spec . SS9)
  lim type-of-metal-selection
  mapped-type string
  mapping-table
     ((0 . "A-36 std stl.")
      (1 . "add hi-strngth steel")
      (2 . "add specialty steel")
      (3 . "structural aluminum")
      (4 . "extruded aluminum"))
slot SS8
  slot-type attribute
  data-type integer
  lfbm (sm_spec . SS8)
  lim type-of-member-selection
  mapped-type string
  mapping-table
     ((0 . "none-required")
      (1 . "beam")
      (2 . "column")
      (3 . "channel")
      (4 . "angle")
      (5 . "plate")
      (6 . "tubular (rect or sq.)")
      (7 . "pipe column")
      (8 . "rectangular rod (solid)")
      (9 . "round rod"))
slot SS7
  slot-type attribute
  data-type integer
  lfbm (sm_spec . SSI)
  lim erection-method-selection
  mapped-type string
  mapping-table
     ((0 . "none required")
      (1 . "bolted connections steel")
      (2 . "welded connections steel")
      (3 . "aluminum structural")
      (4 . "aluminum tubular"))
slot SS6
  slot-type attribute
  data-type integer
  lfbm (sm_spec . SS6)
  lim surface-treatment-selection
  mapped-type string
  mapping-table
     ( (0 . "none required")
      (1 . "paint touch-up per ton")
      (2 . "field paint per ton")
      (3 . "field paint per sqft")
```

```
       (4 . "galvanize per ton"))
  slot SS5
    slot-type attribute
    data-type integer
    lfbm (sm_spec . SS5)
    lim spray-on-fireproofing-selection
    mapped-type string
    mapping-table
      ((0 . "none required")
       (1 . "cementitious")
       (2 . "fiber"))
  slot SS4
    slot-type attribute
    data-type integer
    lfbm (sm_spec . SS4)
    lim anchor-bolts-selection
    mapped-type string
    mapping-table
      ((0 . "none required")
       (1 . "furnish 2 a.b. per col.")
       (2 . "furnish 4 a.b. per col."))
  slot SSO
    slot-type attribute
    data-type integer
    lfbm (sm_spec . SSO)
    lim element-number-selection
    mapped-type string
    mapping-table
      ({0 . "use u.p.c. default")
       (1 . "0311 columns")
       (2 . "0321 beams")
       (3 . "0322 supp floor system")
       (4 . "0323 level roof syst.")
       (5 . "0324 pitched roof syst.")
       (6 . "0330 supp. stwy. constr.")
       (7 . "0411 exterior wall constr.11)
       (8 . "0200 sub structure")
       (9 . "no element required"))
candidate-key ID
primary-key ID

;
```

7.1.3. Project Management Database Schemata

```
lfbs pmd

entity quantity
lfbm quantity
lim construction-quantity
  slot qty_id
    slot-type attribute
    data-type integer
    lfbm (quantity . qty_id)
    lim quantity-id
  slot comp_id
    slot-type relationship
    data-type integer
    lfbm (quantity . comp_id)
    lim component-id
  slot units
    slot-type attribute
    data-type stringlO
    lfbm (quantity . units)
  slot amount
    slot-type attribute
    data-type real
    lfbm (quantity . amount)
  slot unit-cost
    slot-type attribute
    data-type real
    lfbm (quantity . unit_cost)
  slot cost
    slot-type attribute
    data-type real
    lfbm (quantity . cost)
candidate-key (qty_id comp_id)
primary-key (qty_id comp__id)

;
```

7.1.4. Unit Cost Database Schemata

```
lfbs ucd

entity unit_cost
lfbm unit_cost
lim unit-cost
  slot itemcode
    slot-type attribute
    data-type integer
    lfbm (unit_cost . itemcode)
    lim quantity-id
  slot itemdesc
    slot-type attribute
    data-type string50
    lfbm (unit_cost . itemdesc)
  slot itemunit
    slot-type attribute
    data-type string5
    lfbm (unit_cost . itemunit)
  slot matlcost
    slot-type attribute
    data-type real
    lfbm (unit_cost . matlcost)
    units dollars
  slot instcost
    slot-type attribute
    data-type real
    lfbm (unit_cost . instcost)
    units dollars
  slot datemcos
    slot-type attribute
    data-type integer
    lfbm (unit_cost . datemcos)
  slot dateicos
    slot-type attribute
    data-type integer
    lfbm (unit__cost . dateicos)
candidate-key itemcode
primary-key itemcode
;
```

7.1.5. Global Schemata

```
global-schema

entity unit-cost
sources (ucd)
  slot quantity-id
    slot-type attribute
    data-type integer
    sources (ucd)
  slot itemdesc
    slot-type attribute
    data-type string50
    sources (ucd)
  slot itemunit
    slot-type attribute
    data-type string5
    sources (ucd)
  slot unit-cost
    slot-type attribute
    data-type real
    units dollars
  slot matlcost
    slot-type attribute
    data-type real
    sources (ucd)
    units dollars
  slot instcost
    slot-type attribute
    data-type real
    sources (ucd)
    units dollars
  slot datemcos
    slot-type attribute
    data-type integer
    sources (ucd)
  slot dateicos
    slot-type attribute
    data-type integer
    sources (ucd)
candidate-key quantity-id
primary-key quantity-id

entity construction-quantity
sources (hie pmd)
  slot instance-name
    slot-type attribute
    data-type symbol
    sources (hie)
  slot quantity-id
    slot-type relationship
    data-type symbol
    sources (hie pmd)
    relationship-type is-a
  slot component-id
    slot-type relationship
    data-type symbol
    sources (hie pmd)
    relationship-type part-of
  slot units
    slot-type attribute
    data-type symbol
    sources (hie pmd)
  slot amount
    slot-type attribute
    data-type number
    sources (hie pmd)
  slot unit-cost
    slot-type attribute
    data-type number
    sources (hie)
  slot cost
    slot-type attribute
    data-type number
    sources (hie pmd)
candidate-key instance-name
candidate-key (component-id quantity-id)
primary-key (component-id quantity-id)

entity concrete-footing
sources (hie bdd)
  slot instance-name
    slot-type attribute
    data-type symbol
    sources (hie)
  slot component-id
    slot-type attribute
    data-type symbol
    sources (hie bdd)
  slot quantities
    slot-type relationship
    data-type symbol
    sources (hie)
    relationship-type subpart
```

```
slot number
  slot-type attribute
  data-type number
  sources (hie bdd)
slot length
  slot-type attribute
  data-type number
  sources (hie bdd)
  units feet
slot width
  slot-type attribute
  data-type number
  sources (hie bdd)
  units feet
slot footing-thickness
  slot-type attribute
  data-type number
  sources (hie bdd)
  units feet
slot height-of-stepped-footing
  slot-type attribute
  data-type number
  sources (hie bdd)
  units feet
slot grade-elevation
  slot-type attribute
  data-type number
  sources (hie bdd)
  units feet
slot top-elevation
  slot-type attribute
  data-type number
  sources (hie bdd)
  units feet
slot wall-thickness
  slot-type attribute
  data-type number
  sources (hie bdd)
  units feet
slot rock-elevation
  slot-type attribute
  data-type number
  sources (hie bdd)
  units feet
slot long-bar-size
  slot-type attribute
  data-type number
  sources (hie bdd)
slot long-bar-spacing
  slot-type attribute
  data-type number
  sources (hie bdd)
  units inches
slot long-bar-pieces
  slot-type attribute
  data-type number
  sources (hie bdd)
slot long-bar-lbs-lnft
  slot-type attribute
  data-type number
  sources (hie bdd)
  units lbs/lnft
slot long-bar-lbs-ftg
  slot-type attribute
  data-type number
  sources (hie bdd)
  units lbs
slot short-bar-size
  slot-type attribute
  data-type number
  sources (hie bdd)
slot short-bar-spacing
```

```
    slot-type attribute
    data-type number
    sources (hie bdd)
    units inches
  slot short-bar-pieces
    slot-type attribute
    data-type number
    sources (hie bdd)
  slot quality-of-concrete
    slot-type attribute
    data-type symbol
    sources (hie bdd)
  slot concrete-placing-method-select ion
    slot-type attribute
    data-type symbol
    sources (hie bdd)
  slot footing-forms-selection
    slot-type attribute
    data-type symbol
    sources (hie bdd)
  slot earth-excavation-selection
    slot-type attribute
    data-type symbol
    sources (hie bdd)
  slot rock-excavation-selection
    slot-type attribute
    data-type symbol
    sources (hie bdd)
  slot trench-bracing-selection
    slot-type attribute
    data-type symbol
    sources (hie bdd)
  slot backfill-selection
    slot-type attribute
    data-type symbol
    sources (hie bdd)
  slot footing-keyway-selection
    slot-type attribute
    data-type symbol
    sources (hie bdd)
  slot reinforcing-steel-selection
    slot-type attribute
    data-type symbol
    sources (hie bdd)
  slot anchor-boIts-and-base-plates-selection
    slot-type attribute
    data-type symbol
    sources (hie bdd)
candidate-key instance-name
candidate-key component-id
primary-key component-id

entity retaining-wall
sources (hie bdd)
  slot instance-name
    slot-type attribute
    data-type symbol
    sources (hie)
  slot component-id
    slot-type attribute
    data-type symbol
    sources (hie bdd)
  slot quantities
    slot-type relationship
    data-type symbol
    sources (hie)
    relationship-type subpart
  slot number
    slot-type attribute
    data-type number
    sources (hie bdd)
  slot length
```

```
    slot-type attribute
    data-type number
    sources (hie bdd)
    units feet
  slot height
    slot-type attribute
    data-type number
    sources (hie bdd)
    units feet
  slot top-thickness
    slot-type attribute
    data-type number
    sources (hie bdd)
    units feet
  slot base-thickness
    slot-type attribute
    data-type number
    sources (hie bdd)
    units feet
  slot width
    slot-type attribute
    data-type number
    sources (hie bdd)
    units feet
  slot footing-thickness
    slot-type attribute
    data-type number
    sources (hie bdd)
    units feet
  slot stepped-footing-height
    slot-type attribute
    data-type number
    sources (hie bdd)
    units feet
  slot footing-toe-width
    slot-type attribute
    data-type number
    sources (hie bdd)
    units feet
  slot footing-depth
    slot-type attribute
    data-type number
    sources (hie bdd)
    units feet
  slot reinforcing-steel
    slot-type attribute
    data-type number
    sources (hie bdd)
    units lbs/cuyd
  slot quality-of-concrete
    slot-type attribute
    data-type symbol
    sources (hie bdd)
  slot concrete-placement-selection
    slot-type attribute
    data-type symbol
    sources (hie bdd)
  slot foot ing-form-and-finish-selection
    slot-type attribute
    data-type symbol
    sources (hie bdd)
  slot wall-forms-selection
    slot-type attribute
    data-type symbol
    sources (hie bdd)
  slot excavation-selection
    slot-type attribute
    data-type symbol
    sources (hie bdd)
  slot trench-backfill-selection
    slot-type attribute
    data-type symbol
    sources (hie bdd)
  slot retained-backfill-selection
    slot-type attribute
    data-type symbol
    sources (hie bdd)
  slot reinforcing-steel-selection
    slot-type attribute
    data-type symbol
    sources (hie bdd)
  slot footing-key-and-constr-joints-selection
    slot-type attribute
    data-type symbol
    sources (hie bdd)
  slot miscellaneous-selection
    slot-type attribute
    data-type symbol
    sources (hie bdd)
candidate-key instance-name
candidate-key component-id
primary-key component-id

entity structural-metal
sources (hie bdd)
  slot instance-name
    slot-type attribute
    data-type symbol
    sources (hie)
  slot component-id
    slot-type attribute
    data-type symbol
    sources (hie bdd)
  slot quantities
    slot-type relationship
    data-type symbol
    sources (hie)
    relationship-type subpart
  slot number
    slot-type attribute
    data-type number
    sources (hie bdd)
  slot pieces-per-location
    slot-type attribute
    data-type number
    sources (hie bdd)
  slot first-dimension
    slot-type attribute
    data-type number
    sources (hie bdd)
    units inches
  slot second-dimension
    slot-type attribute
    data-type number
    sources (hie bdd)
    units inches
  slot thickness
    slot-type attribute
    data-type number
    sources (hie bdd)
    units inches
  slot length
    slot-type attribute
    data-type number
    sources (hie bdd)
    units feet
  slot weight
    slot-type attribute
    data-type number
    sources (hie bdd)
    units lbs/lnft
  slot fireproofing-thickness
    slot-type attribute
    data-type number
    sources (hie bdd)
    units inches
  slot anchor-bolt-diameter
    slot-type attribute
    data-type number
    sources (hie bdd)
    units inches
  slot anchor-bolt-length
    slot-type attribute
    data-type number
    sources (hie bdd)
    units feet
  slot type-of-metal-selection
    slot-type attribute
    data-type number
    sources (hie bdd)
  slot type-of-member-selection
    slot-type attribute
    data-type symbol
    sources (hie bdd)
  slot erection-method-selection
    slot-type attribute
    data-type symbol
    sources (hie bdd)
  slot surface-treatment-selection
    slot-type attribute
    data-type symbol
    sources (hie bdd)
  slot spray-on-fireproofing-selection
    slot-type attribute
    data-type symbol
    sources (hie bdd)
  slot anchor-bolts-selection
    slot-type attribute
    data-type symbol
    sources (hie bdd)
  slot element-number-selection
    slot-type attribute
    data-type symbol
    sources (hie bdd)
candidate-key instance-name
candidate-key component-id
primary-key component-id

entity construction-quantity
  slot quantity-for
    range concrete-footing
    range retaining-wall
    range structural-metal
  slot quantity-id
    range unit-cost

entity unit-cost
  slot unit-cost
    constraint  (unit-cost.unit-cost == unit-cost.matIcost
                                        + unit-cost.instcost)
;
```

7.2. Data

The following sections show the data in the relations of the building design (BDD), unit cost (UCD), and project management (PMD) databases. The data in the BDD and UCD does not change during HICOST-II processing; therefore, only one version of the data in those databases is shown here. The PMD is initially empty and is updated by HICOST; therefore, two version are included here: after the end of the takeoff process and after the end of the pricing process.

7.2.1. Building Design Database Data

The following six relations represent the three different types of objects contained in the building design database. The relations and attributes are defined in Section 5.2.2.3 of the thesis.

cf_att relation *{describes the attributes ofconcrete footings)*

```
lid  |a   |b         |c         Id         |e         |f         |g         |h         Ii         Ij         |k         I
I  111   1|  150.000 |   3.0001   1.0001  98.6701  94.3301   1.0001   0.0001   0.000!   4.050 |  48.0301
I  12|   6|    5.0001   1.5001   1.0001  98.6701  94.3301   0.0001   0.0001   0.0001   3.0501   5.0501
I  13|   4|    0.0001   3.0001   1.3301   0.0001   0.0001   0.0001   0.0001   0.0001   0.000!   0.0001
I  14|  12|    8.0001   7.0001   1.5001  98.5001  97.5001   0.0001  96.5001   0.0001   8.0601   8.0601
```

cf__spec relation *{describes the construction method for concrete footings)*

```
lid     |ss0    |ssl    |ss2    |ss3    |ss4    |ss5    |ss6    |ss7    |ss8    |ss9    |
I     H I      0|      5|      1|      1|      0|      0|      31      1|      II      0|
I     12 I      0|      3|      0|      1|      0|      0|      31      6|      II      0|
I     13 I      0|      0|      0|      0|      0|      0|      01      5|      9|      0|
I     14 |      5|      4|      0|      4|      0|      1|      1|      4|      7|      2 I
```

rw__att relation *{describes the attributes ofretaining walls)*

```
i i d | a | b         |c         Id         |e         |f         |g         |h         |i         |j         |k         |
I  31|   1|   30.0001   8.6701   0.670 j   1.0001   7.0001   1.2501   3.5001   0.0001   3.0001   0.0001
I  32|   1|    0.0001   0.0001   0.0001   0.0001   7.0001   3.3301   0.0001   0.0001   0.0001   0.0001
I  33|   11154.0001  12.0001   1.0001   1.000|   6.5001   1.5001   3.0001   0.0001   2.3301  70.0001
I  34|   1|   82.0001   6.0001   0.3301   0.0001   0.0001   0.0001   0.0001   0.0001   0.0001   0.0001
```

rw_spec relation *{describes the construction method for retaining walls)*

```
lid     |ss0    |ssl    |ss2    |ss3    |ss4    |ss5    |ss6    |ss7    |ss8    |ss9    |
I     31 |      5|      1|      1|      2|      1|      1|      21      1|      1|      0|
I     32 I      0|      0|      0|      0|      0|      0|      0i      3|      9|      0|
I     33 |      3!      6|      1!      5|      3|      3|      1|      2|      3!      0|
I     34 |      0|      0|      0|      0|      0|      0|      3|      0|      31      0|
```

sm__att relation *{describes the attributes ofstructural metal)*

id	a	b	c	d	e	f	g	h	i	j	k
211	21	5.0001	12.000	12.3001	0.0001	20.000	1120.0001	1.5001	0.000i	0.000	0.000i
221	41	10.0001	8.000	8.000	0.0001	12.500	1 48.0001	2.0001	0.000i	1.250	1.7501
23	10	2.0001	6.000	4.000	0.3701	15.000	1 0.0001	0.0001	0.000i	0.000i	0.000
24	21	4.0001	12.0001	8.0001	0.5001	12.000	1 0.0001	0.0001	0.000	0.7501	1.5001

sm_spec relation *{describes the construction methodfor structural metal)*

lid	iss0	ss4	ss5	ss6	ss7	ss8	ss9
211	2	01	11	11	11	11	01
221	11	11	2	01	11	21	11
23	91	01	01	31	11	41	21
24	7	21	01	01	41	61	41

7.2.2. Unit Cost Database Data

The following relation contains the unit cost information. The relations and attributes are defined in Section 5.2.2.3 of the thesis. The fields DATEMCOS and DATEICOS (the dates for the material and installation costs) are omitted from this display because values for those fields were never stored in the database. The relation contains 100 tuples. The parenthesized number in the ITEMDESC field for each tuple references the item number in *Means Building Construction Cost Data 1986* [Means 86] corresponding the MATLCOST and INSTCOST values.

unit_cost relation

itemcode	itemdesc	itemunit	matlcost	instcost
33100	13000 psi ready-mix concrete (03.3-12-015)	CUYD	48.900	0.0001
33101	12500 psi ready-mix concrete (03.3-12-010)	CUYD	47.250	0.000i
33102	13500 psi ready-mix concrete (03.3-12-020)	CUYD	50.550	0.000
33103	13750 psi ready-mix concrete (03.3-12-025)	CUYD	51.500	0.000i
33104	14000 psi ready-mix concrete (03.3-12-030)	CUYD	52.200	0.000i
33120	pour (wall) concrete ftg - direct (03.3-38-190)	CUYD	0.450	6.7001
33130	pour (wall) concrete ftg - carts (03.3-38-562)	CUYD	1.150	6.050
33151	pour (wall) concrete ftg - pump (03.3-38-195)	CUYD	5.650	11.000
33140	pour (wall)concrete ftg - crane (03.3-38-200)	CUYD	6.650	12.2001
33165	pour (column) concrete ftg - direct (03.3-38-190)	CUYD	0.450	6.7001
33166	pour (column) concrete ftg - carts (03.3-38-562)	CUYD	1.150	6.0501
33167	pour (column) concrete ftg - pump (03.3-38-195)	CUYD	5.650	11.0001
33168	pour (column) concrete ftg - crane (03.3-38-200)	CUYD	6.650	12.200
30136	finish top of footing (03.3-26-001 floor finish)	SQFT	0.000	0.170
31006	wall ftg. forms - earth (03.1-45-005 2 use)	SQFT	0.430	1.4401
31007	column ftg. forms - earth (03.1-45-005 2 use)	SQFT	0.430	1.4401
31008	wall ftg. bulkhead form (03.1-55-100 1 use)	SQFT	0.340	1.2401
31009	Add pilaster wall ftg form (03.1-65-862 2 use)	SQFT	0.910	2.9601
31010	Curved wall ftg form (03.1-65-405 2 use)	SQFT	0.920	3.2601
31013	wall ftg. forms - rock (03.1-45-015 2 use)	SQFT	0.430	1.4401
31014	column ftg. forms - rock (03.1-45-015 2 use)	SQFT	0.430	1.440
31016	pile cap forms (03.1-45-305 2 use)	SQFT	1.240	2.180
222033	Mach ftg. earth excavation (02.3-18-004 3')	CUYD	1.170	1.460
222034	Mach ftg. earth excavation (02.3-18-004 3-4')	CUYD	1.170	1.460
222035	Mach ftg. earth excavation (02.3-18-004 4-5')	CUYD	1.170	1.460
222036	Mach ftg. earth excavation (02.3-18-004 5-6')	CUYD	1.170	1.460
22205	Hand ftg. earth excavation (02.3-18-140)	CUYD	0.000	15.900
22207	Hand ftg. rock excavation (02.3-18-150)	CUYD	0.000	32.000
22211	Blasting rock ftg. excavation (02.3-15-220)	CUYD	1.050	39.000
24100	Open plank bracing earth (02.3-38-390)	SQFT	0.800	2.470

