1988

Technical description of KADBASE

H. C. Howard
*Carnegie Mellon University*

Daniel R. Rehak


Follow this and additional works at: [http://repository.cmu.edu/cee](http://repository.cmu.edu/cee)
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 KADBSE

by

H.C. Howard and D.R. Rehak

EDRC-12-20-88
TECHNICAL DESCRIPTION
OF KADBASSE

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
# Technical Description of KADBASE

## Table of Contents

1. Introduction to the Technical Report .......................... 1
2. KADBASE Query Language ........................................... 2
   2.1. KQL Commands .................................................. 3
      2.1.1. Range .................................................... 3
      2.1.2. Retrieve ............................................... 3
      2.1.3. Update ................................................ 5
   2.2. General Definitions .......................................... 6
      2.2.1. Qualifiers .............................................. 6
      2.2.2. Clauses ............................................... 6
      2.2.3. Expressions .......................................... 7
   2.3. Optimizing Criteria in KQL ................................ 9
      2.3.1. Single-Object Queries: Examples and Discussion .... 9
      2.3.2. KQL Form for Optimizing Criteria .................... 10
      2.3.3. Implications For Constraint Processing .............. 12
3. Message Data Structure ........................................... 13
   3.1. Special Purpose Message Frames ......................... 14
      3.1.1. Message Frame .......................................... 14
      3.1.2. Retrieve Frame ......................................... 14
      3.1.3. Update Frame .......................................... 15
      3.1.4. Reply Frame ........................................... 17
      3.1.5. Entity Variable Frame ................................ 17
   3.2. General Purpose Message Tree Node Frames ............... 18
   3.3. Sample Message Frames ..................................... 21
      3.3.1. Sample SPEX Query ..................................... 21
      3.3.2. Sample SPEX Reply .................................... 30
4. KADBASE Schema Description Details .......................... 31
   4.1. Local Frame-Based Schema Definition ..................... 31
      4.1.1. LFBS Entity Definition ................................ 32
      4.1.2. LFBS Slot Definition .................................. 32
   4.2. Local Frame-Based Mapping ................................ 34
   4.3. Local Integration Mapping ................................. 35
   4.4. Global Schema ................................................ 37
      4.4.1. GS Entity Definition .................................. 38
      4.4.2. GS Slot Definition .................................... 38
   4.5. Global Data Source Mapping ................................ 40
   4.6. Global Integration Mapping ................................ 41
   4.7. Schema Definition Utility .................................. 42
5. KADBASE Communications ....................................... 47
   5.1. Message Files ................................................ 47
   5.2. Sending Messages ............................................ 48
   5.3. Receiving Messages .......................................... 48
<table>
<thead>
<tr>
<th>Section</th>
<th>Pages</th>
</tr>
</thead>
<tbody>
<tr>
<td>6. SPEX Supplement</td>
<td>49</td>
</tr>
<tr>
<td>6.1. Schemata</td>
<td></td>
</tr>
<tr>
<td>6.1.1. SPEX Context Schemata</td>
<td>49</td>
</tr>
<tr>
<td>6.1.2. AISC Database Schemata</td>
<td>51</td>
</tr>
<tr>
<td>6.1.3. Global Schemata</td>
<td>53</td>
</tr>
<tr>
<td>6.2. Program Traces</td>
<td>55</td>
</tr>
<tr>
<td>6.2.1. SPEX Trace</td>
<td>55</td>
</tr>
<tr>
<td>6.2.2. AISC KBDBI Trace</td>
<td>67</td>
</tr>
<tr>
<td>7. HICOST-n Supplement</td>
<td>69</td>
</tr>
<tr>
<td>7.1. Schemata</td>
<td></td>
</tr>
<tr>
<td>7.1.1. HICOST-II Context Schemata</td>
<td>69</td>
</tr>
<tr>
<td>7.1.2. Building Design Database Schemata</td>
<td>75</td>
</tr>
<tr>
<td>7.1.3. Project Management Database Schemata</td>
<td>85</td>
</tr>
<tr>
<td>7.1.4. Unit Cost Database Schemata</td>
<td>86</td>
</tr>
<tr>
<td>7.1.5. Global Schemata</td>
<td>86</td>
</tr>
<tr>
<td>7.2. Data</td>
<td>93</td>
</tr>
<tr>
<td>7.2.1. Building Design Database Data</td>
<td>93</td>
</tr>
<tr>
<td>7.2.2. Unit Cost Database Data</td>
<td>94</td>
</tr>
<tr>
<td>7.2.3. Project Management Database Data</td>
<td>96</td>
</tr>
<tr>
<td>7.3. Component Takeoff Rules</td>
<td></td>
</tr>
<tr>
<td>7.3.1. Concrete Footing Takeoff Rule Set</td>
<td>99</td>
</tr>
<tr>
<td>7.3.2. Retaining Wall Takeoff Rule Set</td>
<td>119</td>
</tr>
<tr>
<td>7.3.3. Structural Metal Takeoff Rule Set</td>
<td>134</td>
</tr>
<tr>
<td>8. JADE Supplement</td>
<td>141</td>
</tr>
<tr>
<td>8.1. Schemata</td>
<td></td>
</tr>
<tr>
<td>8.1.1. JADE Context Schemata</td>
<td>141</td>
</tr>
<tr>
<td>8.1.2. Structural Configuration Database</td>
<td>144</td>
</tr>
<tr>
<td>8.1.3. Analysis Results Database Schemata</td>
<td>145</td>
</tr>
<tr>
<td>8.1.4. Global Schemata</td>
<td>147</td>
</tr>
<tr>
<td>REFERENCES</td>
<td>152</td>
</tr>
</tbody>
</table>
Technical Description of KADBASE

List of Figures

Figure 3-1: Graphical Representation of SPEX Query  

22
Technical Description of KADBAS

List of Tables

Table 3-1: KADBAS Operator Template Frame Definitions 19
Table 3-2: KADBAS Literal Value Template Frame Definitions 20
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
KADB BASE Query Language

KQL (KADB BASE 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 KADB BASE 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 KADB BASE 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 \(\mid\) separate alternate definitions.

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

---

\(^1\)In retrospect, some argument can be made for the use of relational algebra as the internal format for KADB BASE 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:

\[
<\text{range}> ::= \text{range of } <\text{entity-variable}> \text{ is } <\text{entity-name}>
\]

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

\[
\text{range of beaml is beams}
\]