```
|   21401|Solid wood bracing earth (02.3-38-390)              |SQFT   |   0.800!   2.4701
|   24102IH-columns & plank bracing - earth (02.3-40-010)    |SQFT   |   4.9001   5.8501
|   22204lOn-site earth-machine (02.3-03-130)                |CUYD   |   0.5301   0.190|
|  222041|add hand tamp (02.3-18-170 trench)                 |CUYD   |   0.370|   1.2701
|   22206|Earth Bf-Hand only (02.3-03-001)                   |CUYD   |   0.0001   9.1001
|   22217|Earth borrow BF-machine (02.3-05-020)              |CUYD   |   5.0001   2.810|
|   22219IGranular Borrow BF-machine (02.3-05-060 sand)      |CUYD   |   8.0001   2.8101
|   22220IGravel/stone borrow BF-machine only (02.3-05-034)  ICUYD  |  14.0001   2.810|
|   31011|Footing keyway 2x4 (03.1-45-150)                   ILNFT  |   0.0701   0.3001
|   31012IWaterstop (03.1-90-001 4 inch pvc)                 ILNFT  |   0.8501   1.0301
|   31019|Add. matl. Footing keyway 2x4 (03.1-45-150)        ILNFT  |   0.0701   0.3001
|   32101 I re-steel footings (03.2-04-050 #3-#7 bars)       ITON   | 500.000I 330.000|
|   33301|footing anchor bolt (03.1-05-125 1.5"x24")         IPCS   |  10.0001  20.000|
|   33302|footing anchor bolt sleeve (03.1-05-175)           IPCS   |   9.850I   2.000|
|   33303|column ftg. base plate (04.1-30-080 door opng.)    IPCS   |   6.000|  12.150|
|   26300 13000 psi ready-mix concrete (03.3-12-015)         ICUYD  |  48.9001   0.000|
|   2630112500 psi ready-mix concrete (03.3-12-010)          ICUYD  |  47.2501   0.0001
|   26302 13500 psi ready-mix concrete (03.3-12-020)         ICUYD  |  50.5501   0.0001
|   26303 13750 psi ready-mix concrete (03.3-12-025)         ICUYD  |  51.5001   0.0001
|   26304|4000 psi ready-mix concrete (03.3-12-030)          ICUYD  |  52.2001   0.000i
|   26610Ipour ret wall conc ftg - direct (03.3-38-190)      ICUYD  1   0.4501   6.7001
|   26611|pour ret wall conc ftg - carts (03.3-38-562)       ICUYD  |   1.1501   6.0501
|   26612Ipour ret wall conc ftg - pump (03.3-38-195)        ICUYD  1   5.6501  11.0001
|   26616|pour ret wall conc - direct (03.3-38-490)          ICUYD  I   0.600)   8.9501
|   26617|pour ret wall conc - carts (03.3-38-562)           ICUYD  1   1.1501   6.0501
|   26618Ipour ret wall conc - pump (03.3-38-495)            ICUYD  1   6.6501  12.950|
|   26619|pour ret wall conc - crane (03.3-38-500)           ICUYD  1   7.4501  13.700|
|   26623|Finish top of footing (03.3-26-001)                ISQFT  I   0.0001   0.170|
|   26630|Ftg. forms in Earth (03.1-45-005 2 use)            ISQFT  |   0.430|   1.440|
|   26631|Ftg. forms in Earth (03.1-45-005 2 use)            ISQFT  |   0.4301   1.440|
|   26632|Ftg. bulkhead (03.1-55-100 1 use)                  ISQFT  |   0.3401   1.240|
|   26640|Wall forms (03.1-65-215 2 use)                     ISQFT  |   0.7601   2.250|
|   26621|Point & patch (03.3-28-001)                        ISQFT  |   0.0101   0.2801
|   26651|Add batter wall forms (03.1-65-330 2 use)          ISQFT  |   0.0801   0.2301
|   26652|Add ledge form (03.1-65-030 2 use)                 ISQFT  |   0.7101   3.5501
|   26660IDovetail anchor slots (03.1-05-470 20 gauge)       ILNFT  |   0.380I   0.4001
|   42100IFace brick-std. (04.2-60-085 4" thick)             ISQFT  |   1.6101   3.4501
|   41900IClean face brick (04.2-08-001 smooth brick)        ISQFT  |   0.0801   0.5501
|   22214IBF-gravel - machine (02.3-05-034 1/2")             ICUYD  |  14.000|   2.810|
|   22215IBF-backrun - machine (02.3-05-010)                 ICUYD  |   6.5001   2.8101
|   22218IBF-earth borrow hand only (02.3-03-001)            ICUYD  |   0.0001   9.100|
|   22221|On-site earth-machine (02.3-03-130)                ICUYD  |   0.5301   0.1901
|  222211IMachine BF compaction (02.3-03-230 sheepsfoot)     ICUYD  |   1.100|   0.3501
|   22222lEarth borrow BF-machine (02.3-05-020)              ICUYD  |   5.000|   2.8101
|  222221IMachine BF compaction (02.3-03-230 sheepsfoot)     ICUYD  |   1.1001   0.3501
|   32104 Ire-steel - walls (03.2-04-070 #3-#7 bars)         |TON   | 505.0001 230.000|
|   26633 12x4 keyway (03.1-45-150 2x4)                      ILNFT  |   0.070|   0.3701
|   26634 12x6 keyway (03.1-45-155 2x6)                      |LNFT  |   0.090|   0.3201
|   2663512x8 keyway (03.1-45-155 2x6)                       |LNFT  |   0.090|   0.3201
|   2663613x8 keyway (03.1-45-155 2x6)                       |LNFT  |   0.0901   0.320|
|   26653IWall constr. joint (03.1-65-050 bulkhead)          |SQFT  |   1.3701   3.6901
|   26620|Rub exposed wall (03.1-28-005)                     |SQFT  |   0.050|   0.3401
|   26622|Finish top of wall (03.1-28-060)                   |SQFT  |   0.010|   0.510|
|   26665IWaterstop 6" PVC (03.1-90-005)                     |LNFT  |   1.2001   1.1001
|  510500|Add hi-strength steel (05.1-50-410)                |TON   | 150.0001   0.0001
|  510510|Add specialty steel (05.1-50-400)                  |TON   |  75.0001   0.0001
|  510600IFurn. struct, aluminum (05.1-05-010)               |TON   |1600.000|1490.000|
|  510610IFurn. extruded aluminum (05.1-05-030)              |TON   |2200.000|1490.0001
|  510140 1 Furn. Str. Stl. Beams (05.1-50-090)              |TON   | 865.0001 265.0001
|  510010 1 Furn. Str. Stl. H-columns (05.1-50-090)          |TON   | 865.0001 265.0001
|  510020 1 Furn. Str. Stl. Channels (05.1-30-060)           |TON   | 530.0001 220.000|
|  510030 1 Furn . Str. Stl. Angles (05.1-40-040)            |TON   | 580.0001 260.000|
|  510040 1 Furn . Str. Stl. Plates (05.1-50-090)            |TON   | 865.0001 265.0001
|  510070 1 Furn . Str. Stl. Tubular (05.1-26-360)           |TON   | 780.000|  70.000|
|  510080| Furn. Str. Stl. Pipe column (05.1-26-160)         |TON   | 600.0001  50.0001
|  5100601 Furn. Str. Stl. Bars (05.1-50-090)                |TON   | 865.000| 265.000|
|  5100501 Furn. Str. Stl. Rods (05.1-30-132)                |TON   |  53.0001  92.000|
|  991050IField paint str. stl. (05.1-50-652)                |SQFT  |   0.0501   0.1901
|  510520|Add galvanize str. stl. (05.1-50-600)              ITON   | 250.000]   0.0001
|  510100IFurnish anchor bolts (03.1-05-135)                 |LBS   |   0.5701   1.250!
|-----------------------------------------------------------------------------------|
```

7.2.3. Project Management Database Data

Prior to HICOST-II execution, the **quantity** relation in the project management database is empty. The takeoff process fills the QTYJD, COMPJD, UNITS, and AMOUNT slots with the values shown in the first of the following sections. The pricing process causes the values for COST slot to be updated as shown in the second section. In each case, the relation contains 80 tuples. The relations and attributes are defined in Section 5.2.2.3 of the thesis.

Project Management Database after Takeoff

quantity relation

```
Iqty_id        i comp_id        limits      |amount      |cost       |
|--------------------------------------------------------------------|
I        333031            14 |PCS       |     12.000|      0.000|
|        333021            14|PCS        |     24.0001      0.000|
I        333011            14IPCS        I     24.0001      0.000|
I        321011            14|TON        1      6.8921      0.000|
I       2220411            14|CUYD       I     69.8271      0.000|
I        222191            14ICUYD       |     69.8271      0.000|
I        222071            14ICUYD       |     20.000|      0.000|
I       2220331            14ICUYD       I    111.6541      0.000|
I        31014!            14ISQFT       I    540.0001      0.000|
I        301361            14ISQFT       !    672.0001      0.000|
I        331671            14ICUYD       I     37.3331      0.000|
I        331021            14ICUYD       |     37.3331      0.000|
I        310081            13ISQFT       I     15.9601      0.000|
I        310061            13ISQFT       |      8.858|      0.000|
I        331201            13ICUYD       1      0.9841      0.000|
I        321011            12|TON        I      0.2841      0.000|
I       2220411            12ICUYD       I     45.3001      0.000|
I        22204 1           12ICUYD       |     45.3001      0.000|
I        222051            12ICUYD       |      0.867|      0.000|
I       2220361            12ICUYD       |     46.9661      0.000|
I        310091            12ISQFT       |     48.000!      0.000|
I        310061            12ISQFT       |     18.000|      0.000|
I        301361            12ISQFT       |     45.0001      0.000i
I        331201            12ICUYD       |      1.667|      0.000!
I        331001            12ICUYD       |      1.6671      0.000!
I        321011            111 TON       |      2.144!      0.000I
I        310111            11ILNFT       |    150.000|      0.0001
I       2220411            11ICUYD       |    149.6031      0.000|
I        22204 1           11ICUYD       |    149.6031      0.0001
I        222051            11ICUYD       |      5.050|      0.000|
I       2220361            11ICUYD       |    190.3811      0.000i
I        31006!            11ISQFT       |    300.000!      0.000i
I        301361            11ISQFT       |    450.0001      0.000|
I        331201            11ICUYD       |     16.6671      0.000i
|        331001            11ICUYD       |     16.6671      0.000i
|        33303I            34|PCS        |      1.000|      0.000i
|        333021            34|PCS        |      2.0001      0.000i
|        333011            34|PCS        |      2.000|      0.000i
|       2220411            34ICUYD       |      5.692!      0.000i
|        222191            34ICUYD       |      5.692I      0.000i
|        222071            34ICUYD       |      3.1111      0.000i
|       222033|            34ICUYD       |     10.5021      0.000|
|        31014|            34ISQFT       |      0.000|      0.000i
|        301361            34ISQFT       |      0.0001      0.000|
|        33167|            34ICUYD       |      0.000|      0.000|
|        331001            34ICUYD       |      0.000|      0.000|
|        33303|            33|PCS        |      1.000|      0.000|
I        333021            33 |PCS       |      2.000I      0.000|
|        333011            33IPCS        |      2.0001      0.000|
|       2220411            33ICUYD       |     72.2541      0.000|
|        222191            33ICUYD       |     72.254!      0.000|
|        222071              33ICUYD       |   24.5561      0.0001
|       2220331              33ICUYD       |  130.934|      0.0001
|        310141              33ISQFT       |  481.500|      0.0001
|        301361              33ISQFT       | 1001.0001      0.0001
|        331671              33ICUYD       |   55.6111      0.0001
|        331001              33ICUYD       |   55.6111      0.0001
|        33303!              32|PCS        |    1.0001      0.000|
|        333021              32IPCS        |    2.0001      0.000I
|        333011              32IPCS        |    2.0001      0.000I
|       2220411              32ICUYD       |    4.6931      0.000I
|        222191              32ICUYD       |    4.693|      0.000!
|        22207|              32ICUYD       |    0.333|      0.000i
|       2220351              32ICUYD       |    1.5531      0.000i
|       2220331              32ICUYD       |    1.9091      0.000i
|        31014|              32ISQFT       |   46.620I      0.000|
|        301361              32ISQFT       |    0.0001      0.000|
|        331671              32ICUYD       |    0.0001      0.000i
|        331001              32ICUYD       |    0.000|      0.000|
|        333031              31IPCS        |    1.0001      0.000i
|        333021              31IPCS        |    2.0001      0.000i
|        33301!              31IPCS        |    2.000|      0.000I
|       2220411              31ICUYD       |   15.1371      0.000i
|        222191              31ICUYD       |   15.1371      0.000I
|        222071              31ICUYD       |    5.3331      0.000i
|       2220331              31ICUYD       |   26.058|      0.000i
|        310141              31ISQFT       |   92.5001      0.000i
|        301361              31 ISQFT      |  210.0001      0.000i
|        331671              31ICUYD       |    9.7221      0.000i
|        331001              31ICUYD       |    9.7221      0.0001
|-------------------------------------------------------------|
```

Project Management Database after Pricing

quantity relation

```
|qty_id         |comp_id        |units      1 amount    |cost       |
|-------------------------------------------------------------------|
|        333031              14 IPCS        I    12.0001   217.800|
|        33302|              14 IPCS        1    24.0001   284.400|
|        333011              14 IPCS        I    24.0001   720.000|
|        321011              14|TON         I     6.8921  5720.360|
|       2220411              14ICUYD        |    69.827|   114.516|
|        222191              14ICUYD        |    69.8271   754.830|
|        222071              14ICUYD        |    20.000|   640.0001
|       2220331              14ICUYD        |   111.654I   293.650|
|        31014|              14ISQFT        |   540.0001  1009.800|
|        301361              14ISQFT        |   672.000I   114.240!
|        331671              14ICUYD        |    37.333|   621.594I
|        331021              14ICUYD        |    37.3331  1887.183I
|        310081              13ISQFT        |    15.9601    25.2171
|        310061              13ISQFT        |     8.8581    16.5641
|        331201              13ICUYD        |     0.984|     7.0361
|        321011              12|TON         |     0.2841   235.7201
|       2220411              12ICUYD        |    45.3001    74.292I
|        22204|              12ICUYD        |    45.3001    32.6161
|        222051              12ICUYD        |     0.8671    13.785I
|       2220361              12ICUYD        |    46.9661   123.5211
|        310091              12ISQFT        |    48.000I   185.7601
|        310061              12ISQFT        |    18.000|    33.660|
|        301361              12ISQFT        |    45.0001     7.650I
|        331201              12ICUYD        |     1.667!    11.9191
|        33100|              12ICUYD        |     1.667 |   81.5161
|        32101|              111 TON        |     2.1441  1779.5201
|        310111              11ILNFT        |   150.000|    55.5001
|       2220411              11ICUYD        |   149.603|   245.3491
|        22204I              11ICUYD        |   149.603|   107.714I
|        222051              11ICUYD        |     5.0501    80.2951
|       2220361              11ICUYD        |   190.3811   500.7021
|        310061              11ISQFT        |   300.000|   561.000!
|        301361              11 ISQFT       |   450.000|    76.5001
|          33120|           11ICUYD       |    16.6671    119.1691
|          33100|           11ICUYD       |    16.6671    815.0161
|          33303|           34|PCS        |     1.0001     18.1501
|          33302|           34|PCS        |     2.0001     23.7001
|          33301|           34 |PCS       |     2.0001     60.0001
|         222041|           3 4ICUYD      |     5.6921      9.3351
|          22219|           34ICUYD       |     5.6921     61.5311
|          22207|           34 ICUYD      |     3.1111     99.5521
|         222033|           34ICUYD       |    10.5021     27.6201
|          31014|           34ISQFT       |     0.0001      0.0001
|          30136|           34ISQFT       |     0.0001      0.000!
|          33167|           34ICUYD       |     0.0001      0.0001
|          33100|           34ICUYD       |     0.0001      0.0001
|          33303|           33 |PCS       |     1.0001     18.1501
|          33302|           33 |PCS       |     2.0001     23.700|
|          33301|           33|PCS        |     2.0001     60.000|
|         222041|           33ICUYD       |    72.2541    118.497|
|          22219|           33ICUYD       |    72.2541    781.066|
|          22207|           3 3ICUYD      |    24.5561    785.792|
I         222033|           33ICUYD       |   130.9341    344.356|
I          31014|           33 ISQFT      |   481.5001    900.405|
I          30136|           33ISQFT       |  1001.0001    170.170|
1          33167|           33ICUYD       |    55.6111    925.923|
I          33100|           33ICUYD       |    55.6111   2719.378|
I          33303|           32 |PCS       |     1.0001     18.150|
1          33302|           32|PCS        |     2.0001     23.700|
I          33301|           32IPCS        |     2.0001     60.000|
I         222041|           32ICUYD       |     4.6931      7.697|
I          22219|           32ICUYD       |     4.6931     50.731|
I          22207|           32ICUYD       |     0.3331     10.656|
I         222035|           32ICUYD       |     1.5531      4.084|
I         222033|           32ICUYD       |     1.9091      5.021|
I          31014|           32ISQFT       |    46.6201     87.179|
I          30136|           32ISQFT       |     0.0001      0.000|
I          33167|           32ICUYD       |     0.0001      0.000|
I          33100|           32ICUYD       |     0.0001      0.000|
I          33303|           31 IPCS       |     1.0001     18.150|
I          33302|           31|PCS        |     2.0001     23.700|
I          33301|           31IPCS        |     2.0001     60.000|
|         222041|           31ICUYD       |    15.1371     24.825|
|          22219|           31ICUYD       |    15.1371    163.631|
|          22207|           31ICUYD       |     5.3331    170.656|
|         222033|           31ICUYD       |    26.0581     68.533|
|          31014|           31 ISQFT      |    92.5001    172.975|
|          30136|           31ISQFT       |   210.0001     35.700|
|          33167|           31ICUYD       |     9.7221    161.8711
|          33100|           31ICUYD       |     9.7221    475.406|
|-------------------------------------------------------------------|
```

7.3. Component Takeoff Rules

The following sections contain the complete takeoff rule sets for concrete footings, retaining walls, and structural metal. Section 5.2.2.1 of the thesis contains a description of four sample rules taken from these rule sets.

7.3.1. Concrete Footing Takeoff Rule Set

```
 ;;; Rules for quality of concrete
;;;
(take-off-rule  concrete-footing-quality-of-concrete1
  if (    quality-of-concrete == "3000 psi standard mix"
      and (   concrete-placing-method-selection == "none required"
           or concrete-placing-method-selection == "wall ftg pour concrete direct"
           or concrete-placing-method-selection == "wall ftg pour concrete w/carts"
           or concrete-placing-method-selection == "wall ftg pump concrete"
           or concrete-placing-method-selection == "wall ftg pour concrete w/crane"
           or concrete-placing-method-selection == "column ftg pour concrete direct"
           or concrete-placing-method-selection == "column ftg pour concrete w/carts"
           or concrete-placing-method-selection == "column ftg pump concrete"
           or concrete-placing-method-selection == "column ftg pour concrete w/crane"
          )
     )
  then (quantity-id = 33100 ,
        units = "CUYD" ,
        amount =   number * (length * width * footing-thickness) / 27.0 ,
        make-quantity(quantity-id,units,amount)
       )
)