"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-criteria-list> ::= <optimizing-criterion> | <optimizing-criteria-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 wfl.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 = "W10x12")
    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 = "M10x12")
```

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 `joint1` 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 j1 is joints
range of j2 is joints
range of beam is beams
range of aj is adjacent-joints
update combined (aj.joint1 = j1.number, aj.joint2 = j2.number,
    aj.distance = beam.length)
    where beam.left-joint = j1 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> ::= < | <= | > | >= | == | !=
2.2.3. Expressions

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

\[ <\text{expression}> := <\text{arithmetic-expression}> | <\text{string-expression}> | <\text{boolean-expression}> \]

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:

\[ <\text{arithmetic-expression}> ::= <\text{arithmetic-constant}> | <\text{arithmetic-slot-reference}> | <\text{arithmetic-function}> | <\text{arithmetic-aggregate}> | ( <\text{arithmetic-expression}> ) | <\text{unary-arithmetic-operator}> ( <\text{arithmetic-expression}> ) | <\text{arithmetic-expression}> <\text{binary-arithmetic-operator}> <\text{arithmetic-expression}> \]

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 \(^\text{**}\) and \(^\text{^}\) are both used to indicate exponentiation. The operators are shown in order of descending precedence:

\[ \begin{align*}
<\text{unary-arithmetic-operators}> & := + | - \\
<\text{binary-arithmetic-operators}> & := \text{**} | \text{^} | * | / | + | - \\
\end{align*} \]

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.
Technical Description of KADBASE

\[
\text{<arithmetic-function> ::= abs(<arithmetic-expression>) | atan(<arithmetic-expression>) | cos(<arithmetic-expression>) | exp(<arithmetic-expression>) | log(<arithmetic-expression>) | mod(<arithmetic-expression>, <arithmetic-expression>) | sin(<arithmetic-expression>) | sqrt(<arithmetic-expression>) | float(<arithmetic-expression>) | fix(<arithmetic-expression>)}
\]

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

\[
\text{<arithmetic-aggregate> ::= <arithmetic-aggregate-operator>(<arithmetic-expression> [where <qualifier>])}
\]

\[
\text{<arithmetic-aggregate-operator> ::= count | sum | avg | max | min}
\]

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:

\[
\text{<string-expression> ::= "<string-constant>" | <string-slot-reference> | <string-function> | <string-constant> ::= <character> | <string-constant><string-constant>}
\]

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.
Technical Description of KADBASE

\[
\text{string-function} ::= \text{concat}(\text{string-expression} , \text{string-expression}) | \\
\text{substring}(\text{string-expression}, \text{integer-arithmetic-expression}, \text{integer-arithmetic-expression}) | \\
\text{ascii}(\text{arithmetic-expression})
\]

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

\[
\text{boolean-expression} ::= \text{boolean-slot-reference} | \\
\text{boolean-aggregate} | \\
\text{true} | \\
\text{false}
\]

\[
\text{boolean-aggregate} ::= \text{any}(\text{slot-expression} [\text{where <qualifier>}])
\]

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 \text{attributes of object} where \text{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:

\text{QUEL}

\begin{align*}
\text{range of wf is wide-flange-sections} \\
\text{retrieve (wf.number)} \\
\text{where wf.section-modulus} > 300 \\
\text{and wf.weight} = \text{MIN}(\text{wf.weight} \\
\text{where wf.section-modulus} > 300)
\end{align*}

SQL

```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: slot1-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.
Technical Description of KADBASE

```plaintext
retriever
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

---

³This 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.

⁴Since 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.

5Sec 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 KADBAS-specific description of the error resulting from the request. For the prototype implementation, KADBAS 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.
Technical Description of KADBASE

• **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 *lfs-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., "+", "-", "/", 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:
Table 3-1: KADBASE Operator Template Frame Definitions

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

node1
  IS-A: kad-op==
  MESSAGE-SUBPART: node2 node3

node2
  IS-A: kad-integer-value
  VALUE: 10
  PART-OF-MESSAGE: node1

node3
  ISA: kad-slot-value
  ENTITY-REFERENCE: wf-entity-variable
  SLOT-REFERENCE: area
  PART-OF-MESSAGE: node1

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.
Technical Description of KADBASE

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.
**Technical Description of KADBASE**

Retrieval

- **Target List**
  - **Qualifier**
    - Entity Variables
      - Optimizing Criteria
        - $\text{plastic-modulus} \times \text{depth} \leq 2400 \times 120.0 - \sqrt{\frac{36.0}{300}} \times \text{area}$
        - and
          - $\frac{0.9}{\sqrt{\frac{36.0}{300}} \times \text{plastic-modulus}} \times \text{area} + \text{flange-thickness} \times \text{web-thickness} > \frac{\text{flange-width}}{\text{thickness}}$
          - and
            - $\frac{2}{\text{flange-width}} + 3 \times \frac{12}{\text{web-thickness}} + 3 \times \text{depth} - \frac{\text{flange-width}}{\text{thickness}}$

Figure 3-1: Graphical Representation of SPEX Query
Technical Description of KADBASE

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-g00019 spx-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-g00020 spx-msg-tree-g00019
    spx-msg-tree-g00018 spx-msg-tree-g00017
    spx-msg-tree-g00016 spx-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
  ISA: 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-g00023 spx-msg-tree-g00021
  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
  ISA: 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
  ISA: kad-op*
  INSTANCE: message-tree-node
spx-msg-tree-g00026
  TYPE: number
  PART-OF-MESSAGE: spx-msg-tree-g00025
  VALUE: 0.9
  ISA: 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
  ISA: 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
  ISA: 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
  ISA: 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
  ISA: 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
  ISA: kad-slot-value
  INSTANCE: message-tree-node

spx-msg-tree-g00043
  TYPE: number
  PART-OF-MESSAGE: spx-msg-tree-g00041
  VALUE: 12
  ISA: kad-integer-value
  INSTANCE: message-tree-node
Technical Description of KADBASE

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
  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
  ISA: 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
  ISA: 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
  ISA: 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
Technical Description of KADBASE

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

spx-msg-tree-g00054
TYPE: number
PART-OF-MESSAGE: spx-msg-tree-g00048
VALUE: 12
ISA: 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
ISA: kad-op*
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
ISA: kad-slot-value
INSTANCE: message-tree-node

spx-msg-tree-g00057
TYPE: number
PART-OF-MESSAGE: spx-msg-tree-g00055
VALUE: 3
ISA: 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
ISA: 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-g00065 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

27
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
  ISA: 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
  ISA: 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-g00016
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-retrieve-g00013
ENTITY-REFERENCE: spx-entity-g00014
SLOT-REFERENCE: area
IS-A: kad-slot-value
INSTANCE: message-tree-node
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.

```
ais519007098
  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
KADBASSE 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 KADBASSE 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 definition 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 in 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.
Technical Description of KADBASE

- **LOCAL-CONSTRAINTS** — a list of local constraints. 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.

- **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/lf, dollars, etc. Only those units that have been defined in the global set of unit descriptions are allowed.

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.

10 The 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 $M_{kad-units}$.
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.\textsuperscript{11}

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

- $\text{LFBM}$ — 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:

- $\text{LFBM}$ — 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.

\textsuperscript{11}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 the 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.
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.
- **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

---

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/lf, dollars, etc. Only those units that have been defined in the global set of unit descriptions are allowed.\(^\text{14}\)

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 ..., ad infinitum.