(take-off-rule  concrete-footing-quality-of-concrete2
  if (    quality-of-concrete == "deduct for 2500 psi"
      and concrete-placing-method-selection == "stepped wall ftg concrete"
     )
  then (quantity-id = 33101 ,
        units = "CUYD" ,
        amount =   number * (width * footing-thickness
                                    * (1 + 0.5 * footing-thickness)) / 27.0 ,
        make-quantity(quantity-id,units,amount)
       )
)

(take-off-rule  concrete-footing-quality-of-concrete3
  if (    quality-of-concrete == "deduct for 2500 psi"
      and (   concrete-placing-method-selection == "none required"
           or concrete-placing-method-selection == "wall ftg pour concrete direct"
           or concrete-placing-method-select ion == "wall ftg pour concrete w/carts"
           or concrete-placing-method-select ion == "wall ftg pump concrete"
           or concrete-placing-method-select ion == "wall ftg pour concrete w/crane"
           or concrete-placing-method-select ion == "column ftg pour concrete direct"
           or concrete-placing-method-selection == "column ftg pour concrete w/carts"
           or concrete-placing-method-select ion == "column ftg pump concrete"
           or concrete-placing-method-select ion == "column ftg pour concrete w/crane"
          )
     )
  then (quantity-id = 33101 ,
        units = "CUYD" ,
        amount =   number * (length * width * footing-thickness) / 27.0 ,
        make-quantity(quantity-id,units,amount)
       )
)

(take-off-rule  concrete-footing-quality-of-concrete4
  if (    quality-of-concrete == "deductfor 2500 psi"
      and concrete-placing-method-selection == "stepped wall ftg concrete"
     )
  then (quantity-id = 33100 ,
        units = "CUYD" ,
        amount =   number * (width * footing-thickness
                                    * (1 + 0.5 * footing-thickness)) / 27.0 ,
        make-quantity(quant ity-id,units,amount)
       )
)
```

```
 (take-off-rule  concrete-footing-quality-of-concrete5
   if (    quality-of-concrete == "add for 3500 psi"
       and (    concrete-placing-method-selection == "none required"
            or  concrete-placing-method-selection == "wall ftg pour concrete direct"
            or  concrete-placing-method-selection == "wall ftg pour concrete w/carts"
            or  concrete-placing-method-selection == "wall ftg pump concrete"
            or  concrete-placing-method-selection == "wall ftg pour concrete w/crane"
            or  concrete-placing-method-selection == "column ftg pour concrete direct"
            or  concrete-placing-method-selection == "column ftg pour concrete w/carts"
            or  concrete-placing-method-selection == "column ftg pump concrete"
            or  concrete-placing-method-selection == "column ftg pour concrete w/crane"
           )
      )
   then  (quantity-id = 33102 ,
          units = "CUYD" ,
          amount =   number * (length * width * footing-thickness) / 27.0 ,
          make-quantity(quantity-id,units,amount)
        )
)

 (take-off-rule  concrete-footing-quality-of-concrete6
   if (    quality-of-concrete == "add for 3500 psi"
       and concrete-placing-method-selection == "stepped wall ftg concrete"
      )
   then  (quantity-id = 33102 ,
          units = "CUYD" ,
          amount =   number * (width * footing-thickness
                                    * (1 + 0.5 * footing-thickness)) / 27.0 ,
          make-quantity(quantity-id,units,amount)
        )
)

 (take-off-rule  concrete-footing-quality-of-concrete7
   if (    quality-of-concrete == "add for 3750 psi"
       and (    concrete-placing-method-selection == "none required"
            or  concrete-placing-method-selection == "wall ftg pour concrete direct"
            or  concrete-placing-method-selection == "wall ftg pour concrete w/carts"
            or  concrete-placing-method-selection == "wall ftg pump concrete"
            or  concrete-placing-method-selection == "wall ftg pour concrete w/crane"
            or  concrete-placing-method-selection == "column ftg pour concrete direct"
            or  concrete-placing-method-selection == "column ftg pour concrete w/carts"
            or  concrete-placing-method-selection == "column ftg pump concrete"
            or  concrete-placing-method-selection == "column ftg pour concrete w/crane"
           )
      )
   then  (quantity-id = 33103 ,
          units = "CUYD" ,
          amount =   number * (length * width * footing-thickness) / 27.0 ,
          make-quantity(quantity-id,units,amount)
        )
)

 (take-off-rule  concrete-footing-quality-of-concrete8
   if (    quality-of-concrete == "add for 3750 psi"
       and concrete-placing-method-selection == "stepped wall ftg concrete"
      )
   then  (quantity-id = 33103 ,
          units = "CUYD" ,
          amount =   number * (width * footing-thickness
                                    * (1 + 0.5 * footing-thickness))  / 27.0 ,
          make-quantity(quantity-id,units,amount)
        )
)
```

```
(take-off-rule  concrete-footing-quality-of-concrete9
 if (    quality-of-concrete == "add for 4000 psi"
    and (   concrete-placing-method-selection == "none required"
         or concrete-placing-method-selection == "wall ftg pour concrete direct"
         or concrete-placing-method-selection == "wall ftg pour concrete w/carts"
         or concrete-placing-method-selection == "wall ftg pump concrete"
         or concrete-placing-method-selection == "wall ftg pour concrete w/crane"
         or concrete-placing-method-selection == "column ftg pour concrete direct"
         or concrete-placing-method-selection == "column ftg pour concrete w/carts"
         or concrete-placing-method-selection == "column ftg pump concrete"
         or concrete-placing-method-selection == "column ftg pour concrete w/crane"
        )
    )
 then (quantity-id = 33104 ,
       units = "CUYD" ,
       amount =   number * (length * width * footing-thickness) / 27.0 ,
       make-quantity(quantity-id,units,amount)
      )
)

(take-off-rule  concrete-footing-quality-of-concrete10
 if (    quality-of-concrete == "add for 4000 psi"
    and concrete-placing-method-selection == "stepped wall ftg concrete"
    )
 then (quantity-id = 33104 ,
       units = "CUYD" ,
       amount =   number * (width * footing-thickness
                               * (1 + 0.5 * footing-thickness)) / 27.0 ,
       make-quantity(quantity-id,units,amount)
      )
)

;;; Rules for concrete placement
;;;
(take-off-rule  concrete-footing-concrete-and-placing-method1
 if (concrete-placing-method-selection == "wall ftg pour concrete direct")
 then (quantity-id = 33120 ,
       units = "CUYD" ,
       amount =   number * (length * width * footing-thickness) / 27 ,
       make-quantity(quantity-id,units,amount) ,
       quantity-id = 30136 ,
       units = "SQFT" ,
       amount =   number * (length * width) ,
       make-quantity(quantity-id,units,amount)
      )
)

(take-off-rule  concrete-footing-concrete-and-placing-method2
 if (concrete-placing-method-selection == "wall ftg pour concrete w/carts")
 then (quantity-id = 33120 ,
       units = "CUYD" ,
       amount =   number * (length * width * footing-thickness) / 27.0 ,
       make-quantity(quantity-id,units,amount) ,
       quantity-id = 33130 ,    ;;; (Walking cart, 150' haul)
       units = "CUYD" ,
       amount =   number * (length * width * footing-thickness) / 27.0 ,
       make-quantity(quantity-id,units,amount) ,
       quantity-id = 30136 ,
       units = "SQFT" ,
       amount =   number * (length * width) ,
       make-quantity(quantity-id,units,amount)
      )
)
```

```
(take-off-rule  concrete-footing-concrete-and-placing-method3
  if  (concrete-placing-method-selection == "wall ftg pump concrete")
  then  (quantity-id = 33151 ,
        units = "CUYD" ,
        amount =   number *  (length * width * footing-thickness)  / 27.0 ,
        make-quantity(quantity-id,units,amount)  ,
        quantity-id = 30136  ,
        units = "SQFT" ,
        amount =   number *  (length * width),
        make-quantity(quantity-id,units,amount)
       )
)

(take-off-rule  concrete-footing-concrete-and-placing-method4
  if  (concrete-placing-method-selection == "wall ftg pour concrete w/crane")
  then  (quantity-id = 33140  ,
               units = "CUYD"  ,
               amount =   number *  (length * width * footing-thickness)  / 27.0 ,
        make-quantity(quantity-id,units,amount)  ,
        quantity-id = 30136  ,
               units = "SQFT"  ,
               amount =   number *  (length * width)  ,
        make-quantity(quantity-id,units,amount)
       )
)

(take-off-rule  concrete-footing-concrete-and-placing-method5
  if  (concrete-placing-method-selection == "column ftg pour concrete direct")
  then  (quantity-id = 33165  ,
        units = "CUYD" ,
        amount =   number *  (length * width * footing-thickness)  / 27.0  ,
        make-quantity(quant ity-id,units,amount)  ,
        quantity-id = 30136  ,
        units = "SQFT" ,
        amount =   number *  (length * width)  ,
        make-quantity(quantity-id,units,amount)
       )
)

(take-off-rule  concrete-footing-concrete-and-placing-method6
  if  (concrete-placing-method-selection == "column ftg pour concrete w/carts")
  then  (quantity-id = 33165  ,
        units = "CUYD" ,
        amount =   number *  (length * width * footing-thickness)  / 27.0  ,
        make-quantity(quantity-id,units,amount)  ,
        quantity-id = 33166  ,    ;;;  (Walking cart,  150'  haul)
        units = "CUYD" ,
        amount =   number *  (length * width * footing-thickness)  / 27.0  ,
        make-quantity(quantity-id,units,amount),
        quantity-id = 30136  ,
        units = "SQFT" ,
        amount =   number *  (length * width)  ,
        make-quantity(quantity-id,units,amount)
       )
)

(take-off-rule  concrete-footing-concrete-and-placing-method7
  if  (concrete-placing-method-selection == "column ftg pump concrete")
  then  (quantity-id = 33167  ,
               units = "CUYD"  ,
               amount =   number *  (length * width * footing-thickness)  / 27.0 ,
        make-quantity(quantity-id,units,amount)  ,
        quantity-id = 30136  ,
               units = "SQFT"  ,
               amount =   number *  (length * width)  ,
        make-quant ity(quantity-id,units,amount)
       )
)
```

```
(take-off-rule  concrete-footing-concrete-and-placing-method8
  if  (concrete-placing-method-selection == "column  ftg  pour  concrete  w/crane")
  then  (quantity-id = 33168  ,
               units = "CUYD"  ,
               amount =   number  *  (length  *  width  *  footing-thickness)  /  27.0  ,
        make-quantity(quantity-id,units, amount) ,
        quantity-id = 30136  ,
               units = "SQFT"  ,
               amount =   number  *  (length  *  width)  ,
        make-quantity(quantity-id,units,amount)
       )
)

(take-off-rule  conerete-foot ing-concrete-and-placing-method9
  if  (concrete-placing-method-selection == "stepped  wall  ftg  concrete")
  then  (quantity-id = 33120  ,
        units - "CUYD"  ,
        amount =   number  *  (width  *  footing-thickness
                                    *  (1 + 0.5  *  footing-thickness))  /  27.0  ,
        make-quantity(quantity-id,units,amount)
       )
)

;;;  Rules  for  wall  and  column  footing  forms.
;;;
(take-off-rule  concrete-footing-wall-and-column-foot ing-forms1
  if  (footing-forms-selection == "form  wall  ftg  earth")
  then  (quantity-id = 31006  ,
        units = "SQFT"  ,
        amount =   number  *  (2  *  length  *  footing-thickness)  ,
        make-quantity(quantity-id,units,amount)
      )
)

(take-off-rule  concrete-foot ing-wall-and-column-footing-forms2
  if  (footing-forms-selection == "form  wall  ftg  rock")
  then  (quantity-id = 31013  ,
        units = "SQFT"  ,
        amount =   number  *  (2  *  length  *  footing-thickness)  ,
        make-quantity(quantity-id,units,amount)
       )
)

(take-off-rule  concrete-foot ing-wall-and-column-foot ing-forms3
  if  (footing-forms-selection == "form  column  ftg  earth")
  then  (quantity-id = 31007  ,
        units = "SQFT"  ,
        amount =   number  *  ((2  *  length  +  2  *  width)
                              *  footing-thickness)  ,
        make-quantity(quantity-id,units,amount)
       )
)
```

```
(take-off-rule  concrete-footing-wall-and-column-foot ing-forms4
  if  (footing-forms-selection == "form column  ftg  rock")
  then  (quantity-id = 31014  ,
        units = "SQFT"  ,
        amount =   number *  ((2  *  length + 2  * width)
                              * footing-thickness)  ,
        make-quantity(quantity-id, units,amount)
       )
)
(take-off-rule  concrete-footing-wall-and-column-foot ing-forms5
  if  (footing-forms-selection == "stepped  ftg  forms")
  then  (quantity-id = 31006  ,
        units = "SQFT"  ,
        amount =   number *  (footing-thickness
                              *  (  1  +  footing-thickness  12)),
        make-quantity(quantity-id,units,amount)  ,
        quantity-id = 31008  ,
        units = "SQFT"  ,
        amount =   number *  (width * footing-thickness)  ,
        make-quantity(quant ity-id,units,amount)
       )
)
(take-off-rule  concrete-foot ing-wall-and-column-footing-forms6
  if  (footing-forms-selection == "pilaster  wall  ftg")
  then  (quantity-id = 31006  ,
        units = "SQFT"  ,
        amount =   number *  (2 * width * footing-thickness)  ,
        make-quantity(quantity-id,units,amount)  ,
        quantity-id = 31009  ,
        units = "SQFT"  ,
        amount =   number *  ((2  * width + length)
                              * footing-thickness)  ,
        make-quantity(quantity-id,units,amount)
       )
)
(take-off-rule  concrete-footing-wall-and-column-foot ing-forms7
  if  (footing-forms-selection == "curved  wall  ftg")
  then  (quantity-id = 31010  ,
        units = "SQFT"  ,
        amount =   number *  (2 * length * footing-thickness)  ,
        make-quantity(quantity-id,units,amount)
       )
)
(take-off-rule  concrete-footing-wall-and-column-foot ing-forms8
  if  (footing-forms-selection == "pile  cap  forms")
  then  (quantity-id = 31016  ,
        units = "SQFT"  ,
        amount =   number *  ((2  *  length + 2  * width)
                              * footing-thickness)  ,
        make-quantity(quantity-id,units,amount)
       )
)
(take-off-rule  concrete-foot ing-wall-and-column-footing-forms9
  if  (footing-forms-selection == "wall  ftg  bulkhds.")
  then  (quantity-id = 31008  ,
        units = "SQFT"  ,
        amount =   number *  (width * footing-thickness)  ,
        make-quantity(quantity-id,units,amount)
       )
)
```

```
;;; Rules for earth excavation
;;;
(take-off-rule  concrete-footing-earth-excavationl
  if (    earth-excavation-selection == "mach. to top of rock"
      and grade-elevation - rock-elevation <= 3 )
  then (quantity-id = 222033 ,
        units = "CUYD" ,
        amount =   number *
           ((length + 2 + (grade-elevation - rock-elevation) / 3)
            *  (width•+ 2 + (grade-elevation - rock-elevation) / 3)
            * ( grade-elevation - rock-elevation)) / 21 ,
        make-quantity(quantity-id,units,amount)
       }
)

(take-off-rule  concrete-footing-earth-excavation2
  if (    earth-excavation-selection == "mach. to top of rock"
      and grade-elevation - rock-elevation > 3
      and grade-elevation - rock-elevation <= 4 )
  then (quantity-id = 222034 ,
        units = "CUYD" ,
        amount =   number *
           ((length + 2 + (grade-elevation - rock-elevation) / 3)
            *  (width + 2 + (grade-elevation - rock-elevation) / 3)
            * ( grade-elevation - rock-elevation)) / 27 ,
        make-quantity(quantity-id,units,amount)
       )
)

(take-off-rule  concrete-footing-earth-excavation3
  if (    earth-excavation-selection == "mach. to top of rock"
      and grade-elevation - rock-elevation > 4
      and grade-elevation - rock-elevation <= 5 )
  then (quantity-id = 222035 ,
        units = "CUYD" ,
        amount =   number *
           ((length + 2 + (grade-elevation - rock-elevation) / 3)
            *  (width + 2 + (grade-elevation - rock-elevation) / 3)
            * ( grade-elevation - rock-elevation)) / 27 ,
        make-quantity(quant ity-id,units,amount)
       )
)

(take-off-rule  concrete-footing-earth-excavation4
  if (    earth-excavation-selection == "mach. to top of rock"
      and grade-elevation - rock-elevation > 5
      and grade-elevation - rock-elevation <= 6 )
  then (quantity-id = 222036 ,
        units = "CUYD" ,
        amount =   number *
           ((length + 2 + (grade-elevation - rock-elevation) / 3)
            *  (width + 2 + (grade-elevation - rock-elevation) / 3)
            * ( grade-elevation - rock-elevation)) / 27 ,
        make-quantity(quantity-id,units,amount)
       )
)

(take-off-rule  concrete-footing-earth-excavat ion5
  if (    earth-excavation-selection == "hand excavation only"
      and rock-excavation-selection == "none-required" )
  then (quantity-id = 22205 ,
        units = "CUYD" ,
        amount =   number *
           ((length + 1.5 + (grade-elevation - top-elevation
                                              + footing-thickness) / 3)
            *  (width + 1.5 + (grade-elevation - top-elevation
                                               + footing-thickness) / 3)
            * ( grade-elevation - top-elevation + footing-thickness)) / 27 ,
        make-quantity(quantity-id,units,amount)
       )
)
```

```
(take-off-rule  concrete-footing-earth-excavation6
  if (    earth-excavation-selection == "hand excavation only"
      and (   rock-excavation-selection == "machine w/ hand dress-up"
           or rock-excavation-selection == "hand excavation only"
           or rock-excavation-selection == "blasting w/machine & hand dress-up" ))
  then (quantity-id = 22205 ,
        units = "CUYD" ,
        amount =   number *
           ((length + 2 + (grade-elevation - rock-elevation) / 3)
            * (width + 2 + (grade-elevation - rock-elevation) / 3)
            * ( grade-elevation - rock-elevation)) / 21 ,
        make-quantity(quantity-id,units,amount)
       )
)

(take-off-rule  concrete-footing-earth-excavation7
  if (    earth-excavation-selection == "machine w/ hand dress-up"
      and grade-elevation - top-elevation + footing-thickness <= 3 )
  then (quantity-id = 222033 ,
        units = "CUYD" ,
        amount =   number *
           ((length + 1.5 + (grade-elevation - top-elevation
                                              + footing-thickness) / 3)
            * (width + 1.5 + (grade-elevation - top-elevation
                                               + footing-thickness) / 3)
            * ( grade-elevation - top-elevation + footing-thickness)) / 27 ,
        make-quantity(quantity-id,units,amount),
        quantity-id = 22205 ,
        units = "CUYD" ,
        amount =   number *  ((length + 1.5)  * (width + 1.5)  * 0.2)  / 27 ,
        make-quantity(quantity-id,units,amount)
       )
)

(take-off-rule  concrete-footing-earth-excavation8
  if (    earth-excavation-selection == "machine w/ hand dress-up"
      and grade-elevation - top-elevation + footing-thickness > 3
      and grade-elevation - top-elevation + footing-thickness <= 4 )
  then (quantity-id = 222034 ,
        units = "CUYD" ,
        amount =   number *
           ((length + 1.5 + (grade-elevation - top-elevation
                                              + footing-thickness) / 3)
            * (width + 1.5 + (grade-elevation - top-elevation
                                               + footing-thickness) / 3)
            * ( grade-elevation - top-elevation + footing-thickness)) / 27 ,
        make-quantity(quantity-id,units,amount) ,
        quantity-id = 22205 ,
        units = "CUYD" ,
        amount =   number *  ((length + 1.5)  * (width + 1.5)  * 0.2)  / 27 ,
        make-quantity(quantity-id,units,amount)
       )
)

(take-off-rule  concrete-footing-earth-excavation9
  if (    earth-excavation-selection == "machine w/ hand dress-up"
      and grade-elevation - top-elevation + footing-thickness > 4
      and grade-elevation - top-elevation + footing-thickness <= 5 )
  then (quantity-id = 222035 ,
        units = "CUYD" ,
        amount =   number *
           ((length + 1.5 + (grade-elevation - top-elevation
                                              + footing-thickness) / 3)
            * (width + 1.5 + (grade-elevation - top-elevation
                                               + footing-thickness) / 3)
            * ( grade-elevation - top-elevation + footing-thickness)) / 27 ,
        make-quantity(quantity-id,units,amount) ,
        quantity-id = 22205 ,
        units = "CUYD" ,
        amount =   number *  ((length + 1.5)  * (width + 1.5)  * 0.2)  / 27 ,
        make-quantity(quantity-id,units,amount)
       )
)
```

```
(take-off-rule  concrete-footing-earth-excavation10
  if (    earth-excavation-selection == "machine w/ hand dress-up"
      and grade-elevation - top-elevation + footing-thickness > 5
      and grade-elevation - top-elevation + footing-thickness <= 6 )
  then  (quantity-id = 222036 ,
         units = "CUYD" ,
         amount =   number *
             ((length + 1.5 +  (grade-elevation - top-elevation
                                               + footing-thickness)  / 3)
              *  (width + 1.5 +  (grade-elevation - top-elevation
                                                + footing-thickness)  / 3)
              *  ( grade-elevation - top-elevation + footing-thickness)) / 21 ,
         make-quantity(quantity-id,units,amount)  ,
         quantity-id = 22205  ,
         units = "CUYD11 ,
         amount =   number *  ((length + 1.5)  *  (width + 1.5)  * 0.2)  / 27 ,
         make-quantity(quantity-id,units,amount)
        )
)

;;; Rules for rock excavation
;;;
(take-off-rule  concrete-footing-rock-excavation1
  if (    rock-excavation-selection == "machine w/ hand dress-up"
      and grade-elevation - top-elevation + footing-thickness <= 3 )
           ;;; use total depth of trench to determine machine excavation cost
  then  (quantity-id = 222033 ,
         units = "CUYD" ,
         amount =   number *
                  ((length + 2)  *  (width + 2)
                   *  (rock-elevation - top-elevation + footing-thickness))  / 27.0 ,
         make-quantity(quantity-id,units,amount)  ,
         quantity-id = 22207  ,
         units = "CUYD" ,
         amount =   number *  ((length + 2)  *  (width + 2)  * 0.5)  / 27.0  ,
         make-quantity(quantity-id,units,amount)
        )
)

(take-off-rule  concrete-footing-rock-excavation2
  if (    rock-excavation-selection == "machine w/ hand dress-up"
      and grade-elevation - top-elevation + footing-thickness > 3
      and grade-elevation - top-elevation + footing-thickness <= 4 )
           ;;; use total depth of trench to determine machine excavation cost
  then  (quantity-id = 222034 ,
         units = "CUYD" ,
         amount =   number *
                  ((length + 2)  *  (width + 2)
                   *  (rock-elevation - top-elevation + footing-thickness))  / 27.0 ,
         make-quantity(quantity-id,units,amount)  ,
         quantity-id = 22207  ,
         units = "CUYD" ,
         amount =   number *  ((length + 2)  *  (width +2 )  * 0.5)  / 27.0  ,
         make-quantity(quantity-id,units,amount)
        )
)

(take-off-rule  concrete-footing-rock-excavation3
  if (    rock-excavation-selection == "machine w/ hand dress-up"
      and grade-elevation - top-elevation + footing-thickness > 4
      and grade-elevation - top-elevation + footing-thickness <= 5 )
           ;;; use total depth of trench to determine machine excavation cost
  then  (quantity-id = 222035 ,
         units = "CUYD" ,
         amount =   number *  ((length +2 )  *  (width + 2)
                   *  (rock-elevation - top-elevation + footing-thickness))  / 27.0 ,
         make-quantity(quantity-id,units,amount)  ,
         quantity-id = 22207  ,
         units = "CUYD" ,
         amount =   number *  ((length + 2)  *  (width +2 )  * 0.5)  / 27.0  ,
         make-quantity(quantity-id,units,amount)
        )
)
```

```
(take-off-rule  concrete-footing-rock-excavation4
  if (    rock-excavation-selection == "machine w/ hand dress-up"
      and grade-elevation - top-elevation + footing-thickness > 5
      and grade-elevation - top-elevation + footing-thickness <= 6 )
          ;;; use total depth of trench to determine machine excavation cost
  then (quantity-id = 222036 ,
        units = "CUYD" ,
        amount =   number * ((length +2 ) * (width + 2)
                 * (rock-elevation - top-elevation + footing-thickness)) / 27.0 ,
        make-quantity(quantity-id,units,amount) ,
        quantity-id = 22207 ,
        units = "CUYD" ,
        amount =  number * ((length + 2) * (width + 2) * 0.5) / 27.0 ,
        make-quantity(quantity-id,units,amount) ,
       )
)

(take-off-rule  concrete-footing-rock-excavation5
  if (rock-excavation-selection == "hand excavation only")
  then (quantity-id = 22207 ,
        units = "CUYD" ,
        amount =  number * ((length +2 ) * (width + 2) * 0.5) / 27.0 ,
        make-quantity(quantity-id,units,amount)
       )
)

(take-off-rule  concrete-footing-rock-excavation6
  if (rock-excavation-selection == "blasting w/machine & hand dress-up")
  then (quantity-id = 22211 ,
        units = "CUYD" ,
        amount =   number * ((length +2 ) * (width + 2)
                 * (rock-elevation - top-elevation + footing-thickness)) / 27.0 ,
        make-quantity(quantity-id,units,amount) ,
        quantity-id = 22207 ,
        units = "CUYD" ,
        amount =  number * ((length + 2) * (width + 2) * 0.5) / 27.0 ,
        make-quantity(quantity-id,units,amount)
       )
)

;;; Rules for trench bracing
;;;
(take-off-rule  concrete-footing-trench-bracing-selection1
  if (trench-bracing-selection == "jacks only wall ftg")
  then (quantity-id = 24104 ,
        units = "CUYD" ,
        amount =   number * (2 * length
                 * (grace-elevation - top-elevation + footing-thickness)) ,
        make-quantity(quantity-id,units,amount)
       )
)

(take-off-rule  concrete-footing-trench-bracing-selection2
  if (trench-bracing-selection == "open plank wall ftg")
  then (quantity-id = 24100 ,
        units = "CUYD" ,
        amount =   number * (2 * length
                 * (grace-elevation - top-elevation + footing-thickness)) ,
        make-quantity(quantity-id,units,amount)
       )
)

(take-off-rule  concrete-footing-trench-bracing-selection3
  if (trench-bracing-selection == "solid wood wall ftg")
  then (quantity-id = 24101 ,
        units = "CUYD" ,
        amount =   number * (2 * length
             * (grace-elevation - top-elevation + footing-thickness)) ,
        make-quantity(quantity-id,units,amount)
       )
)
```

```
(take-off-rule  concrete-foot ing-trench-bracing-selection4
  if (trench-bracing-selection == "solid wood col. ftg")
  then (quantity-id = 24101 ,
        units = "CUYD" ,
        amount =   number * (2 * (length +2) + 2 * (width + 2)
                 * (grace-elevation - top-elevation + footing-thickness)) ,
        make-quantity(quantity-id,units,amount)
       )
)

(take-off-rule  concrete-footing-trench-bracing-selection5
  if (trench-bracing-selection == "h-cols & wd plank col.ftg")
  then (quantity-id = 24102 ,
        units = "CUYD" ,
        amount =   number *
           (2 * (length +2) + 2 * (width + 2)
              * (grace-elevation - top-elevation + footing-thickness)) ,
        make-quantity(quantity-id,units,amount)
       )
)

;;; Rules for backfill
;;;
(take-off-rule  concrete-footing-backfill-selectionl
  if (    backfill-selection == "earth mach. w/ hand tamp"
      and rock-excavation-selection == "none required" )
  then (quantity-id = 22204 ,
        units = "CUYD" ,
        amount =   number *
          ((length + 1.5  + (grade-elevation - top-elevation
                                              + footing-thickness) / 3)
           * (width +1.5  + (grade-elevation - top-elevation
                                               + footing-thickness) / 3)
           * (grade-elevation - top-elevation + footing-thickness)
           - (length * width * footing-thickness)
           - length * wall-thickness * (grade-elevation - top-elevation))
          / 27.0 ,
        make-quantity(quantity-id,units,amount) ,
        quantity-id = 222041 ,
        units = "CUYD" ,
        amount =   number *
          ((length + 1.5  + (grade-elevation - top-elevation
                                              + footing-thickness) / 3)
           * (width +1.5  + (grade-elevation - top-elevation
                                               + footing-thickness) / 3)
           * (grade-elevation - top-elevation + footing-thickness)
           - (length * width * footing-thickness)
           - length * wall-thickness * (grade-elevation - top-elevation))
          / 27.0 ,
        make-quantity(quantity-id,units,amount)
       )
)
```

```
(take-off-rule  concrete-footing-backfill-selection2
  if (    backfill-selection == "earth mach. w/ hand tamp"
      and (   rock-excavation-selection == "machine w/ hand dress-up"
           or rock-excavation-selection == "hand excavation only"
           or rock-excavation-selection == "blasting w/machine & hand dress-up" ))
  then (quantity-id = 22204 ,
        units = "CUYD" ,
        amount =   number *
          ((length + 1.5  + (grade-elevation - top-elevation
                                              + footing-thickness) / 3)
           * (width +1.5  + (grade-elevation - top-elevation
                                               + footing-thickness) / 3)
           * (grade-elevation - top-elevation + footing-thickness)
           - (length * width * footing-thickness)
           - length * wall-thickness * (grade-elevation - top-elevation))
          / 27 ,
        make-quantity(quantity-id,units,amount) ,
        quantity-id = 222041 ,
        units = "CUYD" ,
        amount =   number *
          ((length + 1.5  + (grade-elevation - top-elevation
                                              + footing-thickness) / 3)
           * (width +1.5  + (grade-elevation - top-elevation
                                               + footing-thickness) / 3)
           * (grade-elevation - top-elevation + footing-thickness)
           - (length * width * footing-thickness)
           - length * wall-thickness * (grade-elevation - top-elevation))
          / 27 ,
        make-quantity(quantity-id,units,amount)
       )
)

(take-off-rule  concrete-footing-backfill-selection3
  if (    backfill-selection == "earth borrow mach. w/ hand tamp"
      and rock-excavation-selection == "none required" )
  then (quantity-id = 22217 ,
        units = "CUYD" ,
        amount =   number *
          ((length + 1.5  + (grade-elevation - top-elevation
                                              + footing-thickness) / 3)
           * (width +1.5  + (grade-elevation - top-elevation
                                               + footing-thickness) / 3)
           * (grade-elevation - top-elevation + footing-thickness)
           - (length * width * footing-thickness)
           - length * wall-thickness * (grade-elevation - top-elevation))
          / 27 ,
        make-quantity(quantity-id,units,amount) ,
        quantity-id = 222041 ,
        units = "CUYD" ,
        amount =   number *
          ((length + 1.5  + (grade-elevation - top-elevation
                                              + footing-thickness) / 3)
           * (width + 1.5  + (grade-elevation - top-elevation
                                               + footing-thickness) / 3)
           * (grade-elevation - top-elevation + footing-thickness)
           - (length * width * footing-thickness)
           - length * wall-thickness * (grade-elevation - top-elevation))
          / 27 ,
        make-quantity(quantity-id,units, amount)
       )
)
```

```
(take-off-rule concrete-foot ing-backfill-select ion4
  if (    backfill-selection == "earth borrow mach. w/ hand tamp"
      and (   rock-excavation-selection == "machine w/ hand dress-up"
           or rock-excavation-selection == "hand excavation only"
           or rock-excavation-selection == "blasting w/machine & hand dress-up" ))
  then (quantity-id = 22217 ,
        units = "CUYD" ,
        amount =   number *
          ((length + 1.5  + (grade-elevation - top-elevation
                                            + footing-thickness) / 3)
           * (width +1 . 5  + (grade-elevation - top-elevation
                                             + footing-thickness) / 3)
           * (grade-elevation - top-elevation + footing-thickness)
           - (length * width * footing-thickness)
           - length * wall-thickness *  (grade-elevation - top-elevation))
          / 21 ,
        make-quantity(quantity-id,units,amount) ,
        quantity-id = 222041 ,
        units = "CUYD" ,
        amount =   number *
          ((length + 1.5  + (grade-elevation - top-elevation
                                            + footing-thickness) / 3)
           * (width +1 . 5  + (grade-elevation - top-elevation
                                             + footing-thickness) / 3)
           * (grade-elevation - top-elevation + footing-thickness)
           - (length * width * footing-thickness)
           - length * wall-thickness *  (grade-elevation - top-elevation))
          / 27 ,
        make-quantity(quantity-id,units,amount)
       )
)

(take-off-rule concrete-footing-backfill-selection5
  if (    backfill-selection == "earth hand only"
      and rock-excavation-selection == "none required" )
  then (quantity-id = 22206 ,
        units = "CUYD" ,
        amount =   number *
          ((length + 1.5  + (grade-elevation - top-elevation
                                            + footing-thickness) / 3)
           * (width +1 . 5  + (grade-elevation - top-elevation
                                             + footing-thickness) / 3)
           * (grade-elevation - top-elevation + footing-thickness)
           - (length * width * footing-thickness)
           - length * wall-thickness *  (grade-elevation - top-elevation))
          / 27 ,
        make-quantity(quantity-id,units,amount) ,
        quantity-id = 222041 ,
        units = "CUYD" ,
        amount =   number *
          ((length + 1.5  + (grade-elevation - top-elevation
                                            + footing-thickness) / 3)
           * (width +1 . 5  + (grade-elevation - top-elevation
                                             + footing-thickness) / 3)
           * (grade-elevation - top-elevation + footing-thickness)
           - (length * width * footing-thickness)
           - length * wall-thickness *  (grade-elevation - top-elevation))
          / 27 ,
        make-quantity(quantity-id,units,amount) ,
       )
)
```

```
(take-off-rule  concrete-footing-backfill-selection6
  if (    backfill-selection == "earth hand only"
          and (rock-excavation-selection == "machine w/ hand dress-up"
               or rock-excavation-selection == "hand excavation only"
               or rock-excavation-selection ==
                                "blasting w/machine & hand dress-up" ))
  then (quantity-id = 22206 ,
        units = "CUYD" ,
        amount =   number *
          ((length + 1.5  + (grade-elevation - top-elevation
                                              + footing-thickness) / 3)
           *  (width +1.5  + (grade-elevation - top-elevation
                                               + footing-thickness) / 3)
           *  (grade-elevation - top-elevation + footing-thickness)
           -  (length * width * footing-thickness)
           -  length * wall-thickness *  (grade-elevation - top-elevation))
          / 27 ,
        make-quantity(quantity-id,units,amount) ,
        quantity-id = 222041 ,
        units = "CUYD" ,
        amount =   number *
          ((length + 1.5  + (grade-elevation - top-elevation
                                              + footing-thickness) / 3)
           *  (width +1.5  + (grade-elevation - top-elevation
                                               + footing-thickness) / 3)
           *  (grade-elevation - top-elevation + footing-thickness)
           -  (length * width * footing-thickness)
           -  length * wall-thickness *  (grade-elevation - top-elevation))
          / 27 ,
        make-quantity(quantity-id,units,amount)
       )
)

(take-off-rule  concrete-footing-backfill-select ion7
  if (    backfill-selection == "granular borrow mach. w/ hand tamp"
      and rock-excavation-selection == "none required" )
  then (quantity-id = 22219 ,
        units = "CUYD" ,
        amount =   number *
          ((length + 1.5  + (grade-elevation - top-elevation
                                              + footing-thickness) / 3)
           *  (width +1.5  + (grade-elevation - top-elevation
                                               + footing-thickness) / 3)
           *  (grade-elevation - top-elevation + footing-thickness)
           -  (length * width * footing-thickness)
           -  length * wall-thickness *  (grade-elevation - top-elevation))
          / 27 ,
        make-quantity(quantity-id,units,amount) ,
        quantity-id = 222041 ,
        units = "CUYD" ,
        amount =   number *
          ((length + 1.5  + (grade-elevation - top-elevation
                                              + footing-thickness) / 3)
           *  (width + 1.5  + (grade-elevation - top-elevation
                                               + footing-thickness) / 3)
           *  (grade-elevation - top-elevation + footing-thickness)
           -  (length * width * footing-thickness)
           -  length * wall-thickness *  (grade-elevation - top-elevation))
          / 27 ,
        make-quantity(quantity-id,units,amount)
       )
)
```

```
(take-off-rule concrete-footing-back.fiIl-selection8
  if (    backfill-selection == "granular borrow mach. w/ hand tamp"
      and (   rock-excavation-selection == "machine w/ hand dress-up"
           or rock-excavation-selection == "hand excavation only"
           or rock-excavation-selection == "blasting w/machine & hand dress-up" ))
  then (quantity-id = 22219 ,
        units = "CUYD" ,
        amount =   number *
           ((length + 1.5  + (grade-elevation - top-elevation
                                               + footing-thickness) / 3)
           * (width +1.5  + (grade-elevation - top-elevation
                                                + footing-thickness) / 3)
           * (grade-elevation - top-elevation + footing-thickness)
           - (length * width * footing-thickness)
           - length * wall-thickness *  (grade-elevation - top-elevation))
          / 27 ,
        make-quantity(quantity-id,units,amount) ,
        quantity-id = 222041 ,
        units = "CUYD" ,
        amount =   number *
           ((length +1.5  + (grade-elevation - top-elevation
                                               + footing-thickness) / 3)
           * (width +1.5  + (grade-elevation - top-elevation
                                                + footing-thickness) / 3)
           * (grade-elevation - top-elevation + footing-thickness)
           - (length * width * footing-thickness)
           - length * wall-thickness *  (grade-elevation - top-elevation))
          / 27 ,
        make-quantity(quant ity-id,units,amount)
       )
)

(take-off-rule concrete-footing-backfill-selection9
  if (    backfill-selection == "gravel/stone borrow mach. only"
      and rock-excavation-selection == "none required" )
  then (quantity-id = 22220 ,
        units = "CUYD" ,
        amount =   number *
           ((length + 1.5  + (grade-elevation - top-elevation
                                               + footing-thickness) / 3)
           * (width +1.5  + (grade-elevation - top-elevation
                                                + footing-thickness) / 3)
           * (grade-elevation - top-elevation + footing-thickness)
           - (length * width * footing-thickness)
           - length * wall-thickness *  (grade-elevation - top-elevation))
          / 27 ,
        make-quan :ity (quantity-id, units, amount)
       )
)

(take-off-rule concrete-foot ing-backfill-select ion10
  if (    backfill-selection == "gravel/stone borrow mach. only"
      and (   rock-excavation-selection == "machine w/ hand dress-up"
           or rock-excavation-selection == "hand excavation only"
           or rock-excavation-selection == "blasting w/machine & hand dress-up" ))
  then (quantity-id = 22220 ,
        units = "CUYD" ,
        amount =   number *
           ((length + 1.5  + (grade-elevation - top-elevation
                                               + footing-thickness) / 3)
           * (width +1.5  + (grade-elevation - top-elevation
                                                + footing-thickness) / 3)
           * (grade-elevation - top-elevation + footing-thickness)
           - (length * width * footing-thickness)
           - length * wall-thickness *  (grade-elevation - top-elevation))
          / 27 ,
        make-quantity(quantity-id,units,amount)
       )
)
```