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

\[
\text{UNIT-COST} = \text{MATLCOST} + \text{INSTCOST}
\]

That constraint is represented by the constraint frame and the five message tree node frames shown below.
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.
Technical Description of KADBASE

\[<\text{LFBS description} := \text{lfbs} \ <\text{lfbs name}> \]
\[<\text{lfbs entities}> := <\text{lfbs entity description}> | \]
\[<\text{lfbs entities}> <\text{lfbs entity description}> \]

\[<\text{lfbs entity description} := \text{entity} \ <\text{entity name}> \]
\[\text{lfbm} <\text{local data structure name}> \]
\[\text{lim} <\text{global entity name}> \]
\[<\text{lfbs slots}> \]
\[\text{primary-key} <\text{slot list}> \]
\[\text{candidate-key} <\text{slot list}> \]

\[<\text{slotlist} := <\text{slot name}> | <\text{slot list}> <\text{slot name}> \]

\[<\text{slotlists} := <\text{slotlist}> | <\text{slot list}> <\text{slot list}> \]

\[<\text{lfbs slots} := <\text{lfbs slot description}> | \]
\[<\text{lfbs slots} <\text{lfbs slot description}> \]

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

\[<\text{slottype} := \text{attribute} | \text{relationship} \]

\[<\text{datatypename} := \text{integer} | \text{real} | \text{decimal} | \text{number} | \text{character} | \text{string}[[\text{<integer>}}] | \text{boolean} \]

\[<\text{lfbs slots} := (<\text{local data structure name}> . <\text{local attribute name}> ) | \]
\[<\text{lfbs slots} (<\text{local data structure name}> . <\text{local attribute name}> ) \]

\[<\text{mapping table} := (<\text{local value}> . <\text{global value}> ) | \]
\[<\text{mapping table} (<\text{local value}> . <\text{global value}> ) \]

\[<\text{relationship type} := \text{is-a} | \text{part-of} | \text{instance} | \text{is-a-inv} | \text{subpart} | \text{instance-finv} \]

\[<\text{relationship range declarations} := <\text{entity range declaration}> | \]
\[<\text{relationship range declarations}> <\text{entity range declaration}> \]

\[<\text{entity range declaration} := \text{entity} <\text{entity name}> <\text{slot range declaration}> | \]
\[<\text{entity range declaration} <\text{slot range declaration}> \]

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

17 See Chapter 2 for definition of clause.
Technical Description of KADBASE

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
dependency 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

dependency 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.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:

- \textit{K2i&-message-name(component-name)} — returns a unique message name formed from the requesting component's name\footnote{Components are identified by unique three-character names; e.g., "kui" for KADBASE User Interface, "nda" for Network Data Access Manager, etc.} and a local time stamp.
- \textit{Kad'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.
- \textit{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.
- \textit{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 of KADBASE 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 contain 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.
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)
slot weight
    slot-type attribute
data-type real
    lfbm (w__shape . weight)
slot area
    slot-type attribute
data-type real
    lfbm (w__shape . area)
Technical Description of KADBASE

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
Technical Description of KADBASE

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)
slot weight
Technical Description of KADBASE

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)
lfm web-thickness

slot bf
slot-type attribute
data-type real
lfbm (w_shape . bf)
lfm flange-width

slot tf
slot-type attribute
data-type real
lfbm (w_shape . tf)
lfm 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)
Technical Description of KADBASE

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

;
<table>
<thead>
<tr>
<th>Attribute</th>
<th>Description</th>
<th>Data Type</th>
<th>Sources</th>
</tr>
</thead>
<tbody>
<tr>
<td>slot</td>
<td>sources (spx ais)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>weight</td>
<td>slot-type attribute</td>
<td>real</td>
<td></td>
</tr>
<tr>
<td>slot area</td>
<td>data-type real</td>
<td></td>
<td></td>
</tr>
<tr>
<td>slot depth</td>
<td>sources (spx ais)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>slot web-thickness</td>
<td>slot-type attribute</td>
<td>real</td>
<td></td>
</tr>
<tr>
<td>slot flange-width</td>
<td>data-type real</td>
<td></td>
<td></td>
</tr>
<tr>
<td>slot flange-thickness</td>
<td>slot-type attribute</td>
<td>real</td>
<td></td>
</tr>
<tr>
<td>slot k</td>
<td>data-type real</td>
<td></td>
<td></td>
</tr>
<tr>
<td>slot bf2tf</td>
<td>sources (spx ais)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>slot fyp</td>
<td>slot-type attribute</td>
<td>real</td>
<td></td>
</tr>
<tr>
<td>slot dtw</td>
<td>data-type real</td>
<td></td>
<td></td>
</tr>
<tr>
<td>slot fyppp</td>
<td>sources (spx ais)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>slot rt</td>
<td>slot-type attribute</td>
<td>real</td>
<td></td>
</tr>
<tr>
<td>slot daf</td>
<td>data-type real</td>
<td></td>
<td></td>
</tr>
<tr>
<td>slot ri</td>
<td>slot-type attribute</td>
<td>real</td>
<td></td>
</tr>
<tr>
<td>slot ra</td>
<td>data-type real</td>
<td></td>
<td></td>
</tr>
<tr>
<td>slot nt</td>
<td>slot-type attribute</td>
<td>real</td>
<td></td>
</tr>
<tr>
<td>slot moment-of-inertia-x</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
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 KBSI, and concludes with the SPEX post-processing.

CMULisp, 5-16-86, from Franz Lisp, Opus 38
1.(load 'spex)
[load spex.l]
[fasl /usrce0/jhg/framekit/rulekit.o]
[load /usrce0/jhg/framekit/general.1]
[load /usrce0/jhg/framekit/macro.1]
[load /usrce0/jhg/framekit/machacks.1]
[fasl /usrce0/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/cmutfpl.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 -1l77 -lc

[load /usr/jhg/thesis/spex/evk.s.l]
[load /usr/jhg/thesis/spex/post-pro.l]
[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/hch/kad/kadutil.o]
[fasl /usr/hch/kad/kui/parse/kui-scanner.o]
[fasl /usr/hch/kad/kadcomm.o]

/* WELCOME TO SPEX */
Paragraph 1

Paragraph 2

Paragraph 3

Paragraph 4

Paragraph 5

Paragraph 6

Paragraph 7

Paragraph 8

Paragraph 9

Paragraph 10

Paragraph 11

Paragraph 12

Paragraph 13

Paragraph 14

Paragraph 15

Paragraph 16

Paragraph 17

Paragraph 18

Paragraph 19

Paragraph 20

Paragraph 21

Paragraph 22

Paragraph 23

Paragraph 24

Paragraph 25

Paragraph 26

Paragraph 27

Paragraph 28

Paragraph 29

Paragraph 30

Paragraph 31

Paragraph 32

Paragraph 33

Paragraph 34

Paragraph 35

Paragraph 36

Paragraph 37

Paragraph 38

Paragraph 39

Paragraph 40

Paragraph 41

Paragraph 42

Paragraph 43

Paragraph 44

Paragraph 45

Paragraph 46

Paragraph 47

Paragraph 48

Paragraph 49

Paragraph 50

Paragraph 51

Paragraph 52

Paragraph 53

Paragraph 54

Paragraph 55

Paragraph 56

Paragraph 57

Paragraph 58

Paragraph 59

Paragraph 60

Paragraph 61

Paragraph 62

Paragraph 63

Paragraph 64

Paragraph 65

Paragraph 66

Paragraph 67

Paragraph 68

Paragraph 69

Paragraph 70

Paragraph 71

Paragraph 72

Paragraph 73

Paragraph 74

Paragraph 75

Paragraph 76

Paragraph 77

Paragraph 78

Paragraph 79

Paragraph 80

Paragraph 81

Paragraph 82

Paragraph 83

Paragraph 84

Paragraph 85

Paragraph 86

Paragraph 87

Paragraph 88

Paragraph 89

Paragraph 90

Paragraph 91

Paragraph 92

Paragraph 93

Paragraph 94

Paragraph 95

Paragraph 96

Paragraph 97

Paragraph 98

Paragraph 99

Paragraph 100
Technical Description of KADBASE

(frame:)