```
;;; Rules for footing keyway
;;;
(take-off-rule concrete-footing-footing-keyway-selection1
  if (footing-keyway-selection == "2x4 ftg key")
  then (quantity-id = 31011 ,
        units = "LNFT" ,
        amount =  number * (length) ,
        make-quantity(quantity-id,units,amount)
       )
)
(take-off-rule concrete-footing-footing-keyway-selection2
  if (footing-keyway-selection == "split key w/waterstop")
  then (quantity-id = 31011 ,
        units = "LNFT" ,
        amount =  number * (length) ,
        make-quantity(quantity-id,units,amount) ,
        quantity-id = 31012 ,
        units = "LNFT" ,
        amount =  number * (length) ,
        make-quantity(quantity-id,units, amount)
       )
)
(take-off-rule concrete-footing-footing-keyway-selection3
  if (footing-keyway-selection == "4x4 key or 2x8 key")
  then (quantity-id = 31011 ,
        units = "LNFT" ,
        amount =  number * (length) ,
        make-quantity(quantity-id,units,amount) ,
        quantity-id = 31019 ,
        units = "LNFT" ,
        amount =  number * (0.67 * length) ,
        make-quantity(quantity-id,units,amount)
       )
)
(take-off-rule concrete-footing-footing-keyway-selection4
  if (footing-keyway-selection == "4x6 key or 3x8 key")
  then (quantity-id = 31011 ,
        units = "LNFT" ,
        amount =  number * (length) ,
        make-quantity(quantity-id,units,amount) ,
        quantity-id = 31019 ,
        units = "LNFT" ,
        amount =  number * (1.33 * length) ,
        make-quantity(quantity-id,units,amount)
       )
)
(take-off-rule concrete-footing-footing-keyway-selection5
  if (footing-keyway-selection == "4x12 key")
  then (quantity-id = 31011 ,
        units = "LNFT" ,
        amount =  number * (length) ,
        make-quantity(quantity-id,units,amount) ,
        quantity-id = 31019 ,
        units = "LNFT" ,
        amount =  number * (3.33 * length) ,
        make-quantity(quantity-id,units,amount)
       )
)
```

```
;;; Rules for reinforcing steel
;;;
(take-off-rule  concrete-footing-reinforcing-steel-selection1
  if (    reinforcing-steel-selection == "column footings lbs/cuyd calc"
      and (length * width * footing-thickness / 27.0) <= 1.0 )
  then (quantity-id = 32101 ,
        units = "TON" ,
        amount =   number *
                 ((length * width * footing-thickness / 27.0) * 85.0) / 1000.0 ,
        make-quantity(quantity-id,units,amount)
       )
)

(take-off-rule  concrete-footing-reinforcing-steel-selection2
  if (    reinforcing-steel-selection == "column footings lbs/cuyd calc"
      and (length * width * footing-thickness / 27.0) > 1.0
      and (length * width * footing-thickness / 27.0) <= 5.0 )
  then (quantity-id = 32101 ,
        units = "TON" ,
        amount =   number *
                 ((length * width * footing-thickness / 27.0) * 90.0) / 1000.0 ,
        make-quantity(quantity-id,units,amount)
       )
)

(take-off-rule  concrete-foot ing-reinforeing-steel-selection3
  if (    reinforcing-steel-selection == "column footings lbs/cuyd calc"
      and (length * width * footing-thickness / 27.0) > 5.0 )
  then (quantity-id = 32101 ,
        units = "TON" ,
        amount =   number *
                 ((length * width * footing-thickness / 27.0) * 100.0) / 1000.0 ,
        make-quantity(quantity-id,units,amount)
       )
)

(take-off-rule  concrete-footing-reinforeing-steel-selection4
  if (reinforcing-steel-selection == "wall footing 50 lfb/cuyd")
  then (quantity-id = 32101 ,
        units = "TON" ,
        amount =   number *
                 ((length * width * footing-thickness / 27.0) * 50.0) / 1000.0 ,
        make-quantity(quantity-id,units,amount)
       )
)

(take-off-rule  concrete-foot ing-reinforeing-steel-select ion-sizes
  if (   reinforcing-steel-selection == "bars by pieces both ways"
      or reinforcing-steel-selection == "bars by spacing both ways"
      or reinforcing-steel-selection == "long bar by piece short bar by spacing"
      or reinforcing-steel-selection == "long bar by spacing short bar by pieces"
      or reinforcing-steel-selection == "lbs/lnft wall ftg with j dim"
      or reinforcing-steel-selection == "lbs/footing column footing with j dim" )
  then (long-bar-diameter  = hicost-rebar-diameter(long-bar-size) ,
        long-bar-weight    = hicost-rebar-weight(long-bar-size) ,
        short-bar-diameter = hicost-rebar-diameter(short-bar-size) ,
        short-bar-weight    = hicost-rebar-weight(short-bar-size)
       )
)
```

```
(take-off-rule  concrete-footing-reinforcing-steel-selection5
  if  (reinforcing-steel-selection == "bars by pieces both ways")
  then  (quantity-id = 32101 ,
        units = "TON" ,
        amount =   number
                 * ((length + (length / 20.0) * (24 * long-bar-size / 12.0))
                    * long-bar-pieces * long-bar-weight) / 1000.0 ,
        make-quantity(quantity-id,units,amount) ,
        quantity-id = 32101 ,
        units = "TON" ,
        amount =   number
                 * ((width + (width / 20.0) * (24 * short-bar-size / 12.0))
                    * short-bar-pieces * short-bar-weight) / 1000.0 ,
        make-quantity(quantity-id,units,amount)
       )
)

(take-off-rule  concrete-foot ing-reinforcing-steel-selection6
  if  (reinforcing-steel-selection == "bars by spacing both ways")
  then  (quantity-id = 32101 ,
        units = "TON" ,
        amount =   number
                 * ((length + (length / 20.0) * (24 * long-bar-size / 12.0))
                    * (width / (long-bar-size / 12.0) + 1) * long-bar-weight)
                 / 1000.0 ,
        make-quantity(quantity-id,units,amount) ,
        quantity-id = 32101 ,
        units = "TON" ,
        amount =   number
                 * ((width + (width / 20.0) * (24 * short-bar-size / 12.0))
                    * (length / (short-bar-size / 12.0) + 1) * short-bar-weight)
                 / 1000.0 ,
        make-quantity(quantity-id,units, amount)
       )
)

(take-off-rule  concrete-foot ing-reinforeing-steel-selection?
  if  (reinforcing-steel-selection == "long bar by piece short bar by spacing")
  then  (quantity-id = 32101 ,
        units = "TON" ,
        amount =   number
                 * ((length + (length / 20.0) * (24 * long-bar-size / 12.0))
                    * long-bar-pieces * long-bar-weight) / 1000.0 ,
        make-quantity(quantity-id,units,amount) ,
        quantity-id = 32101 ,
        units = "TON" ,
        amount =   number
                 * ((width + (width / 20.0) * (24 * short-bar-size / 12.0))
                    * (length / (short-bar-size / 12.0) + 1) * short-bar-weight)
                 / 1000.0 ,
        make-quantity(quantity-id,units,amount)
       )
)

(take-off-rule  concrete-footing-reinforeing-steel-selection8
  if  (reinforcing-steel-selection == "long bar by spacing short bar by pieces")
  then  (quantity-id = 32101 ,
        units = "TON" ,
        amount =   number
                 * ((length + (length / 20.0) * (24 * long-bar-size / 12.0))
                    * (width / (long-bar-size / 12.0) +1 ) * long-bar-weight)
                 / 1000.0 ,
        make-quantity(quantity-id,units,amount) ,
        quantity-id = 32101 ,
        units = "TON" ,
        amount =   number
                 * ((width + (width / 20.0) * (24 * short-bar-size / 12.0))
                    * short-bar-pieces * short-bar-weight) / 1000.0 ,
        make-quantity(quantity-id,units,amount)
       )
)
```

```
(take-off-rule  concrete-foot ing-reinforcing-steel-selection9
  if  (reinforcing-steel-selection == "lbs/lnft  wall  ftg  with  j  dim")
  then  (quantity-id = 32101  ,
         units = "TON"  ,
         amount =   number *  (length *  long-bar-lbs-lnft)  /  1000.0  ,
         make-quantity(quantity-id,units,amount)
        )
)

(take-off-rule  concrete-foot ing-reinforcing-steel-select ion10
  if  (reinforcing-steel-selection == "lbs/footing  column  footing  with  j  dim")
  then  (quantity-id = 32101  ,
         units = "TON"  ,
         amount =   number *  (length *  long-bar-lbs-ftg)  /  1000.0  ,
         make-quantity(quantity-id,units,amount)
        )
)

;;; Rules for anchor bolts and base plates
;;;
(take-off-rule  concrete-foot ing-anchor-boIts-and-base-plates-select ion1
  if  (anchor-bolts-and-base-plates-selection == "set  2ab"  )
  then  (quantity-id = 33301  ,
         units = "PCS"  ,
         amount =   number *  (2)  ,
         make-quantity(quant ity-id,units,amount)  ,
         quantity-id = 33302  ,
         units = "PCS"  ,
         amount =   number *  (2)  ,
         make-quantity(quantity-id,units,amount)
        )
)

(take-off-rule  concrete-footing-anchor-boIts-and-base-plates-selection2
  if  (anchor-bolts-and-base-plates-selection == "set  4ab"  )
  then  (quantity-id = 33301  ,
         units = "PCS"  ,
         amount =   number *  (4)  ,
         make-quantity(quantity-id,units, amount) ,
         quantity-id = 33302  ,
         units = "PCS"  ,
         amount =   number *  (4)  ,
         make-quantity(quantity-id,units,amount)
        )
)

(take-off-rule  concrete-foot ing-anchor-boIts-and-base-plates-selection3
  if  (anchor-beIts-and-base-plates-selection == "set  2ab  grout  base  pit")
  then  (quantity-id = 33301  ,
         units = "PCS"  ,
         amount =   number *  (2)  ,
         make-quantity(quantity-id,units,amount)  ,
         quantity-id = 33302  ,
         units = "PCS"  ,
         amount =   number *  (2)  ,
         make-quantity(quantity-id,units,amount)  ,
         quantity-id = 33303  ,
         units = "PCS"  ,
         amount =   number  ,
         make-quantity(quantity-id,units,amount)
        )
)
```

```
(take-off-rule  concrete-foot ing-anchor-bolts-and-base-plates-selection4
  if  (anchor-bolts-and-base-plates-selection == "set  4ab  grout  base  pit")
  then  (quantity-id = 33301  ,
        units = "PCS"  ,
        amount =   number *  (4)  /
        make-quantity(quantity-id,units,amount)  ,
        quantity-id = 33302  ,
        units = "PCS"  ,
        amount =   number *  (4)  ,
        make-quantity(quantity-id,units,amount)  ,
        quantity-id = 33303  ,
        units = "PCS"  ,
        amount =   number  ,
        make-quantity(quantity-id,units, amount)
       )
)
(take-off-rule  concrete-footing-anchor-boIts-and-base-plates-selection5
  if  (anchor-bolts-and-base-plates-selection == "set  2ab  set  and  grout  base  pit")
  then  (quantity-id = 33301  ,
        units = "PCS"  ,
        amount =   number *  (2)  ,
        make-quantity(quantity-id,units,amount)  ,
        quantity-id = 33302  ,
        units = "PCS"  ,
        amount =   number *  (2)  ,
        make-quantity(quantity-id,units,amount)  ,
        quantity-id = 33303  ,
        units = "PCS"  ,
        amount =   number  ,
        make-quantity(quantity-id,units,amount)
       )
)
(take-off-rule  concrete-footing-anchor-bolts-and-base-plates-selection6
  if  (anchor-bolts-and-base-plates-selection == "set  4ab  set  and  grout  base  pit")
  then  (quantity-id = 33301  ,
        units = "PCS"  ,
        amount =   number *  (4)  ,
        make-quantity(quantity-id,units,amount)  ,
        quantity-id = 33302  ,
        units = "PCS"  ,
        amount =   number *  (4)  ,
        make-quantity(quantity-id,units,amount)  ,
        quantity-id = 33303  ,
        units = "PCS"  ,
        amount =   number  ,
        make-quantity(quantity-id,units,amount)
       )
)
(take-off-rule  concrete-foot ing-anchor-boIts-and-base-plates-selection7
  if  (anchor-bolts-and-base-plates-selection == "set  &  grout  base  plate")
  then  (quantity-id = 33303  ,
        units = "PCS"  ,
        amount =   number  ,
        make-quantity(quantity-id,units,amount)
       )
)
```

7.3.2. Retaining Wall Takeoff Rule Set

```
;;; Rules for quality of concrete
;;;
(take-off-rule retaining-wall-quality-of-concrete-amount 1
  if (    (   wall-forms-selection ==  "none required"
           or wall-forms-selection == "both sides vertical"
           or wall-forms-selection == "one side battered")
      and concrete-placement-selection != "stepped wall ftg direct")
  then (amount =   number * ( ( (length * height
                                        * (top-thickness + base-thickness) / 2.0
                                        + length * width * footing-thickness)
                                / 27.0)))
)
(take-off-rule retaining-wall-quality-of-concrete-amount2
  if (    (   wall-forms-selection ==  "none required"
           or wall-forms-selection == "both sides vertical"
           or wall-forms-selection == "one side battered" )
      and concrete-placement-selection == "stepped wall ftg direct" )
  then (amount =   number * ( ( (length * height
                                        * (top-thickness + base-thickness) / 2.0
                                        + length * width * footing-thickness)
                                / 27.0)
                              + (width * stepped-footing-height
                                       * (1.0 + stepped-footing-height / 2.0))))
)
(take-off-rule retaining-wa11-qua1ity-of-concrete-amount3
  if (    (   wall-forms-selection == "add ledge"
           or wall-forms-selection == "add ledge & brick facing")
      and concrete-placement-selection != "stepped wall ftg direct" )
  then (amount =   number * ( ( (length * height
                                        * (top-thickness + base-thickness) / 2.0
                                        + length * width * footing-thickness)
                                / 27.0)
                              - (length * height * top-thickness) / 27.0))
)
(take-off-rule retaining-wall-quality-of-concrete-amount4
  if (    (   wall-forms-selection == "add ledge"
           or wall-forms-selection == "add ledge & brick facing")
      and concrete-placement-selection == "stepped wall ftg direct" )
  then (amount =   number * ( ( (length * height
                                        * (top-thickness + base-thickness) / 2.0
                                        + length * width * footing-thickness)
                                / 27.0)
                              - (length * height * top-thickness) / 27.0
                              + (width * stepped-footing-height
                                       * (1.0 + stepped-footing-height / 2.0))))
)
(take-off-rule retaining-wall-quality-of-concreteO
  if (    quality-of-concrete == "3000 psi standard mix")
  then (quantity-id = 33100 ,
        units = "CUYD" /
        make-quantity(quantity-id,units,amount)
       )
)
(take-off-rule retaining-wall-quality-of-concretel
  if (    quality-of-concrete == "deduct for 2500 psi")
  then (quantity-id = 33101 ,
        units = "CUYD" ,
        make-quantity(quantity-id,units,amount)
       )
)
```

```
(take-off-rule  retaining-wall-quality-of-concretel
  if (    quality-of-concrete == "deduct for 2500 psi")
  then (quantity-id = 33101 ,
        units = "CUYD" ,
        make-quantity(quantity-id,units,amount)
       )
)

(take-off-rule  retaining-wall-quality-of-concrete2
  if (    quality-of-concrete == "add for 3500 psi")
  then (quantity-id = 33102 ,
        units = "CUYD" ,
        make-quantity(quantity-id,units,amount)
       )
)

(take-off-rule  retaining-wall-quality-of-concrete3
  if (    quality-of-concrete == "add for 4000 psi")
  then (quantity-id = 33103 ,
        units = "CUYD" ,
        make-quantity(quantity-id,units,amount)
       )
)

;;; Rules for concrete placement
;;;
(take-off-rule  retaining-wall-concrete-placement-selection-amount1
  if (   wall-forms-selection ==  "none required"
      or wall-forms-selection == "both sides vertical"
      or wall-forms-selection == "one side battered" )
  then (wall-amount =   number * (  (  (length * height
                                           * (top-thickness + base-thickness) / 2.0)
                                      / 27.0)))
)

(take-off-rule  retaining-wall-concrete-placement-selection-amount2
  if (   wall-forms-selection == "add ledge"
      or wall-forms-selection == "add ledge & brick facing" )
  then (wall-amount =   number * (  (  (length * height
                                              *  (top-thickness + base-thickness)
                                              /  2.0)
                                      / 27.0)
                                    - (length * height * top-thickness) / 27.0))
)

(take-off-rule  retaining-wall-concrete-placement-selection1
  if (concrete-placement-selection == "ftg & wall direct")
  then (quantity-id = 26616 ,
        units = "CUYD" ,
        amount = wall-amount ,
        make-quantity(quantity-id,units,amount) ,
        quantity-id = 26610 ,
        units = "CUYD" ,
        amount =   number * ( length * width * footing-thickness) ,
        make-quantity(quantity-id,units,amount)
       )
)

(take-off-rule  retaining-wall-concrete-placement-selection2
  if (concrete-placement-selection == "ftg direct wall w/carts")
  then (quantity-id = 26616 ,
        units = "CUYD" ,
        amount = wall-amount ,
        make-quantity(quantity-id,units,amount) ,
        quantity-id = 26617 ,   ; wall with carts
        units = "CUYD" ,
        amount = wall-amount ,
        make-quantity(quantity-id,units,amount) ,
        quantity-id = 26610 ,
        units = "CUYD" ,
        amount =   number * ( length * width * footing-thickness) ,
        make-quantity(quantity-id,units,amount)
       )
)
```

```
(take-off-rule  retaining-wall-concrete-placement-selection3
  if  (concrete-placement-selection == "ftg direct  wall  - pump")
  then  (quantity-id = 26618 ,
        units = "CUYD" ,
        amount = wall-amount  ,
        make-quantity(quantity-id,units,amount)  ,
        quantity-id = 26610  ,
        units = "CUYD" ,
        amount =   number *  (  length * width * footing-thickness) ,
        make-quantity(quantity-id,units,amount)
       )
)
(take-off-rule  retaining-wall-concrete-placement-selection4
  if  (concrete-placement-selection == "ftg direct  wall w/crane")
  then  (quantity-id = 26619 ,
        units - "CUYD" ,
        amount = wall-amount  ,
        make-quantity(quantity-id,units,amount)  ,
        quantity-id = 26610  ,
        units = "CUYD" ,
        amount =   number *  (  length * width * footing-thickness) ,
        make-quantity(quantity-id,units,amount)
       )
)
(take-off-rule  retaining-wall-concrete-placement-selection5
  if  (concrete-placement-selection == "ftg & wall w/carts")
  then  (quantity-id = 26616 ,
        units = "CUYD" ,
        amount = wall-amount  ,
        make-quantity(quantity-id,units,amount)  ,
        quantity-id = 26617  ,    ; wall with carts
        units = "CUYD" ,
        amount = wall-amount  ,
        make-quantity(quantity-id,units,amount)  ,
        quantity-id = 26610  ,
        units = "CUYD" ,
        amount =   number *  (  length * width * footing-thickness) ,
        make-quantity(quantity-id,units,amount),
        quantity-id = 26611  ,    ; footing with carts
        units = "CUYD" ,
        amount =   number *  (  length * width * footing-thickness) ,
        make-quantity(quantity-id,units,amount)
       )
)
(take-off-rule  retaining-wall-concrete-placement-selection6
  if  (concrete-placement-selection == "ftg w/carts wall w/pump")
  then  (quantity-id = 26618 ,
        units = "CUYD" ,
        amount = wall-amount  ,
        make-quantity(quantity-id,units,amount)  ,
        quantity-id = 26610  ,
        units = "CUYD" ,
        amount =   number *  (  length * width * footing-thickness) ,
        make-quantity(quantity-id,units,amount)  ,
        quantity-id = 26611  ,    ; footing with carts
        units = "CUYD" ,
        amount =   number *  (  length * width * footing-thickness) ,
        make-quantity(quantity-id,units,amount)
       )
)
```

```
(take-off-rule  retaining-wall-concrete-placement-selection7
  if  (concrete-placement-selection == "ftg w/carts wall w/crane")
  then  (quantity-id = 26619 ,
        units = "CUYD" ,
        amount = wall-amount ,
        make-quantity(quantity-id,units,amount) ,
        quantity-id = 26610 ,
        units = "CUYD" ,
        amount =   number *  ( length * width * footing-thickness) ,
        make-quantity(quantity-id,units,amount) ,
        quantity-id = 26611 ,    ; footing with carts
        units = "CUYD" ,
        amount =   number *  ( length * width * footing-thickness) ,
        make-quantity(quantity-id,units,amount)
       )
)