(deffact constraint-ct-def
(frame: 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:)

(deffact constraint-h-def
(frame: 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:)

(deffact constraint-Af-def
(frame: 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:)

(deffact constraint-Ix-def
(frame: 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:)

(deffact constraint-Sxt-def
(frame: 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

(frame:)

(frame:)

58
Technical Description of KAD BASE

(frame:)
(value constraint-ct-def constraint-Ix-def))

(defprop 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:)
(value 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
)
)
(expr-type (value numeric))

(frame:)
(value 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:)
(value 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:)
(value 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 < « «
(\(\phi\) -
(\(\phi\) - 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))
Technical Description of KADBAS

(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-c1
 (constraint-LFB-AVERAGE-c1
 (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
Technical Description of KADBASE

The following message was just posted: SATISFY_CONSTRAINT_SET

The query frame is:

```
(defprop query-frame (reply
  (if-added (map-database-results-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 // (2 * 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))))
```
Calling KADBAS(E/Ingres Database

I 1 <Enter> spex-kbsi (query-frame)
I |1 <Enter> spex-kbsi-syntax-query (query-frame)
I | |1 <Enter> spex-kbsi-syntax-and (((2400.0 <= (0.9 * & * &)) (bf / & <= 147 / sqrt ...)) (bf / & > 65 / sqrt ...))
I | | |1 <Enter> spex-kbsi-syntax-and ( ((bf / & <= 147 / sqrt ...) (bf / & > 65 / sqrt ...))
I | | | |1 <Enter> spex-kbsi-syntax-and (((bf / & > 65 / sqrt ...) (bf / (2 * tf)) > 65 / sqrt ...))
I | | | | |1 <EXIT> spex-kbsi-syntax-and ( ((bf / (2 * tf)) > 65 / sqrt ...))
I | | | | | |1 <EXIT> spex-kbsi-syntax-and (((2400.0 <= (0.9 * (1.0) * (6 - & * &)) (bf / (2 * tf)) <= 147 / sqrt ...))
I | | | | | | |1 <EXIT> spex-kbsi-syntax-and ( (bf / (2 * tf)) > 65 / sqrt ...))
I | |1 <EXIT> spex-kbsi-syntax-query spx-retrieve-g00013
I | |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.
I |1 <EXIT> spex-kbsi-send t
I |1 <Enter> spex-kbsi-receive nil
KAD-WAIT-FOR-MESSAGE: The system name spx[load /../ce/usr/hch/kad/db/aisc/message/in/ais520368635]
KAD-LOAD-MESSAGE: The message name is ais520368635
I |1 <EXIT> spex-kbsi-receive ais-reply-g00005
KAD-GS-LFBS-REPLY: Translating reply.
I |1 <Enter> spex-kbsi-syntax-reply (ais-reply-g00005 query-frame)
I |1 <EXIT> spex-kbsi-syntax-reply ((311.0 1.06 14.38 12.67 1.735 1.735 ...))
KAD-DELETE-MESSAGE: Deleting message ais520368635 for system spx
rm: /../ce/usr/hch/kad/spex/message/in/ais520368639 nonexistent
rm: override protection 644 for /usr/hch/kad/spex/message/in/ais520368635? y
I |1 <EXIT> spex-kbsi t

The following message was just posted: SET_SOLUTION_SUCCESSFUL
I 3 <Enter> do-all-actions ((debug-output1) (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:)

62
Technical Description of KADBASE

(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:)


Technical Description of KADBASE

(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 2.7638235733513))
   (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))
   (value (value 0.0))
   (status (value binding)))
  frame:)

64
Technical Description of KADBASE

(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))))

The following message was just posted: FINAL_SOLUTION_FOUND
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-FOR-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

<table>
<thead>
<tr>
<th>zx</th>
<th>depth</th>
<th>tw</th>
<th>bf</th>
<th>tf</th>
<th>area</th>
<th>designation</th>
</tr>
</thead>
<tbody>
<tr>
<td>77.9001</td>
<td>12.0601</td>
<td>0.345</td>
<td>9.995</td>
<td>0.575</td>
<td>15.600</td>
<td>W12x53</td>
</tr>
</tbody>
</table>
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
Technical Description of KADBASE

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)
Technical Description of KADBSE

<table>
<thead>
<tr>
<th>Slot Name</th>
<th>Slot Type</th>
<th>Data Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>grade-elevation</td>
<td>attribute</td>
<td>number</td>
<td>lfbm (concrete-footing . grade-elevation)</td>
</tr>
<tr>
<td>top-elevation</td>
<td>attribute</td>
<td>number</td>
<td>lfbm (concrete-footing . top-elevation)</td>
</tr>
<tr>
<td>wall-thickness</td>
<td>attribute</td>
<td>number</td>
<td>lfbm (concrete-footing . wall-thickness)</td>
</tr>
<tr>
<td>rock-elevation</td>
<td>attribute</td>
<td>number</td>
<td>lfbm (concrete-footing . rock-elevation)</td>
</tr>
<tr>
<td>long-bar-size</td>
<td>attribute</td>
<td>number</td>
<td>lfbm (concrete-footing . long-bar-size)</td>
</tr>
<tr>
<td>long-bar-spacing</td>
<td>attribute</td>
<td>number</td>
<td>lfbm (concrete-footing . long-bar-spacing)</td>
</tr>
<tr>
<td>long-bar-pieces</td>
<td>attribute</td>
<td>number</td>
<td>lfbm (concrete-footing . long-bar-pieces)</td>
</tr>
<tr>
<td>long-bar-lbs-lnft</td>
<td>attribute</td>
<td>number</td>
<td>lfbm (concrete-footing . long-bar-lbs-lnft)</td>
</tr>
<tr>
<td>long-bar-lbs-ftg</td>
<td>attribute</td>
<td>number</td>
<td>lfbm (concrete-footing . long-bar-lbs-ftg)</td>
</tr>
<tr>
<td>short-bar-size</td>
<td>attribute</td>
<td>number</td>
<td>lfbm (concrete-footing . short-bar-size)</td>
</tr>
<tr>
<td>short-bar-spacing</td>
<td>attribute</td>
<td>number</td>
<td>lfbm (concrete-footing . short-bar-spacing)</td>
</tr>
<tr>
<td>short-bar-pieces</td>
<td>attribute</td>
<td>number</td>
<td>lfbm (concrete-footing . short-bar-pieces)</td>
</tr>
<tr>
<td>quality-of-concrete</td>
<td>attribute</td>
<td>symbol</td>
<td>lfbm (concrete-footing . quality-of-concrete)</td>
</tr>
<tr>
<td>footing-forms-selection</td>
<td>attribute</td>
<td>selection</td>
<td>lfbm (concrete-footing . footing-forms-selection)</td>
</tr>
<tr>
<td>earth-excavation-selection</td>
<td>attribute</td>
<td>selection</td>
<td>lfbm (concrete-footing . earth-excavation-selection)</td>
</tr>
</tbody>
</table>
Technical Description of KADBASE

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 reinforcing-steel-selection
slot-type attribute
data-type symbol
lfbm (concrete-footing . reinforcing-steel-selection)
slot anchor-bolts-and-base-plates-selection
slot-type attribute
data-type symbol
lfbm (concrete-footing . anchor-bolts-and-base-plates-selection)

candidate-key instance-name
candidate-key component-id
primary-key component-id

entity retaining-wall
lfbm retaining-wall
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 . component-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)
Technical Description of KADBASE

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-selection
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-selection)
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)
candidate-key instance-name
candidate-key component-id
primary-key component-id
entity structural-metal
lfbm structural-metal
Technical Description of KADBASE

<table>
<thead>
<tr>
<th>Slot Instance-Name</th>
<th>Slot Type Attribute</th>
<th>Data Type Symbol</th>
<th>Ifbm (Structural-Metal . Instance-Name)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Slot Component-ID</td>
<td>Slot Type Attribute</td>
<td>Data Type Symbol</td>
<td>Ifbm (Structural-Metal . Component-ID)</td>
</tr>
<tr>
<td>Slot Quantities</td>
<td>Slot Type Relationship</td>
<td>Data Type Symbol</td>
<td>Ifbm (Structural-Metal . Quantities)</td>
</tr>
<tr>
<td>Relationship-Type</td>
<td>Sub-Part</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Slot Number</td>
<td>Slot Type Attribute</td>
<td>Data Type Number</td>
<td>Ifbm (Structural-Metal . Number)</td>
</tr>
<tr>
<td>Slot Pieces-Per-Location</td>
<td>Slot Type Attribute</td>
<td>Data Type Number</td>
<td>Ifbm (Structural-Metal . Pieces-Per-Location)</td>
</tr>
<tr>
<td>Slot First-Dimension</td>
<td>Slot Type Attribute</td>
<td>Data Type Number</td>
<td>Ifbm (Structural-Metal . First-Dimension)</td>
</tr>
<tr>
<td>Units Inches</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Slot Second-Dimension</td>
<td>Slot Type Attribute</td>
<td>Data Type Number</td>
<td>Ifbm (Structural-Metal . Second-Dimension)</td>
</tr>
<tr>
<td>Units Inches</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Slot Thickness</td>
<td>Slot Type Attribute</td>
<td>Data Type Number</td>
<td>Ifbm (Structural-Metal . Thickness)</td>
</tr>
<tr>
<td>Units Inches</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Slot Length</td>
<td>Slot Type Attribute</td>
<td>Data Type Number</td>
<td>Ifbm (Structural-Metal . Length)</td>
</tr>
<tr>
<td>Units Feet</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Slot Weight</td>
<td>Slot Type Attribute</td>
<td>Data Type Number</td>
<td>Ifbm (Structural-Metal . Weight)</td>
</tr>
<tr>
<td>Units Lbs/Lnft</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Slot Fireproofing-Thickness</td>
<td>Slot Type Attribute</td>
<td>Data Type Number</td>
<td>Ifbm (Structural-Metal . Fireproofing-Thickness)</td>
</tr>
<tr>
<td>Units Inches</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Slot Anchor-Bolt-Diameter</td>
<td>Slot Type Attribute</td>
<td>Data Type Number</td>
<td>Ifbm (Structural-Metal . Anchor-Bolt-Diameter)</td>
</tr>
<tr>
<td>Units Inches</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Slot Anchor-Bolt-Length</td>
<td>Slot Type Attribute</td>
<td>Data Type Number</td>
<td>Ifbm (Structural-Metal . Anchor-Bolt-Length)</td>
</tr>
<tr>
<td>Units Feet</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Slot Type-Of-Metal-Selection</td>
<td>Slot Type Attribute</td>
<td>Data Type Number</td>
<td>Ifbm (Structural-Metal . Type-Of-Metal-Selection)</td>
</tr>
<tr>
<td>Slot Type-Of-Member-Selection</td>
<td>Slot Type Attribute</td>
<td>Data Type Symbol</td>
<td>Ifbm (Structural-Metal . Type-Of-Member-Selection)</td>
</tr>
<tr>
<td>Slot Erection-Method-Selection</td>
<td>Slot Type Attribute</td>
<td>Data Type Symbol</td>
<td>Ifbm (Structural-Metal . Erection-Method-Selection)</td>
</tr>
</tbody>
</table>

74
Technical Description of KADBASE

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
  lim (cf_att . ID)
lfbm (cf_att . ID)
lfbm (cf_att . component-id)
slot A
  slot-type attribute
data-type integer
  lim (cf_att . A)
lfbm (cf_att . A)
lfbm (cf_att . number)
slot B
  slot-type attribute
Technical Description of KADBASE

data-type real
lfbm (cf_att . B)
lm length
units feet
slot C
slot-type attribute
data-type real
lfbm (cf_att . C)
lm width
units feet
slot D1
slot-type attribute
data-type real
lfbm (cf_att . D)
lm footing-thickness
units feet
slot D2
slot-type attribute
data-type real
lfbm (cf_att . D)
lm height-of-stepped-footing
units feet
slot E
slot-type attribute
data-type real
lfbm (cf_att . E)
lm grade-elevation
units feet
slot F
slot-type attribute
data-type real
lfbm (cf_att . F)
lm top-elevation
units feet
slot G
slot-type attribute
data-type real
lfbm (cf_att . G)
lm wall-thickness
units feet
slot H
slot-type attribute
data-type real
lfbm (cf_att . H)
lm rock-elevation
units feet
slot J1
slot-type attribute
data-type real
lfbm (cf_att . J)
lm long-bar-pieces
mapped-type integer
direct-function (lambda (x) (fix x))
slot J2
slot-type attribute
data-type real
lfbm (cf_att . J)
lm 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)
lm 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
Technical Description of KADBASE

1. **lfbm (cf_att . J)**
   - **lim long-bar-lbs-lnft**
   - **units lbs/lnft**
   - **slot J5**
     - **slot-type attribute**
     - **data-type real**

2. **lfbm (cf_att . J)**
   - **lim long-bar-lbs-ftg**
   - **units lbs**
   - **slot J1**
     - **slot-type attribute**
     - **data-type real**

3. **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_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 SS9**
     - **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 SS8**
     - **slot-type attribute**
     - **data-type integer**

4. **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"))`
Technical Description of KADBASE

(4. "form column ftg rock")
(5. "stepped ftg forms")
(6. "pilaster wall ftg")
(7. "curved wall ftg")
(8. "pile cap forms")
(9. "wall ftg bulkhd.")

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

78
mapped-type string
mapping-table
(0 . "none required")
(1 . "column footings lbs/cu yd calc")
(2 . "wall footing 50 lfb/cu yd")
(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/ln ft 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
Technical Description of KADBASE

units feet

slot G1
  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 w/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

80
Technical Description of KADBASE

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 csr1 level bf- no surcharge")
 (2 . "index csr1 sloped bf- no surcharge")
 (3 . "index csr1 level bf- hwy surchg (light)")
 (4 . "index csr1 level bf- railwy surch (heavy)")
 (5 . "lbs/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")

81
Technical Description of KADBASE

(3 . "2x8 key in ftg")
(4 . "3x8 key in ftg")
(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."
(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)
Technical Description of KADBASE

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-strength 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 . SS7)
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"))
Technical Description of KADBASE

(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.
   (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 string10
  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
    units dollars
    lfbm (unit_cost . matlcost)
slot instcost
    slot-type attribute
data-type real
    units dollars
    lfbm (unit_cost . instcost)
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
Technical Description of KADBASE