(take-off-rule  retaining-wall-concrete-placement-selection8
  if  (concrete-placement-selection == "ftg & wall w/pump")
  then  (quantity-id = 26618 ,
        units = "CUYD" ,
        amount = wall-amount ,
        make-quantity(quantity-id,units,amount) ,
        quantity-id = 26612 ,
        units = "CUYD" ,
        amount =   number *  ( length * width * footing-thickness) ,
        make-quantity(quantity-id,units,amount)
       )
)

(take-off-rule  retaining-wall-concrete-placement-selection9
  if  (concrete-placement-selection == "stepped wall  ftg direct")
  then  (quantity-id = 26610 ,
        units = "CUYD" ,
        amount =   number *  (width *  stepped-footing-height
                                    *  (1.0 + stepped-footing-height / 2.0))
                           / 27.0 ,
        make-quantity(quantity-id,units,amount)
       )
)

;;; Rules for footing forms and finish
;;;
(take-off-rule  retaining-wall-foot ing-form-and-finish-selectionl
  if  (footing-form-and-finish-selection == "in earth")
  then  (quantity-id = 26630 ,
        units = "SQFT" ,
        amount =   number *  (2 * length * footing-thickness) ,
        make-quantity(quantity-id,units,amount) ,
        quantity-id = 26623 ,
        units = "SQFT" ,
        amount =   number *  (length * width)  ,
        make-quantity(quantity-id,units,amount)
       )
)

(take-off-rule  retaining-wall-footing-form-and-finish-selection2
  if  (footing-form-and-finish-selection == "in rock")
  then  (quantity-id = 26631 ,
        units = "SQFT" ,
        amount =   number *  (2 * length * footing-thickness) ,
        make-quantity(quantity-id,units,amount) ,
        quantity-id = 26623 ,
        units = "SQFT" ,
        amount =   number *  (length * width)  ,
        make-quantity(quantity-id,units,amount)
       )
)
```

```
(take-off-rule  retaining-wall-footing-form-and-finish-selection3
  if  (footing-form-and-finish-selection == "stepped wall  ftg  forms")
  then  (quantity-id = 26630  ,
        units = "SQFT"  ,
        amount =   number *  (2  *  footing-thickness
                                *  (1.0 + footing-thickness / 2.0))  ,
        make-quantity(quantity-id,units, amount)  ,
        quantity-id = 26632  ,
        units = "SQFT"  ,
        amount =   number *  (length * width)  ,
        make-quantity(quantity-id,units,amount)
       )
)

(take-off-rule  retaining-wall-footing-form-and-finish-selection4
  if  (footing-form-and-finish-selection ==  "ftg  bulkhead")
  then  (quantity-id = 26632  ,
        units = "SQFT"  ,
        amount =   number *  (length * width)  ,
        make-quantity(quantity-id,units,amount)
       )
)

;;; Rules for wall forms
;;;
(take-off-rule  retaining-wall-wall-forms-selection1
  if  (wall-forms-selection == "both  sides  vertical")
  then  (quantity-id = 26640  ,
        units = "SQFT"  ,
        amount =   number *  (2 * length * height)  ,
        make-quantity(quantity-id,units,amount)  ,
        quantity-id = 26621  ,
        units = "SQFT"  ,
        amount =   number *  (2 * length * height)  ,
        make-quantity(quantity-id,units,amount)
       )
)

(take-off-rule  retaining-wall-wall-forms-selection2
  if  (wall-forms-selection == "one  side  battered")
  then  (quantity-id = 26640  ,
        units = "SQFT"  ,
        amount =   number *  (2 * length * height)  ,
        make-quantity(quantity-id,units,amount)  ,
        quantity-id = 26651  ,
        units = "SQFT"  ,
        amount =   number *  (length * height)  ,
        make-quantity(quantity-id,units,amount)  ,
        quantity-id = 26621  ,
        units = "SQFT"  ,
        amount =   number *  (2 * length * height)  ,
        make-quantity(quantity-id,units,amount)
       )
)

(take-off-rule  retaining-wall-wall-forms-selection3
  if  (wall-forms-selection == "add  ledge")
  then  (quantity-id = 26652  ,
        units = "SQFT"  ,
        amount =   number *  (length * height)  ,
        make-quantity(quantity-id,units,amount)  ,
        quantity-id = 26660  ,    ; dove-anchors
        units = "LNFT"  ,
        amount =   number *  (0.75 * length * height)  ,
        make-quantity(quantity-id,units,amount)
       )
)
```

```
(take-off-rule  retaining-wall-wall-forms-selection4
  if  (wall-forms-selection == "add ledge & brick facing")
  then  (quantity-id = 26652  ,
        units = "SQFT"  ,
        amount =   number *  (length * height)  ,
        make-quantity(quantity-id,units, amount)  ,
        quantity-id = 26660  ,    ; dovetail anchors
        units = "SQFT"  ,
        amount =   number *  (0.75 * length * height)  ,
        make-quantity(quantity-id,units,amount)  ,
        quantity-id = 42100  ,    ; face brick-std.
        units = "SQFT"  ,
        amount =   number *  (length * height)  ,
        make-quantity(quantity-id,units,amount)  ,
        quantity-id = 41901  ,    ; clean face brick
        units = "SQFT"  ,
        amount =   number *  (length * height)  ,
        make-quantity(quantity-id,units,amount)
       )
)

;;; Rules for excavation
;;;
(take-off-rule  retaining-wall-excavationl
  if  (    excavation-selection == "earth machine w/hand dress"
      and footing-depth <= 3  )
  then  (quantity-id = 222033  ,
        units = "CUYD"  ,
        amount =   number
                  *  (length *  (width + 2 + (footing-depth + footing-thickness)  / 3.0)
                            *  (footing-depth + footing-thickness)
                            +  length / 2.0
                                      *  (width + 2
                                               + 2 *  (footing-depth + footing-thickness)
                                                   / 3.0)
                                      *  (height  -  footing-depth))
                  / 27.0  ,
        make-quantity(quantity-id,units,amount)  ,
        quantity-id = 22205  ,
        units = "CUYD"  ,
        amount =   number *  (length *  (width + 2)  * 0.2)  / 27.0  ,
        make-quantity(quantity-id,units,amount)
       )
)

(take-off-rule  retaining-wall-excavation2
  if  (    excavation-selection == "earth machine w/hand dress"
      and footing-depth > 3
      and footing-depth <= 4  )
  then  (quantity-id = 222034  ,
        units = "CUYD"  ,
        amount =   number
                  *  (length *  (width + 2 + (footing-depth + footing-thickness)  / 3.0)
                            *  (footing-depth + footing-thickness)
                            +  length / 2.0
                                      *  (width + 2
                                               + 2 *  (footing-depth + footing-thickness)
                                                   / 3.0)
                                      *  (height  -  footing-depth))
                  / 27.0  ,
        make-quantity(quantity-id,units,amount)  ,
        quantity-id = 22205  ,
        units = "CUYD"  ,
        amount =   number *  (length *  (width + 2)  * 0.2)  / 27.0  ,
        make-quantity(quantity-id,units,amount)
       )
)
```

```
(take-off-rule  retaining-wall-excavation3
  if (    excavation-selection == "earth machine w/hand dress"
      and footing-depth > 4
      and footing-depth <= 5 )
  then (quantity-id = 222035 ,
        units = "CUYD" ,
        amount =   number
                 * (length * (width + 2 + (footing-depth + footing-thickness) / 3.0)
                           * (footing-depth + footing-thickness)
                           + length / 2.0
                                     * (width + 2
                                              + 2 * (footing-depth + footing-thickness)
                                                  / 3.0)
                                     * (height - footing-depth))
                 / 27.0 ,
        make-quantity(quantity-id,units,amount) ,
        quantity-id = 22205 ,
        units = "CUYD" ,
        amount =   number * (length * (width + 2) * 0.2) / 27.0 ,
        make-quantity(quant ity-id,units,amount)
       )
)

(take-off-rule  retaining-wall-excavation4
  if (    excavation-selection == "earth machine w/hand dress"
      and footing-depth > 5
      and footing-depth <= 6 )
  then (quantity-id = 222036 ,
        units = "CUYD" ,
        amount =   number
                 * (length * (width + 2 + (footing-depth + footing-thickness) / 3.0)
                           * (footing-depth + footing-thickness)
                           + length / 2.0
                                     * (width + 2
                                              + 2 * (footing-depth + footing-thickness)
                                                  / 3.0)
                                     * (height - footing-depth))
                 / 27.0 ,
        make-quantity(quantity-id,units,amount) ,
        quantity-id = 22205 ,
        units = "CUYD" ,
        amount =   number * (length * (width + 2) * 0.2) / 27.0 ,
        make-quantity(quantity-id,units,amount)
       )
)

(take-off-rule  retaining-wall-excavation5
  if (    excavation-selection == '"earth hand" )
  then (quantity-id = 22205 ,
        units = "CUYD" ,
        amount =   number
                 * (length * (width + 2 + (footing-depth + footing-thickness) / 3.0)
                           * (footing-depth + footing-thickness)
                           + length / 2.0
                                     * (width + 2
                                              + 2 * (footing-depth + footing-thickness)
                                                  / 3.0)
                                     * (height - footing-depth))
                 / 27.0 ,
        make-quantity(quantity-id,units,amount)
       )
)
```

```
(take-off-rule  retaining-wall-excavation6
  if (    excavation-selection == "rock machine w/hand dress"
      and footing-depth <= 3 )
  then (quantity-id = 222033 ,
        units = "CUYD" ,
        amount =   number
                  * (length *  (width + 2 + (footing-depth + footing-thickness) / 3.0)
                             *  (footing-depth + footing-thickness)
                             + length / 2.0
                                      * (width + 2
                                                + 2 * (footing-depth + footing-thickness)
                                                    / 3.0)
                                      * (height - footing-depth))
                  / 27.0 ,
        make-quantity(quantity-id,units,amount) ,
        quantity-id = 22207 ,
        units = "CUYD" ,
        amount =   number * (length * (width +2 ) * 0.2) / 27.0 ,
        make-quantity(quantity-id,units,amount)
       )
)

(take-off-rule  retaining-wall-excavation7
  if (    excavation-selection == "rock machine w/hand dress"
      and footing-depth > 3
      and footing-depth <= 4 )
  then (quantity-id = 222034 ,
        units = "CUYD" ,
        amount =   number
                  * (length *  (width + 2 + (footing-depth + footing-thickness) / 3.0)
                             *  (footing-depth + footing-thickness)
                             + length / 2.0
                                      * (width + 2
                                                + 2 * (footing-depth + footing-thickness)
                                                    / 3.0)
                                      * (height - footing-depth))
                  / 27.0 ,
        make-quantity(quantity-id,units,amount) ,
        quantity-id = 22207 ,
        units = "CUYD" ,
        amount =   number * (length * (width + 2) * 0.2) / 27.0 ,
        make-quantity(quantity-id,units,amount)
       )
)

(take-off-rule  retaining-wall-excavation8
  if (    excavation-selection == "rock machine w/hand dress"
      and footing-depth > 4
      and footing-depth <= 5 )
  then (quantity-id = 222035 ,
        units = "CUYD" ,
        amount =   number
                  * (length *  (width + 2 + (footing-depth + footing-thickness) / 3.0)
                             *  (footing-depth + footing-thickness)
                             + length / 2.0
                                      * (width + 2
                                                + 2 * (footing-depth + footing-thickness)
                                                    / 3.0)
                                      * (height - footing-depth))
                  / 27.0 ,
        make-quantity(quantity-id,units,amount) ,
        quantity-id = 22207 ,
        units = "CUYD" ,
        amount =   number * (length * (width + 2) * 0.2) / 27.0 ,
        make-quantity(quantity-id,units,amount)
       )
)
```

```
(take-off-rule  retaining-wa11-excavation9
  if (    excavation-selection == "rock machine w/hand dress"
      and footing-depth > 5
      and footing-depth <= 6 )
  then (quantity-id = 222036 ,
        units = "CUYD" ,
        amount =   number
                  * (length * (width + 2 + (footing-depth + footing-thickness) / 3.0)
                            * (footing-depth + footing-thickness)
                            + length / 2.0
                                      * (width + 2
                                               + 2 * (footing-depth + footing-thickness)
                                                   / 3.0)
                                      * (height - footing-depth))
                  / 27.0 ,
        make-quantity(quantity-id,units,amount) ,
        quantity-id = 22207 ,
        units = "CUYD" ,
        amount =   number * (length * (width +2 ) * 0.2) / 27.0 ,
        make-quantity(quantity-id,units,amount)
       )
)

(take-off-rule  retaining-wa11-excavation10
  if (    excavation-selection == "'rock hand" )
  then (quantity-id = 22207 ,
        units = "CUYD" ,
        amount =   number
                  * (length * (width + 2 + (footing-depth + footing-thickness) / 3.0)
                            * (footing-depth + footing-thickness)
                            + length / 2.0
                                      * (width + 2
                                               + 2 * (footing-depth + footing-thickness)
                                                   / 3.0)
                                      * (height - footing-depth))
                  / 27.0 ,
        make-quantity(quantity-id,units,amount)
       )
)

;;; Rules for trench backfill
//)
(take-off-rule  retaining-wal1-trench-backfill-selectionl
  if (trench-backfill-selection == "earth adjacent mach placed hand tamp")
  then (quantity-id = 22204 ,
        units = "CUYD" ,
        amount =   number
                  * (length * footing-toe-width * footing-depth
                     + length * (footing-depth + footing-thickness)
                              * (1 + (footing-depth + footing-thickness) / 6.0))
                  / 27.0 ,
        make-quantity(quantity-id,units,amount) ,
        quantity-id = 222041 ,     ; hand tamp trench backfill
        units = "CUYD" ,
        amount =   number
                  * (length * footing-toe-width * footing-depth
                     + length * (footing-depth + footing-thickness)
                              * (1 + (footing-depth + footing-thickness) / 6.0))
                  / 27.0 ,
        make-quantity(quantity-id,units,amount)
       )
)
```

```
(take-off-rule  retaining-wall-trench-backfill-selection2
  if  (trench-backfill-selection == "earth adjacent hand placed hand tamp")
  then (quantity-id = 22206 ,
        units = "CUYD" ,
        amount =   number
                 *  (length * footing-toe-width * footing-depth
                    + length *  (footing-depth + footing-thickness)
                             *  (1 + (footing-depth + footing-thickness) / 6.0))
                 / 27.0 ,
        make-quantity(quantity-id,units,amount) ,
        quantity-id = 222041 ,     ; hand tamp trench backfill
        units = "CUYD" ,
        amount =   number
                 *  (length * footing-toe-width * footing-depth
                    + length *  (footing-depth + footing-thickness)
                             *  (1 + (footing-depth + footing-thickness) / 6.0))
                 / 27.0 ,
        make-quantity(quantity-id,units,amount)
       )
)

(take-off-rule  retaining-wall-trench-backfill-selection3
  if  (trench-backfill-selection == "earth hauled mach placed hand tamp")
  then (quantity-id = 22217 ,
        units = "CUYD" ,
        amount =   number
                 *  (length * footing-toe-width * footing-depth
                    + length *  (footing-depth + footing-thickness)
                             *  (1 + (footing-depth + footing-thickness) / 6.0))
                 / 27.0 ,
        make-quantity(quantity-id,units,amount) ,
        quantity-id = 222041 ,     ; hand tamp trench backfill
        units = "CUYD" ,
        amount =   number
                 *  (length * footing-toe-width * footing-depth
                    + length *  (footing-depth + footing-thickness)
                             *  (1 + (footing-depth + footing-thickness) / 6.0))
                 / 27.0 ,
        make-quantity(quantity-id,units,amount)
       )
)

(take-off-rule  retaining-wall-trench-backfill-selection4
  if  (trench-backfill-selection == "earth hauled hand placed hand tamp")
  then (quantity-id = 22218 ,
        units = "CUYD" ,
        amount =   number
                 *  (length * footing-toe-width * footing-depth
                    + length *  (footing-depth + footing-thickness)
                             *  (1 + (footing-depth + footing-thickness) / 6.0))
                 / 27.0 ,
        make-quantity(quantity-id,units,amount) ,
        quantity-id = 222041 ,     ; hand tamp trench backfill
        units = "CUYD" ,
        amount =   number
                 *  (length * footing-toe-width * footing-depth
                    + length *  (footing-depth + footing-thickness)
                             *  (1 + (footing-depth + footing-thickness) / 6.0))
                 / 27.0 ,
        make-quantity(quantity-id,units,amount)
       )
)
```

```
;;; Rules for retained backfill
;;;
(take-off-rule  retaining-wall-retained-backfill-selectionl
  if {retained-backfill-selection == "earth mach placed mach compact")
  then  (quantity-id = 22221 ,
        units = "CUYD" ,
        amount =   number
                 * (length * height * (width - footing-toe-width
                                             - top-thickness)
                    + length * (height + footing-thickness)
                             * (1 + (height + footing-thickness) / 6.0))
                 / 27.0 ,
        make-quantity(quantity-id,units,amount) ,
        quantity-id = 222211 ,    ; machine tamp trench backfill
        units = "CUYD" ,
        amount =   number
                 * (length * height * (width - footing-toe-width
                                             - top-thickness)
                    + length * (height + footing-thickness)
                             * (1 + (height + footing-thickness) / 6.0))
                 / 27.0 ,
        make-quantity(quantity-id,units,amount)
       )
)
(take-off-rule  retaining-wall-retained-backfill-selection2
  if  (retained-backfill-selection == "earth mach placed hand tamp")
  then  (quantity-id = 22204 ,
        units = "CUYD" ,
        amount =   number
                 * (length * height * (width - footing-toe-width
                                             - top-thickness)
                    + length * (height + footing-thickness)
                             * (1 + (height + footing-thickness) / 6.0))
                 / 27.0 ,
        make-quantity(quantity-id,units,amount) ,
        quantity-id = 222041 ,    ; hand tamp trench backfill
        units = "CUYD" ,
        amount =   number
                 * (length * height * (width - footing-toe-width
                                             - top-thickness)
                    + length * (height + footing-thickness)
                             * (1 + (height + footing-thickness) / 6.0))
                 / 27.0 ,
        make-quantity(quantity-id,units,amount)
       )
)
(take-off-rule  retaining-wall-retained-backfill-selection3
  if  (retained-backfill-selection == "earth borrow mach placed mach compact")
  then  (quantity-id = 22222 ,
        units = "CUYD" ,
        amount =   number
                 * (length * height * (width - footing-toe-width
                                             - top-thickness)
                    + length * (height + footing-thickness)
                             * (1 + (height + footing-thickness) / 6.0))
                 / 27.0 ,
        make-quantity(quantity-id,units,amount) ,
        quantity-id = 222221 ,    ; machine tamp trench backfill
        units = "CUYD" ,
        amount =   number
                 * (length * height * (width - footing-toe-width
                                             - top-thickness)
                    + length * (height + footing-thickness)
                             * (1 + (height + footing-thickness) / 6.0))
                 / 27.0 ,
        make-quantity(quantity-id,units,amount)
       )
)
```

```
(take-off-rule  retaining-wall-retained-backfill-selection4
  if  (retained-backfill-selection == "earth borrow mach placed hand tamp")
  then  (quantity-id = 22222  ,
        units = "CUYD"  ,
        amount =   number
                 *  (length * height *  (width -  footing-toe-width
                                              -  top-thickness)
                    + length *  (height + footing-thickness)
                             *  (1 + (height + footing-thickness)  / 6.0))
                 / 27.0  ,
        make-quantity(quantity-id,units,amount)  ,
        quantity-id = 222041  ,     ; hand tamp trench backfill
        units = "CUYD"  ,
        amount =   number
                 *  (length * height *  (width -  footing-toe-width
                                              -  top-thickness)
                    + length *  (height + footing-thickness)
                             *  (1 + (height + footing-thickness)  / 6.0))
                 / 27.0  ,
        make-quantity(quantity-id,units,amount)
       )
)