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

date-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)

date-instance-name
  slot-type attribute
  data-type symbol
  sources (hie)

slot quantity-id
  slot-type relationship
  data-type symbol
  sources (hie)

slot numbers
  slot-type relationship
  data-type symbol
  sources (hie)

relationship-type subpart

entity construction-quantity
  sources (hie pmd)

entity concrete-footing
  sources (hie bdd)
Technical Description of KADBAS

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
Technical Description of KADBASE

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-selection
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-bolts-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
Technical Description of KADBASE

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 footing-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
Technical Description of KADBASE

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

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 versions 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 of concrete footings)

<table>
<thead>
<tr>
<th>id</th>
<th>a</th>
<th>b</th>
<th>c</th>
<th>d</th>
<th>e</th>
<th>f</th>
<th>g</th>
<th>h</th>
<th>i</th>
<th>j</th>
<th>k</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>111</td>
<td>150.000</td>
<td>3.0001</td>
<td>1.0001</td>
<td>98.6701</td>
<td>94.3301</td>
<td>1.0001</td>
<td>0.0001</td>
<td>0.0001</td>
<td>0.0001</td>
<td>4.050</td>
</tr>
<tr>
<td>2</td>
<td>12</td>
<td>5.0001</td>
<td>1.5001</td>
<td>1.0001</td>
<td>96.6701</td>
<td>94.3301</td>
<td>0.0001</td>
<td>0.0001</td>
<td>0.0001</td>
<td>0.0001</td>
<td>3.0501</td>
</tr>
<tr>
<td>3</td>
<td>13</td>
<td>0.0001</td>
<td>3.0001</td>
<td>1.3301</td>
<td>0.0001</td>
<td>0.0001</td>
<td>0.0001</td>
<td>0.0001</td>
<td>0.0001</td>
<td>0.0001</td>
<td>0.0001</td>
</tr>
<tr>
<td>4</td>
<td>14</td>
<td>8.0001</td>
<td>7.0001</td>
<td>1.5001</td>
<td>95.0001</td>
<td>97.5001</td>
<td>0.0001</td>
<td>0.0001</td>
<td>0.0001</td>
<td>0.0001</td>
<td>8.0601</td>
</tr>
</tbody>
</table>

cf_spec relation (describes the construction method for concrete footings)

<table>
<thead>
<tr>
<th>id</th>
<th>ss0</th>
<th>ss1</th>
<th>ss2</th>
<th>ss3</th>
<th>ss4</th>
<th>ss5</th>
<th>ss6</th>
<th>ss7</th>
<th>ss8</th>
<th>ss9</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
<td>5</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>31</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>6</td>
<td>0</td>
<td>3</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>31</td>
<td>6</td>
<td>0</td>
</tr>
<tr>
<td>3</td>
<td>5</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>51</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>4</td>
<td>1</td>
<td>5</td>
<td>4</td>
<td>0</td>
<td>4</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>4</td>
<td>7</td>
</tr>
</tbody>
</table>

rw_att relation (describes the attributes of retaining walls)

<table>
<thead>
<tr>
<th>id</th>
<th>a</th>
<th>b</th>
<th>c</th>
<th>d</th>
<th>e</th>
<th>f</th>
<th>g</th>
<th>h</th>
<th>i</th>
<th>j</th>
<th>k</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>31</td>
<td>30.0001</td>
<td>8.6701</td>
<td>0.670</td>
<td>j 1.0001</td>
<td>7.0001</td>
<td>1.2501</td>
<td>3.5001</td>
<td>0.0001</td>
<td>3.0001</td>
<td>0.0001</td>
</tr>
<tr>
<td>2</td>
<td>32</td>
<td>0.0001</td>
<td>0.0001</td>
<td>0.0001</td>
<td>0.0001</td>
<td>7.0001</td>
<td>3.3301</td>
<td>0.0001</td>
<td>0.0001</td>
<td>0.0001</td>
<td>0.0001</td>
</tr>
<tr>
<td>3</td>
<td>33</td>
<td>11154.0001</td>
<td>12.0001</td>
<td>1.0001</td>
<td>1.0001</td>
<td>6.5001</td>
<td>1.5001</td>
<td>3.0001</td>
<td>0.0001</td>
<td>2.3301</td>
<td>7.00001</td>
</tr>
<tr>
<td>4</td>
<td>34</td>
<td>82.0001</td>
<td>6.0001</td>
<td>0.3301</td>
<td>0.0001</td>
<td>0.0001</td>
<td>0.0001</td>
<td>0.0001</td>
<td>0.0001</td>
<td>0.0001</td>
<td>0.0001</td>
</tr>
</tbody>
</table>

rw_spec relation (describes the construction method for retaining walls)

<table>
<thead>
<tr>
<th>id</th>
<th>ss0</th>
<th>ss1</th>
<th>ss2</th>
<th>ss3</th>
<th>ss4</th>
<th>ss5</th>
<th>ss6</th>
<th>ss7</th>
<th>ss8</th>
<th>ss9</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>31</td>
<td>5</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>1</td>
<td>21</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>32</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>3</td>
<td>0</td>
</tr>
<tr>
<td>3</td>
<td>33</td>
<td>3</td>
<td>6</td>
<td>1</td>
<td>5</td>
<td>3</td>
<td>3</td>
<td>1</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>4</td>
<td>34</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>3</td>
<td>0</td>
<td>31</td>
</tr>
</tbody>
</table>
### Technical Description of KADBASE

#### sm_att relation (describes the attributes of structural metal)

<table>
<thead>
<tr>
<th>id</th>
<th>a</th>
<th>b</th>
<th>c</th>
<th>d</th>
<th>e</th>
<th>f</th>
<th>g</th>
<th>h</th>
<th>i</th>
<th>j</th>
<th>k</th>
</tr>
</thead>
<tbody>
<tr>
<td>211</td>
<td>21</td>
<td>5.0001</td>
<td>12.000</td>
<td>12.3001</td>
<td>0.0001</td>
<td>20.000</td>
<td>1120.000</td>
<td>1.5001</td>
<td>0.0001</td>
<td>0.000</td>
<td>0.000</td>
</tr>
<tr>
<td>221</td>
<td>41</td>
<td>10.0000</td>
<td>8.0000</td>
<td>8.0000</td>
<td>0.0001</td>
<td>12.500</td>
<td>48.0000</td>
<td>2.0001</td>
<td>0.0001</td>
<td>1.250</td>
<td>1.250</td>
</tr>
<tr>
<td>23</td>
<td>10</td>
<td>2.0001</td>
<td>6.0000</td>
<td>4.0000</td>
<td>0.3701</td>
<td>15.000</td>
<td>1.0000</td>
<td>0.0001</td>
<td>0.0001</td>
<td>0.000</td>
<td>0.000</td>
</tr>
<tr>
<td>24</td>
<td>21</td>
<td>4.0001</td>
<td>12.000</td>
<td>12.5001</td>
<td>0.5001</td>
<td>12.000</td>
<td>1.0000</td>
<td>0.0001</td>
<td>0.0001</td>
<td>0.750</td>
<td>1.500</td>
</tr>
</tbody>
</table>

#### sm_spec relation (describes the construction method for structural metal)

<table>
<thead>
<tr>
<th>lid</th>
<th>issO</th>
<th>ss4</th>
<th>ss5</th>
<th>ss6</th>
<th>ss7</th>
<th>ss8</th>
<th>ss9</th>
</tr>
</thead>
<tbody>
<tr>
<td>211</td>
<td>21</td>
<td>11</td>
<td>11</td>
<td>11</td>
<td>11</td>
<td>11</td>
<td>12</td>
</tr>
<tr>
<td>221</td>
<td>41</td>
<td>01</td>
<td>01</td>
<td>01</td>
<td>11</td>
<td>41</td>
<td>21</td>
</tr>
<tr>
<td>23</td>
<td>21</td>
<td>01</td>
<td>01</td>
<td>31</td>
<td>11</td>
<td>41</td>
<td>21</td>
</tr>
<tr>
<td>24</td>
<td>7</td>
<td>21</td>
<td>01</td>
<td>01</td>
<td>41</td>
<td>61</td>
<td>41</td>
</tr>
</tbody>
</table>

#### 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

<table>
<thead>
<tr>
<th>itemcode</th>
<th>itemdesc</th>
<th>itemunit</th>
<th>matlcost</th>
<th>instcost</th>
</tr>
</thead>
<tbody>
<tr>
<td>33100</td>
<td>13000 psi ready-mix concrete (03.3-12-015)</td>
<td>CUYD</td>
<td>48.900</td>
<td>0.0001</td>
</tr>
<tr>
<td>33101</td>
<td>12500 psi ready-mix concrete (03.3-12-010)</td>
<td>CUYD</td>
<td>47.250</td>
<td>0.0001</td>
</tr>
<tr>
<td>33102</td>
<td>13500 psi ready-mix concrete (03.3-12-020)</td>
<td>CUYD</td>
<td>50.550</td>
<td>0.0001</td>
</tr>
<tr>
<td>33103</td>
<td>13750 psi ready-mix concrete (03.3-12-025)</td>
<td>CUYD</td>
<td>51.500</td>
<td>0.0001</td>
</tr>
<tr>
<td>33104</td>
<td>14000 psi ready-mix concrete (03.3-12-030)</td>
<td>CUYD</td>
<td>52.200</td>
<td>0.0001</td>
</tr>
<tr>
<td>33120</td>
<td>Pour (wall) concrete ftg - direct (03.3-38-190)</td>
<td>CU ''YD</td>
<td>0.450</td>
<td>6.7001</td>
</tr>
<tr>
<td>33130</td>
<td>Pour (wall) concrete ftg - carts (03.3-38-562)</td>
<td>CUYD</td>
<td>1.150</td>
<td>6.0501</td>
</tr>
<tr>
<td>33151</td>
<td>Pour (wall) concrete ftg - pump (03.3-38-195)</td>
<td>CUYD</td>
<td>5.650</td>
<td>11.000</td>
</tr>
<tr>
<td>33140</td>
<td>Pour (wall) concrete ftg - crane (03.3-38-200)</td>
<td>CUYD</td>
<td>6.650</td>
<td>12.2001</td>
</tr>
<tr>
<td>33165</td>
<td>Pour (column) concrete ftg - direct (03.3-38-190)</td>
<td>CUYD</td>
<td>0.450</td>
<td>6.7001</td>
</tr>
<tr>
<td>33166</td>
<td>Pour (column) concrete ftg - carts (03.3-38-562)</td>
<td>CUYD</td>
<td>1.150</td>
<td>6.0501</td>
</tr>
<tr>
<td>33167</td>
<td>Pour (column) concrete ftg - pump (03.3-38-195)</td>
<td>CUYD</td>
<td>5.650</td>
<td>11.000</td>
</tr>
<tr>
<td>33168</td>
<td>Pour (column) concrete ftg - crane (03.3-38-200)</td>
<td>CUYD</td>
<td>6.650</td>
<td>12.2001</td>
</tr>
<tr>
<td>31013</td>
<td>Finish top of footing (03.3-26-001 floor finish)</td>
<td>SQFT</td>
<td>0.430</td>
<td>1.4401</td>
</tr>
<tr>
<td>31006</td>
<td>Wall ftg. forms - earth (03.1-45-005 2 use)</td>
<td>ISQFT</td>
<td>0.340</td>
<td>1.2401</td>
</tr>
<tr>
<td>31007</td>
<td>Column ftg. forms - earth (03.1-45-005 2 use)</td>
<td>ISQFT</td>
<td>0.910</td>
<td>2.9601</td>
</tr>
<tr>
<td>31008</td>
<td>Wall ftg. bulkhead form (03.1-55-100 1 use)</td>
<td>ISQFT</td>
<td>0.920</td>
<td>2.6701</td>
</tr>
<tr>
<td>31009</td>
<td>Add pilaster wall ftg form (03.1-65-862 2 use)</td>
<td>ISQFT</td>
<td>0.430</td>
<td>1.4401</td>
</tr>
<tr>
<td>31010</td>
<td>Curved wall ftg form (03.1-65-405 2 use)</td>
<td>ISQFT</td>
<td>0.340</td>
<td>1.2401</td>
</tr>
<tr>
<td>31003</td>
<td>Wall ftg. forms - rock (03.1-45-015 2 use)</td>
<td>ISQFT</td>
<td>0.430</td>
<td>1.4401</td>
</tr>
<tr>
<td>31014</td>
<td>Column ftg. forms - rock (03.1-45-015 2 use)</td>
<td>ISQFT</td>
<td>0.430</td>
<td>1.4401</td>
</tr>
<tr>
<td>31016</td>
<td>Pile cap forms (03.1-45-305 2 use)</td>
<td>ISQFT</td>
<td>1.240</td>
<td>2.1800</td>
</tr>
<tr>
<td>222033</td>
<td>Mach ftg. earth excavation (02.3-18-004 3')</td>
<td>CUYD</td>
<td>1.170</td>
<td>1.460</td>
</tr>
</tbody>
</table>
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, COMPDJ, 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