(take-off-rule  retaining-wall-retained-backfill-selection5
  if  (retained-backfill-selection == "gravel mach placed")
  then  (quantity-id = 22214  ,
        units = "CUYD"  ,
        amount =   number
                 *  (length * height *  (width -  footing-toe-width
                                              -  top-thickness)
                    + length *  (height + footing-thickness)
                             *  (1 + (height + footing-thickness)  / 6.0))
                 / 27.0  ,
        make-quantity(quantity-id,units,amount)
       )
)

(take-off-rule  retaining-wall-retained-backfill-selection6
  if  (retained-backfill-selection == "bank run mach placed w/hand tamp")
  then  (quantity-id = 22215  ,
        units = "CUYD"  ,
        amount =   number
                 *  (length * height *  (width -  footing-toe-width
                                              -  top-thickness)
                    + length *  (height + footing-thickness)
                             *  (1 + (height + footing-thickness)  / 6.0))
                 / 27.0  ,
        make-quantity(quantity-id,units,amount)  ,
        quantity-id = 222041  ,     ; hand tamp trench backfill
        units = "CUYD"  ,
        amount =   number
                 *  (length * height *  (width -  footing-toe-width
                                              -  top-thickness)
                    + length *  (height + footing-thickness)
                             *  (1 + (height + footing-thickness)  / 6.0))
                 / 27.0  ,
        make-quantity(quantity-id,units,amount)
       )
)

;;;  Rules  for  reinforcing  steel
;;;
(take-off-rule  retaining-wall-reinforcing-steel-selectionl
  if  (reinforcing-steel-selection == "index crsi  level bf-  no  surcharge")
  then  (quantity-id = 32104  ,
        units = "TON"  ,
        amount =   number
                 *  (length * hicost-rw-reinf-steel(height  ,
                                                    reinforcing-steel-selection))
                 / 1000.0  ,
        make-quantity(quantity-id,units,amount)
       )
)
```

```
(take-off-rule  retaining-wa11-reinforcing-steel-selection2
  if  (reinforcing-steel-selection == "index crsi  sloped bf-  no  surcharge")
  then  (quantity-id = 32104  ,
        units = "TON"  ,
        amount =   number
                 *  (length * hicost-rw-reinf-steel(height  ,
                                                    reinforcing-steel-selection))
                 / 1000.0  ,
        make-quantity(quantity-id,units,amount)
       )
)
(take-off-rule  retaining-wall-reinforcing-steel-selection3
  if  (reinforcing-steel-selection == "index crsi  level bf-  hwy  surchg  (light)")
  then  (quantity-id = 32104  ,
        units = "TON"  ,
        amount =   number
                 *  (length * hicost-rw-reinf-steel(height  ,
                                                    reinforcing-steel-selection))
                 / 1000.0  ,
        make-quantity(quantity-id,units,amount)
       )
)
(take-off-rule  retaining-wall-reinforeing-steel-select ion4
  if  (reinforcing-steel-selection == "index crsi  level bf-  railwy surch  (heavy)")
  then  (quantity-id = 32104  ,
        units = "TON"  ,
        amount =   number
                 *  (length * hicost-rw-reinf-steel(height  ,
                                                    reinforcing-steel-selection))
                 / 1000.0  ,
        make-quantity(quantity-id,units,amount)
       )
)
(take-off-rule  retaining-wa11-reinforcing-steel-selection5
  if  (reinforcing-steel-selection ==  "lbs/cuyd")
  then  (quantity-id = 32104  ,
        units = "TON"  ,
        amount =   number  *  ((length  * height
                                      *  (top-thickness + base-thickness)
                                      /  2.0
                                      +  length  * width  * footing-thickness)
                              * reinforcing-steel / 27.0)  / 1000.0  ,
        make-quantity(quantity-id,units,amount)
       )
)
;;;  Rules  for  footing keyway and construction  joint  selection
;;;
(take-off-rule  retaining-wall-footing-key-and-constr-joints-selectionl
  if  (footing-key-and-constr-joints-selection == "2x4  key  in  ftg")
  then  (quantity-id = 26633  ,
        units = "LNFT"  ,
        amount =   number  * length  ,
        make-quantity(quantity-id,units,amount)
       )
)
(take-off-rule  retaining-wall-footing-key-and-constr-joints-selection2
  if  (footing-key-and-constr-joints-selection == "2x6  key  in  ftg")
  then  (quantity-id = 26634  ,
        units = "LNFT"  ,
        amount =   number  * length  ,
        make-quantity(quantity-id,units,amount)
       )
)
```

```
(take-off-rule  retaining-wall-footing-key-and-constr-joints-selection3
  if  (footing-key-and-constr-joints-selection == "2x8 key in ftg")
  then  (quantity-id = 26635 ,
        units = "LNFT" ,
        amount =   number * length ,
        make-quantity(quantity-id,units,amount)
       )
)

(take-off-rule  retaining-wall-footing-key-and-constr-joints-selection4
  if  (footing-key-and-constr-joints-selection == "3x8 key in ftg")
  then  (quantity-id = 26636 ,
        units = "LNFT" ,
        amount =   number * length ,
        make-quantity(quantity-id,units,amount)
       )
)

(take-off-rule  retaining-wall-footing-key-and-constr-joints-selection5
  if  (footing-key-and-constr-joints-selection == "2x6 ftg key & one wall constr jt")
  then  (quantity-id = 26633 ,
        units = "LNFT" ,
        amount =   number * length ,
        make-quantity(quantity-id,units,amount) ,
        quantity-id = 26653 ,
        units = "SQFT" ,
        amount =   number * (height * (top-thickness + base-thickness) / 2.0) ,
        make-quantity(quantity-id,units,amount)
       )
)

(take-off-rule  retaining-wall-footing-key-and-constr-joints-selection6
  if  (footing-key-and-constr-joints-selection ==
                "2x6 ftg key & wall constr jt  30 in. oc")
  then  (quantity-id = 26633 ,
        units = "LNFT" ,
        amount =   number * length ,
        make-quantity(quantity-id,units,amount) ,
        quantity-id = 26653 ,
        units = "SQFT" ,
        amount =   number * ((height * (top-thickness + base-thickness) / 2.0)
                              * (length / 30.0)) ,
        make-quantity(quantity-id,units,amount)
       )
)

(take-off-rule  retaining-wall-footing-key-and-constr-joints-selection?
  if  (footing-key-and-constr-joints-selection == "one wall constr jt")
  then  (quantity-id = 26653 ,
        units = "SQFT" ,
        amount =   number * (height * (top-thickness + base-thickness) / 2.0) ,
        make-quantity(quantity-id,units,amount)
       )
)

(take-off-rule  retaining-wall-footing-key-and-constr-joints-selection8
  if  (footing-key-and-constr-joints-selection == "wall constr jt 30 in. oc")
  then  (quantity-id = 26653 ,
        units = "SQFT" ,
        amount =   number * ((height * (top-thickness + base-thickness) / 2.0)
                              * (length / 30.0)) ,
        make-quantity(quantity-id,units,amount)
       )
)

;;; Rules for miscellaneous
;;;
(take-off-rule  retaining-wall-miscellaneous-selectionl
  if  (miscellaneous-selection == "1 in. diam weep holes 36 in. o.c")
  then  (quantity-id = 26661 ,
        units = "PCS" ,
        amount =   number * (length / 3.0) ,
        make-quantity(quantity-id,units,amount)
       )
)
```

```
(take-off-rule  retaining-wall-miscellaneous-selection2
  if  (miscellaneous-selection == "1-1/2 in. diam weep holes 36 in.  o.c")
  then  (quantity-id = 26662 ,
        units = "PCS" ,
        amount =   number *  (length / 3.0)  ,
        make-quantity(quantity-id,units,amount)
       )
)
(take-off-rule  retaining-wall-miscellaneous-selection3
  if  (miscellaneous-selection == "2 in. diam weep holes 36 in.  o.c")
  then  (quantity-id = 26663 ,
        units = "PCS" ,
        amount =   number *  (length / 3.0)  ,
        make-quantity(quantity-id,units,amount)
       )
)
(take-off-rule  retaining-wall-miscellaneous-selection4
  if  (miscellaneous-selection == "3 in. diam weep holes 36 in.  o.c")
  then  (quantity-id = 26664 ,
        units = "PCS" ,
        amount =   number *  (length / 3.0)  ,
        make-quantity(quantity-id,units,amount)
       )
)
(take-off-rule  retaining-wall-miscellaneous-selection5
  if  (miscellaneous-selection == "rub exposed wall sfc & finish top")
  then  (quantity-id = 26620 ,
        units = "SQFT" ,
        amount =   number *  (length *  (height - width + 1.0))  ,
        make-quantity(quantity-id,units,amount)  ,
        quantity-id = 26621 ,    ; SUBTRACT POINT & PATCH
        units = "SQFT" ,
        amount = -   number *  (length *  (height - width + 1.0))  ,
        make-quantity(quantity-id,units,amount)  ,
        quantity-id = 26622 ,
        units = "SQFT" ,
        amount =   number * length ,
        make-quantity(quantity-id,units,amount)
       )
)
(take-off-rule  retaining-wall-miscellaneous-selection6
  if  (miscellaneous-selection == "ftg waterstop ")
  then  (quantity-id = 26665 ,
        units = "LNFT" ,
        amount =   number * length ,
        make-quantity(quantity-id,units,amount)
       )
)
(take-off-rule  retaining-wall-miscellaneous-selection?
  if  (miscellaneous-selection == "wall waterstop")
  then  (quantity-id = 26665 ,
        units = "LNFT" ,
        amount =   number * width ,
        make-quantity(quantity-id,units,amount)
       )
)
(take-off-rule  retaining-wall-miscellaneous-selection8
  if  (miscellaneous-selection == "wall waterstop  30 in o.c")
  then  (quantity-id = 26665 ,
        units = "LNFT" ,
        amount =   number *  (height * length / 30.0)  ,
        make-quantity(quantity-id,units,amount)
       )
)
```

7.3.3. Structural Metal Takeoff Rule Set

```
;;; RULES FOR STRUCTURAL METAL
;;;
;;; Rules for type of metal and type of member
;;;  (Note:  0.0034 is pounds of steel per square inch per linear foot.
;;;  Likewise,  0.0015 is pounds of aluminum per square inch per linear
;;;  foot.)
(take-off-rule  structural-metal-type-of-member-selectionl
  if (    (   type-of-metal-selection == "A-36 std stl."
           or type-of-metal-selection == "add hi-strngth steel"
           or type-of-metal-selection == "add specialty steel")
      and (type-of-member-selection == "beam"
           or type-of-member-selection == "column"
           or type-of-member-selection == "channel"))
  then (amount  =   number *  (pieces-per-location * length
                                                    * weight / 1000.0) ,
       )
)

(take-off-rule  structural-metal-type-of-member-selection2
  if (    (   type-of-metal-selection == "A-36 std stl."
           or type-of-metal-selection == "add hi-strngth steel"
           or type-of-metal-selection == "add specialty steel")
      and type-of-member-selection == "angle")
  then (amount =   number
                 *  (pieces-per-location * length
                                         *  (first-dimension + second-dimension)
                                         * thickness * 0.0034)
       )
)

(take-off-rule  structural-meta1-type-of-member-selection3
  if (    (   type-of-metal-selection == "A-36 std stl."
           or type-of-metal-selection == "add hi-strngth steel"
           or type-of-metal-selection == "add specialty steel")
      and type-of-member-selection == "plate")
  then (amount =   number
                 *  (pieces-per-location * length * first-dimension
                                         * thickness * 0.0034)
       )
)

(take-off-rule  structural-metal-type-of-member-select ion4
  if (    (   type-of-metal-selection == "A-36 std stl."
           or type-of-metal-selection == "add hi-strngth steel"
           or type-of-metal-selection == "add specialty steel")
      and type-of-member-selection == "tubular  (rect or sq.)[11])
  then (amount =   number * pieces-per-location
                           * (length * 2 * (first-dimension + second-dimension)
                                     * thickness * 0.0034)
       )
)

(take-off-rule  structural-meta1-type-of-member-selection5
  if (    (   type-of-metal-selection == "A-36 std stl."
           or type-of-metal-selection == "add hi-strngth steel"
           or type-of-metal-selection == "add specialty steel")
      and type-of-member-selection == "pipe column")
  then (amount =   number * pieces-per-location
                           * (length * (3.14 * first-dimension)
                                     * thickness * 0.0034)
       )
)
```

```
(take-off-rule structural-metal-type-of-member-selection6
  if (    (   type-of-metal-selection == "A-36 std stl."
           or type-of-metal-selection == "add hi-strngth steel"
           or type-of-metal-selection == "add specialty steel")
      and type-of-member-selection == "rectangular rod (solid)")
  then (amount =   number * pieces-per-location
                          * (length * first-dimension
                                    * second-dimension * 0.0034)
       )
)

(take-off-rule structural-metal-type-of-member-selection7
  if (    (   type-of-metal-selection == "A-36 std stl."
           or type-of-metal-selection == "add hi-strngth steel"
           or type-of-metal-selection == "add specialty steel")
      and type-of-member-selection == "round rod")
  then (amount =   number * pieces-per-location
                          * (length * ((first-dimension / 2.0) ** 2)
                                    * 3.14 * 0.0034)
       )
)

(take-off-rule structura1-metal-type-of-member-selection8
  if (    type-of-metal-selection == "structural aluminum"
      and (type-of-member-selection == "beam"
           or type-of-member-selection == "column"
           or type-of-member-selection == "channel"))
  then (amount  =   number * (pieces-per-location * length
                                                  * weight / 1000.0) ,
       )
)

(take-off-rule structural-metal-type-of-member-selection9
  if (    type-of-metal-selection == "structural aluminum"
      and type-of-member-selection == "angle")
  then (amount =   number
                 * (pieces-per-location * length
                                        * (first-dimension + second-dimension)
                                        * thickness * 0.0015)
       )
)

(take-off-rule structural-metal-type-of-member-selection10
  if (    type-of-metal-selection == "structural aluminum"
      and type-of-member-selection == "plate")
  then (amount =   number
                 * (pieces-per-location * length * first-dimension
                                        * thickness * 0.0015)
       )
)

(take-off-rule structural-metal-type-of-member-selectionll
  if (    type-of-metal-selection == "extruded aluminum"
      and type-of-member-selection == "tubular (rect or sq.)")
  then (amount =   number * pieces-per-location
                          * (length * 2 * (first-dimension + second-dimension)
                                    * thickness * 0.0015)
       )
)

(take-off-rule structural-metal-type-of-member-selectionl2
  if (    type-of-metal-selection == "extruded aluminum"
      and type-of-member-selection == "pipe column")
  then (amount =   number * pieces-per-location
                          * (length * (3.14 * first-dimension)
                                    * thickness * 0.0015)
       )
)
```

```
(take-off-rule  structural-metal-type-of-metal-selectionl
  if (    type-of-metal-selection == "A-36 std stl."
      and type-of-member-selection == "beam")
  then (quantity-id = 510140 ,
        units = "TON",
        make-quantity(quantity-id,units,amount)
       )
)

(take-off-rule  structural-metal-type-of-metal-selection2
  if (    type-of-metal-selection == "A-36 std stl."
      and type-of-member-selection == "column")
  then (quantity-id = 510010 ,
        units = "TON",
        make-quantity(quantity-id,units,amount)
       )
)

(take-off-rule  structural-metal-type-of-metal-selection3
  if (    type-of-metal-selection == "A-36 std stl."
      and type-of-member-selection == "channel")
  then (quantity-id = 510020 ,
        units = "TON",
        make-quantity(quantity-id,units,amount)
       )
)

(take-off-rule  structural-metal-type-of-meta1-selection4
  if (    type-of-metal-selection == "A-36 std stl."
      and type-of-member-selection == "angle")
  then (quantity-id = 510030 ,
        units = "TON",
        make-quantity(quantity-id,units,amount)
       )
)

(take-off-rule  structural-metal-type-of-metal-selection5
  if (    type-of-metal-selection == "A-36 std stl."
      and type-of-member-selection == "plate")
  then (quantity-id = 510040 ,
        units = "TON",
        make-quantity(quantity-id,units,amount)
       )
)

(take-off-rule  structural-metal-type-of-metal-selection6
  if (    type-of-metal-selection == "A-36 std stl."
      and type-of-member-selection == "tubular (rect or sq.)1')
  then (quantity-id = 510070 ,
        units = "TON",
        make-quantity(quantity-id,units,amount)
       )
)

(take-off-rule  structural-metal-type-of-metal-selection7
  if (    type-of-metal-selection == "A-36 std stl."
      and type-of-member-selection == "pipe column")
  then (quantity-id = 510080 ,
        units = "TON",
        make-quantity(quantity-id,units,amount)
       )
)

(take-off-rule  structural-metal-type-of-metal-selection8
  if (    type-of-metal-selection == "A-36 std stl."
      and type-of-member-selection == "rectangular rod (solid)")
  then (quantity-id = 510060 ,
        units = "TON",
        make-quantity(quantity-id,units,amount)
       )
)
```

```
(take-off-rule  structural-metal-type-of-metal-select ion9
  if (    type-of-metal-selection == "A-36 std stl."
      and type-of-member-selection == "round rod")
  then (quantity-id = 510050 ,
        units = "TON",
        make-quantity(quantity-id,units,amount)
       )
)
(take-off-rule  structural-metal-type-of-metal-seiectionlO
  if (    type-of-metal-selection == "add hi-strngth steel"
      and type-of-member-selection == "beam")
  then (quantity-id = 510140 ,
        units = "TON"/
        make-quantity(quantity-id,units,amount) ,
        quantity-id = 510500 ,
        units = "TON",
        make-quantity(quantity-id,units,amount)
       )
)
(take-off-rule  structural-metal-type-of-metal-select ion11
  if (    type-of-metal-selection == "add hi-strngth steel"
      and type-of-member-selection == "column")
  then (quantity-id = 510010 ,
        units = "TON",
        make-quantity(quantity-id,units,amount) ,
        quantity-id = 510500 ,
        units = "TON",
        make-quantity(quantity-id,units,amount)

       )
)
(take-off-rule  structural-metal-type-of-metal-select ion12
  if (    type-of-metal-selection == "add hi-strngth steel"
      and type-of-member-selection == "channel")
  then (quantity-id = 510020 ,
        units = "TON",
        make-quantity(quantity-id,units,amount) ,
        quantity-id = 510500 ,
        units = "TON",
        make-quantity(quantity-id,units,amount)

       )
)
(take-off-rule  structural-metal-type-of-metal-selection13
  if (    type-of-metal-selection == "add hi-strngth steel"
      and type-of-member-selection == "angle")
  then (quantity-id = 510030 ,
        units = "TON",
        make-quantity(quantity-id,units,amount) ,
        quantity-id = 510500 ,
        units = "TON",
        make-quantity(quantity-id,units,amount)

       )
)
(take-off-rule  structural-metal-type-of-metal-selectionl4
  if (    type-of-metal-selection == "add hi-strngth steel"
      and type-of-member-selection == "plate")
  then (quantity-id = 510040 ,
        units = "TON",
        make-quantity(quantity-id,units,amount) ,
        quantity-id = 510500 ,
        units = "TON",
        make-quantity(quantity-id,units,amount)

       )
)
```

```
(take-off-rule  structural-metal-type-of-metal-selectionl5
  if (    type-of-metal-selection == "add hi-strngth steel"
      and type-of-member-selection == "tubular (rect or sq.)")
  then (quantity-id = 510070 ,
        units = "TON",
        make-quantity(quantity-id,units,amount)
       )
)
(take-off-rule  structural-metal-type-of-metal-selectionl6
  if (    type-of-metal-selection == "add hi-strngth steel"
      and type-of-member-selection == "pipe column")
  then (quantity-id = 510080 ,
        units = "TON",
        make-quantity(quantity-id,units,amount) ,
        quantity-id = 510500 ,
        units = "TON",
        make-quantity(quantity-id,units,amount)
       )
)
(take-off-rule  structural-metal-type-of-metal-selectionl7
  if (    type-of-metal-selection == "add hi-strngth steel"
      and type-of-member-selection == "rectangular rod (solid)")
  then (quantity-id = 510060 ,
        units = "TON",
        make-quantity(quantity-id,units,amount) ,
        quantity-id = 510500 ,
        units = "TON",
        make-quantity(quantity-id,units,amount)
       )
)
(take-off-rule  structural-metal-type-of-metal-selectionl8
  if (    type-of-metal-selection == "add hi-strngth steel"
      and type-of-member-selection == "round rod")
  then (quantity-id = 510050 ,
        units = "TON",
        make-quantity(quantity-id,units,amount) ,
        quantity-id = 510500 ,
        units = "TON",
        make-quantity(quantity-id,units,amount)
       )
)
(take-off-rule  structural-metal-type-of-metal-selection20
  if (    type-of-metal-selection == "add specialty steel"
      and type-of-member-selection == "beam")
  then (quantity-id = 510140 ,
        units = "TON",
        make-quantity(quantity-id,units,amount) ,
        quantity-id = 510510 ,
        units = "TON",
        make-quantity(quantity-id,units,amount)
       )
)
(take-off-rule  structural-metal-type-of-metal-selection21
  if (    type-of-metal-selection == "add specialty steel"
      and type-of-member-selection == "column")
  then (quantity-id = 510010 ,
        units = "TON",
        make-quantity(quantity-id,units,amount) ,
        quantity-id = 510510 ,
        units = "TON",
        make-quantity(quantity-id,units,amount)
       )
)
```

```
(take-off-rule  structural-metal-type-of-metal-selection22
  if (    type-of-metal-selection == "add specialty steel"
      and type-of-member-selection == "channel")
  then (quantity-id = 510020 ,
        units = "TON",
        make-quantity(quantity-id,units,amount) ,
        quantity-id = 510510 ,
        units = "TON",
        make-quantity(quantity-id,units,amount)

       )
)
(take-off-rule  structural-metal-type-of-metal-selection23
  if (    type-of-metal-selection == "add specialty steel"
      and type-of-member-selection == "angle")
  then (quantity-id = 510030 ,
        units = "TON",
        make-quantity(quantity-id,units,amount) ,
        quantity-id = 510510 ,
        units = "TON",
        make-quantity(quantity-id,units,amount)

       )
)
(take-off-rule  structural-metal-type-of-metal-selection24
  if (    type-of-metal-selection == "add specialty steel"
      and type-of-member-selection == "plate")
  then (quantity-id = 510040 ,
        units = "TON",
        make-quantity(quantity-id,units,amount) ,
        quantity-id = 510510 ,
        units = "TON",
        make-quantity(quantity-id,units,amount)

       )
)
(take-off-rule  structural-meta1-type-of-metal-selection25
  if (    type-of-metal-selection == "add specialty steel"
      and type-of-member-selection == "tubular (rect or sq.)")
  then (quantity-id = 510070 ,
        units = "TON",
        make-quantity(quantity-id,units,amount)
       )
)
(take-off-rule structural-metal-type-of-metal-selection2 6
  if (    type-of-metal-selection == "add specialty steel"
      and type-of-member-selection == "pipe column")
  then (quantity-id = 510080 ,
        units = "TON",
        make-quantity(quantity-id,units,amount) ,
        quantity-id = 510510 ,
        units = "TON",
        make-quantity(quantity-id,units,amount)
       )
)
(take-off-rule  structural-metal-type-of-metal-selection27
  if (    type-of-metal-selection == "add specialty steel"
      and type-of-member-selection == "rectangular rod (solid)")
  then (quantity-id = 510060 ,
        units = "TON",
        make-quantity(quantity-id,units,amount) ,
        quantity-id = 510510 ,
        units = "TON",
        make-quantity(quantity-id,units,amount)
       )
)
```

```
(take-off-rule  structural-metal-type-of-metal-selection28
  if (    type-of-metal-selection == "add specialty steel"
      and type-of-member-selection == "round rod")
  then (quantity-id = 510050 ,
        units = "TON",
        make-quantity(quantity-id,units,amount) ,
        quantity-id = 510510 ,
        units = "TON",
        make-quantity(quantity-id,units,amount)
       )
)

(take-off-rule  structural-metal-type-of-metal-selection30
  if (type-of-metal-selection == "structural aluminum")
  then (quantity-id = 510600 ,
        units = "TON",
        make-quantity(quantity-id,units,amount) ,
       )
)

(take-off-rule  structural-metal-type-of-metal-selection31
  if (type-of-metal-selection == "extruded aluminum")
  then (quantity-id = 510600 ,
        units = "TON",
        make-quantity(quantity-id,units,amount) ,
       )
)

;;; Rules for surface treatment
;;;  (Note:  amount is set by type-of-member-selection rules.)
;;;
(take-off-rule  structural-meta1-surface-treatment-selection1
  if (surface-treatment-selection == "field paint per ton")
  then (quantity-id = 991040 ,
        units = "TON" ,
        make-quantity(quantity-id,units,amount)
       )
)

(take-off-rule  structura1-meta1-surface-treatment-selection2
  if (surface-treatment-selection == "galvanize per ton")
  then (quantity-id = 991520 ,
        units = "TON" ,
        make-quantity(quantity-id,units,amount)
       )
)

;;; Rules for anchor bolts.
;;;
(take-off-rule  structura1-metal-anchor-bolts-selectionl
  if (anchor-bolts-selection  == "furnish 2 a.b. per col.")
  then (quantity-id = 510100,
        units = "LBS" ,
        amount =   number * (2 * pieces-per-location
                               * ((anchor-bolt-diameter / 2.0)  ** 2)
                               * 3.14 * 3.4) ,
        make-quantity(quantity-id,units,amount)
       )
)

(take-off-rule  structural-meta1-anchor-boIts-selection2
  if (anchor-bolts-selection  == "furnish 4 a.b. per col.")
  then (quantity-id = 510100,
        units = "LBS" ,
        amount =   number * (2 * pieces-per-location
                               * ((anchor-bolt-diameter / 2.0)  ** 2)
                               * 3.14 * 3.4) ,
        make-quantity(quantity-id,units,amount)
       )
)
```

Chapter 8
JADE Supplement

This chapter is intended to supplement the description of the JADE-KADBASE implementation found in Section 5.3 of the thesis. The supplement contains the schema description information for the JADE KBS context, the structural configuration database (SCD), the analysis results database (ARD) (including the LFBS, LFBM, and LIM for each component) as well as the GS, GDSM, and GIM.

8.1. Schemata

The following three sections describe the schemata for JADE, the structural configuration database, the analysis results database, and the global schema. The schema descriptions are expressed in the problem-oriented language of the schema description utility (described in Section 4.7). Each of the component description sections contains the local frame-based schema (LFBS), the local frame-based mapping (LFBM), and the local integration mapping (LIM) grouped into a single set of definitions. The final subsection presents the global schema (GS), the global data source mapping (GDSM), and the global integration mapping (GIM) in the same fashion.

8.1.1. JADE Context Schemata

```
lfbs jade

entity joint
lfbm joint
   slot joint-id
      slot-type attribute
      data-type integer
      lfbm (joint . joint-id)
   slot x
      slot-type attribute
      data-type number
      lfbm (joint . x)
      units feet
   slot y
      slot-type attribute
      data-type number
      lfbm (joint . y)
      units feet
   slot z
      slot-type attribute
      data-type number
      lfbm (joint . z)
      units feet
   slot joined-members
      slot-type relationship
      data-type symbol
      lfbm (joint . joined-members)
      relationship-type sub-part
   slot number-of-members
      slot-type attribute
      data-type integer
      lfbm (joint . number-of-members)
candidate-key joint-id
primary-key joint-id

entity member
lfbm member
   slot member-id
      slot-type attribute
      data-type integer
      lfbm (member . member-id)
   slot jointl
      slot-type relationship
      data-type symbol
      lfbm (member . jointl)
      relationship-type is-a
   slot xl
      slot-type attribute
      data-type number
      lfbm (member . xl)
      units feet
   slot yl
      slot-type attribute
      data-type number
      lfbm (member . yl)
      units feet
   slot zl
      slot-type attribute
      data-type number
      lfbm (member . zl)
      units feet
   slot shearl
      slot-type attribute
      data-type number
      lfbm (member . shearl)
      lim FY1
      units kips
   slot axiall
      slot-type attribute
      data-type number
      lfbm (member . axiall)
      lim FX1
      units kips
   slot moment1
      slot-type attribute
      data-type number
      lfbm (member . momentl)
      lim MZ1
      units kip-inches
   slot fixityl
      slot-type attribute
      data-type string
      lfbm (member . fixityl)
   slot connectionl
      slot-type relationship
      data-type symbol
      lfbm (member . connectionl)
      relationship-type is-a
```

```
slot joint2
   slot-type relationship
   data-type symbol
   lfbm (member . joint2)
   relationship-type is-a
slot x2
   slot-type attribute
   data-type number
   lfbm (member . x2)
   units feet
slot y2
   slot-type attribute
   data-type number
   lfbm (member . y2)
   units feet
slot z2
   slot-type attribute
   data-type number
   lfbm (member . z2)
   units feet
slot shear2
   slot-type attribute
   data-type number
   lfbm (member . shear2)
   lim FY2
   units kips
slot axial2
   slot-type attribute
   data-type number
   lfbm (member . axial2)
   lim FX2
   units kips
slot moment2
   slot-type attribute
   data-type number
   lfbm (member . moment2)
   lim MZ2
   units kip-inches
slot fixity2
   slot-type attribute
   data-type string
   lfbm (member . fixity2)
slot connection2
   slot-type relationship
   data-type symbol
   lfbm (member . connection2)
   relationship-type is-a
slot length
   slot-type attribute
   data-type number
   lfbm (member . length)
   units feet
slot delta-x
   slot-type attribute
   data-type number
   lfbm (member . delta-x)
slot delta-y
   slot-type attribute
   data-type number
   lfbm (member . delta-y)
slot delta-z
   slot-type attribute
   data-type number
   lfbm (member . delta-z)
slot roll-angle
   slot-type attribute
```

```
      data-type number
      lfbm (member . roll-angle)
      units degrees
   slot grade
      slot-type attribute
      data-type string
      lfbm (member . grade)
   slot section-type
      slot-type attribute
      data-type string
      lfbm (member . section-type)
   slot designation
      slot-type attribute
      data-type string
      lfbm (member . designation)
candidate-key member-id
primary-key member-id

entity member
   slot jointl
      range joint
   slot joint2
      range joint

entity joint
   slot joined-members
      range member
;
```

8.1.2. Structural Configuration Database

```
lfbs scd

entity joint
lfbm joint
   slot jid
      slot-type attribute
      data-type integer
      lfbm (joint . jid)
      lim joint-id
   slot x
      slot-type attribute
      data-type number
      lfbm (joint . x)
      units feet
   slot y
      slot-type attribute
      data-type number
      lfbm (joint . y)
      units feet
   slot z
      slot-type attribute
      data-type number
      lfbm (joint . z)
      units feet
candidate-key jid
primary-key jid

entity member
lfbm member
   slot mid
      slot-type attribute
      data-type integer
      lfbm (member . mid)
      lim member-id
   slot jointl
      slot-type relationship
      data-type symbol
      lfbm (member . jointl)
      relationship-type is-a
   slot fixityl
      slot-type attribute
      data-type string
      lfbm (member . fixityl)
   slot joint2
      slot-type relationship
      data-type symbol
      lfbm (member . joint2)
      relationship-type is-a
   slot fixity2
      slot-type attribute
      data-type string
      lfbm (member . fixity2)
   slot roll-angle
      slot-type attribute
      data-type number
      lfbm (member . roll-angle)
      units degrees
   slot grade
      slot-type attribute
      data-type string
      lfbm (member . grade)
   slot section-type
      slot-type attribute
      data-type string
      lfbm (member . section-type)
   slot designation
      slot-type attribute
      data-type string
      lfbm (member . designation)
candidate-key mid
primary-key mid

entity member
   slot jointl
      range joint
   slot joint2
      range joint

entity joint
   slot joined-members
      range member
;
```

8.1.3. Analysis Results Database Schemata

```
lfbs adb

entity endloads
lfbm endloads
lim member
   slot mid
      slot-type attribute
      data-type integer
      lfbm (member . mid)
      lim member-id
   slot FX1
      slot-type attribute
      data-type real
      lfbm (endloads . FX1)
      units kips
   slot FY1
      slot-type attribute
      data-type real
      lfbm (endloads . FY1)
      units kips
   slot FZ1
      slot-type attribute
      data-type real
      lfbm (endloads . FZ1)
      units kips
   slot MX1
      slot-type attribute
      data-type real
      lfbm (endloads . MX1)
      units kip-inches
   slot MY1
      slot-type attribute
      data-type real
      lfbm (endloads . MY1)
      units kip-inches
   slot MZ1
      slot-type attribute
      data-type real
      lfbm (endloads . MZ1)
      units kip-inches
   slot FX2
      slot-type attribute
      data-type real
      lfbm (endloads . FX2)
      units kips
   slot FY2
      slot-type attribute
      data-type real
      lfbm (endloads . FY2)
      units kips
   slot FZ2
      slot-type attribute
      data-type real
      lfbm (endloads . FZ2)
      units kips
   slot MX2
      slot-type attribute
      data-type real
      lfbm (endloads . MX2)
      units kip-inches
   slot MY2
      slot-type attribute
      data-type real
      lfbm (endloads . MY2)
      units kip-inches
   slot MZ2
      slot-type attribute
      data-type real
      lfbm (endloads . MZ2)
      units kip-inches
;
```

8.1.4. Global Schemata

```
global-schema

entity joint
sources (jad scd)
   slot joint-id
      slot-type attribute
      data-type integer
      sources (jad scd)
   slot x
      slot-type attribute
      data-type number
      units feet
      sources (jad scd)
   slot y
      slot-type attribute
      data-type number
      units feet
      sources (jad scd)
   slot z
      slot-type attribute
      data-type number
      units feet
      sources (jad scd)
   slot xl
      slot-type attribute
      data-type number
      units feet
   slot yl
      slot-type attribute
      data-type number
      units feet
   slot zl
      slot-type attribute
      data-type number
      units feet
   slot x2
      slot-type attribute
      data-type number
      units feet
   slot y2
      slot-type attribute
      data-type number
      units feet
   slot z2
      slot-type attribute
      data-type number
      units feet
   slot joined-members
      slot-type relationship
      data-type symbol
      relationship-type sub-part
      sources (jad)
   slot number-of-members
      slot-type attribute
      data-type integer
      sources (jad)
candidate-key joint-id
primary-key joint-id

entity member
sources (jad scd ard)
   slot member-id
   slot-type attribute
   data-type integer
   sources (jad scd ard)
slot jointl
   slot-type relationship
   data-type symbol
   relationship-type is-a
   sources (jad scd)
slot xl
   slot-type attribute
   data-type number
   units feet
   inherited-through jointl
   sources (jad)
slot yl
   slot-type attribute
   data-type number
   units feet
   inherited-through jointl
   sources (jad)
slot zl
   slot-type attribute
   data-type number
   units feet
   inherited-through jointl
   sources (jad)
slot fixityl
   slot-type attribute
   data-type string
   sources (jad scd)
slot connectionl
   slot-type relationship
   data-type symbol
   relationship-type is-a
   sources (jad)
slot joint2
   slot-type relationship
   data-type symbol
   relationship-type is-a
   sources (jad scd)
slot x2
   slot-type attribute
   data-type number
   units feet
   inherited-through joint2
   sources (jad)
slot y2
   slot-type attribute
   data-type number
   units feet
   inherited-through joint2
   sources (jad)
slot z2
   slot-type attribute
   data-type number
   units feet
   inherited-through joint2
   sources (jad)
slot fixity2
   slot-type attribute
   data-type string
   sources (jad scd)
slot connection2
   slot-type relationship
   data-type symbol
   relationship-type is-a
   sources (jad)
slot length
   slot-type attribute
   data-type number
   units feet
   sources (jad)
slot delta-x
   slot-type attribute
   data-type number
   sources (jad)
slot delta-y
   slot-type attribute
   data-type number
   sources (jad)
slot delta-z
   slot-type attribute
   data-type number
   sources (jad)
slot roll-angle
   slot-type attribute
   data-type number
   units degrees
   sources (jad scd)
slot grade
   slot-type attribute
   data-type string
   sources (jad scd)
slot section-type
   slot-type attribute
   data-type string
   sources (jad scd)
slot designation
   slot-type attribute
   data-type string
   sources (jad scd)
slot FX1
   slot-type attribute
   data-type real
   units kips
   sources (jad ard)
slot FY1
   slot-type attribute
   data-type real
   units kips
   sources (jad ard)
slot FZ1
   slot-type attribute
   data-type real
   units kips
   sources (jad ard)
slot MX1
   slot-type attribute
   data-type real
   units kip-inches
   sources (jad ard)
slot MY1
   slot-type attribute
   data-type real
   units kip-inches
   sources (jad ard)
slot MZ1
   slot-type attribute
   data-type real
   units kip-inches
   sources (jad ard)
slot FX2
```

```
      slot-type attribute
      data-type real
      units kips
      sources (jad ard)
   slot FY2
      slot-type attribute
      data-type real
      units kips
      sources (jad ard)
   slot FZ2
      slot-type attribute
      data-type real
      units kips
      sources (jad ard)
   slot MX2
      slot-type attribute
      data-type real
      units kip-inches
      sources (jad ard)
   slot MY2
      slot-type attribute
      data-type real
      units kip-inches
      sources (jad ard)
   slot MZ2
    .  slot-type attribute
      data-type real
      units kip-inches
      sources (jad ard)
candidate-key member-id
primary-key member-id

entity member
   slot jointl
      range joint
   slot joint2
      range joint

entity joint
   slot joined-members
      range member

entity joint
  slot xl
    constraint (joint.xl == joint.x)
  slot yl
    constraint (joint.yl == joint.y)
  slot zl
    constraint (joint.zl == joint.z)
  slot x2
    constraint (joint.x2 == joint.x)
  slot y2
    constraint (joint.y2 == joint.y)
  slot z2
    constraint (joint.z2 == joint.z)

entity member
  slot length
    constraint (member.. length ==
     sqrti   (member.x2 - member.xl) ** 2
          +  (member.y2 - member.yl) ** 2
          +  (member.z2 - member.zl) ** 2) )
  slot delta-x
    constraint (member.delta-x == (member.x2 - member.xl) / member.length)
  slot delta-y
    constraint (member.delta-y == (member.y2 - member.yl) / member.length)
  slot delta-z
    constraint  (member.delta-z == (member.z2 - member.zl)  / member.length)

;
```

References

[AISC 80] *Manual of Steel Construction*, Eighth Edition, American Institute of Steel Construction, Chicago, IL, 1980.

[Date 81] Date, C. J., *An Introduction to Database Systems,* Vol. I, Third Edition, *The System Programming Series*, Addison-Wesley Publishing Co., Reading, Massachusetts, 1981.

[Howard 86] Howard, H. C, and Rehak, D. R., *Interfacing Databases and Knowledge Based Systems for Structural Engineering Applications,* Technical Report EDRC-12-06-86, Engineering Design Research Center, Carnegie-Mellon University, Pittsburgh, PA, November 1986.

[Maher 84] Maher, M. L., *HI-RISE: An Expert System For The Preliminary Structural Design OfHigh Rise Buildings*, unpublished Ph.D. Dissertation, Department of Civil Engineering, Carnegie-Mellon University, 1984.

[Maher 85] Maher, M. L., and Fenves, S. J., *'HI-RISE: An Expert System for the Preliminary Structural Design of High Rise Buildings," in *Knowledge Engineering in Computer-Aided Design,* Gero, J.S., Ed., North-Holland, Amsterdam, 1985.

[Manner 86] Manner, R., "Strong Typing and Physical Units," *SIGPLAN Notices,* Vol. 21, No. 3, pp. 11-20, March 1986.

[Means 86] W.D. Mahoney Ed., *Building Construction Cost Data 1986*, 44th Edition, R.S. Means Company, Inc., Kingston, MA, 1986.

[Stonebraker 76] Stonebraker, M., Wong, E., and Kreps, P., "The Design and Implementation of INGRES," *ACM Transactions on Database Systems,* Vol.1, No. 3, pp. 189-222, September 1976.

[Woodfill81] Woodfill, J., Siegal, P., Ranstrom, J., Meyer, M., and Allman, E., *INGRES Reference Manual,* Version 7, 1981.