<table>
<thead>
<tr>
<th>qty_id</th>
<th>comp_id</th>
<th>limits</th>
<th>amount</th>
<th>cost</th>
</tr>
</thead>
<tbody>
<tr>
<td>333031</td>
<td>14 PCS</td>
<td>12.000</td>
<td>0.000</td>
<td></td>
</tr>
<tr>
<td>333021</td>
<td>14 PCS</td>
<td>24.000</td>
<td>0.000</td>
<td></td>
</tr>
<tr>
<td>333011</td>
<td>14 PCS</td>
<td>24.000</td>
<td>0.000</td>
<td></td>
</tr>
<tr>
<td>321011</td>
<td>14 TON</td>
<td>6.8921</td>
<td>0.000</td>
<td></td>
</tr>
<tr>
<td>2220411</td>
<td>14 CUYD</td>
<td>69.8271</td>
<td>0.000</td>
<td></td>
</tr>
<tr>
<td>2220191</td>
<td>14 CUYD</td>
<td>69.8271</td>
<td>0.000</td>
<td></td>
</tr>
<tr>
<td>222071</td>
<td>14 CUYD</td>
<td>20.000</td>
<td>0.000</td>
<td></td>
</tr>
<tr>
<td>2220331</td>
<td>14 CUYD</td>
<td>111.6541</td>
<td>0.000</td>
<td></td>
</tr>
<tr>
<td>31014</td>
<td>14 SQFT</td>
<td>540.000</td>
<td>0.000</td>
<td></td>
</tr>
<tr>
<td>331671</td>
<td>14 CUYD</td>
<td>37.3331</td>
<td>0.000</td>
<td></td>
</tr>
<tr>
<td>331021</td>
<td>14 CUYD</td>
<td>37.3331</td>
<td>0.000</td>
<td></td>
</tr>
<tr>
<td>310081</td>
<td>14 SQFT</td>
<td>15.9601</td>
<td>0.000</td>
<td></td>
</tr>
<tr>
<td>310061</td>
<td>14 SQFT</td>
<td>8.858</td>
<td>0.000</td>
<td></td>
</tr>
<tr>
<td>331201</td>
<td>14 CUYD</td>
<td>0.984</td>
<td>0.000</td>
<td></td>
</tr>
<tr>
<td>321011</td>
<td>14 TON</td>
<td>0.284</td>
<td>0.000</td>
<td></td>
</tr>
<tr>
<td>2220411</td>
<td>14 CUYD</td>
<td>45.3001</td>
<td>0.000</td>
<td></td>
</tr>
<tr>
<td>222014</td>
<td>14 CUYD</td>
<td>45.3001</td>
<td>0.000</td>
<td></td>
</tr>
<tr>
<td>222051</td>
<td>14 CUYD</td>
<td>0.867</td>
<td>0.000</td>
<td></td>
</tr>
<tr>
<td>2220361</td>
<td>14 CUYD</td>
<td>46.9661</td>
<td>0.000</td>
<td></td>
</tr>
<tr>
<td>310091</td>
<td>14 SQFT</td>
<td>48.0001</td>
<td>0.000</td>
<td></td>
</tr>
<tr>
<td>310061</td>
<td>14 SQFT</td>
<td>18.000</td>
<td>0.000</td>
<td></td>
</tr>
<tr>
<td>301361</td>
<td>14 SQFT</td>
<td>45.0001</td>
<td>0.000</td>
<td></td>
</tr>
<tr>
<td>331201</td>
<td>14 CUYD</td>
<td>1.667</td>
<td>0.000</td>
<td></td>
</tr>
<tr>
<td>331001</td>
<td>14 CUYD</td>
<td>1.667</td>
<td>0.000</td>
<td></td>
</tr>
<tr>
<td>321011</td>
<td>11 TON</td>
<td>2.144</td>
<td>0.000</td>
<td></td>
</tr>
<tr>
<td>310111</td>
<td>11 LNSFT</td>
<td>150.000</td>
<td>0.000</td>
<td></td>
</tr>
<tr>
<td>2220411</td>
<td>11 CUYD</td>
<td>149.6031</td>
<td>0.000</td>
<td></td>
</tr>
<tr>
<td>222041</td>
<td>11 CUYD</td>
<td>149.6031</td>
<td>0.000</td>
<td></td>
</tr>
<tr>
<td>222051</td>
<td>11 CUYD</td>
<td>5.050</td>
<td>0.000</td>
<td></td>
</tr>
<tr>
<td>2220361</td>
<td>11 CUYD</td>
<td>190.3811</td>
<td>0.000</td>
<td></td>
</tr>
<tr>
<td>310061</td>
<td>11 SQFT</td>
<td>300.0001</td>
<td>0.000</td>
<td></td>
</tr>
<tr>
<td>301361</td>
<td>11 SQFT</td>
<td>450.0001</td>
<td>0.000</td>
<td></td>
</tr>
<tr>
<td>331201</td>
<td>11 CUYD</td>
<td>16.6671</td>
<td>0.000</td>
<td></td>
</tr>
<tr>
<td>331001</td>
<td>11 CUYD</td>
<td>16.6671</td>
<td>0.000</td>
<td></td>
</tr>
<tr>
<td>333031</td>
<td>34 PCS</td>
<td>1.000</td>
<td>0.000</td>
<td></td>
</tr>
<tr>
<td>333021</td>
<td>34 PCS</td>
<td>2.0001</td>
<td>0.000</td>
<td></td>
</tr>
<tr>
<td>333011</td>
<td>34 PCS</td>
<td>2.000</td>
<td>0.000</td>
<td></td>
</tr>
<tr>
<td>2220411</td>
<td>34 CUYD</td>
<td>5.6921</td>
<td>0.000</td>
<td></td>
</tr>
<tr>
<td>222191</td>
<td>34 CUYD</td>
<td>5.6921</td>
<td>0.000</td>
<td></td>
</tr>
<tr>
<td>222071</td>
<td>34 CUYD</td>
<td>3.1111</td>
<td>0.000</td>
<td></td>
</tr>
<tr>
<td>222033</td>
<td>34 CUYD</td>
<td>10.5021</td>
<td>0.000</td>
<td></td>
</tr>
<tr>
<td>31014</td>
<td>34 SQFT</td>
<td>0.000</td>
<td>0.000</td>
<td></td>
</tr>
<tr>
<td>301361</td>
<td>34 SQFT</td>
<td>0.0001</td>
<td>0.000</td>
<td></td>
</tr>
<tr>
<td>33167</td>
<td>34 CUYD</td>
<td>0.000</td>
<td>0.000</td>
<td></td>
</tr>
<tr>
<td>331001</td>
<td>34 CUYD</td>
<td>0.000</td>
<td>0.000</td>
<td></td>
</tr>
<tr>
<td>333031</td>
<td>33 PCS</td>
<td>1.000</td>
<td>0.000</td>
<td></td>
</tr>
<tr>
<td>333021</td>
<td>33 PCS</td>
<td>2.0001</td>
<td>0.000</td>
<td></td>
</tr>
<tr>
<td>333011</td>
<td>33 PCS</td>
<td>2.0001</td>
<td>0.000</td>
<td></td>
</tr>
<tr>
<td>2220411</td>
<td>33 CUYD</td>
<td>72.2541</td>
<td>0.000</td>
<td></td>
</tr>
<tr>
<td>222191</td>
<td>33 CUYD</td>
<td>72.2541</td>
<td>0.000</td>
<td></td>
</tr>
</tbody>
</table>
### Technical Description of KADBASE

#### Project Management Database after Pricing

<table>
<thead>
<tr>
<th>qty_id</th>
<th>comp_id</th>
<th>units</th>
<th>amount</th>
<th>cost</th>
</tr>
</thead>
<tbody>
<tr>
<td>333031</td>
<td>14 PCS</td>
<td>I</td>
<td>12.0001</td>
<td>217.800</td>
</tr>
<tr>
<td>333021</td>
<td>14 PCS</td>
<td>I</td>
<td>24.0001</td>
<td>284.400</td>
</tr>
<tr>
<td>333011</td>
<td>14 PCS</td>
<td>I</td>
<td>24.0001</td>
<td>720.000</td>
</tr>
<tr>
<td>321011</td>
<td>14 TON</td>
<td>I</td>
<td>6.8921</td>
<td>5720.360</td>
</tr>
<tr>
<td>2220411</td>
<td>14 TON</td>
<td>I</td>
<td>69.827</td>
<td>114.156</td>
</tr>
<tr>
<td>222191</td>
<td>14 TON</td>
<td>I</td>
<td>69.827</td>
<td>754.830</td>
</tr>
<tr>
<td>222071</td>
<td>14 TON</td>
<td>I</td>
<td>20.000</td>
<td>640.000</td>
</tr>
<tr>
<td>222031</td>
<td>14 TON</td>
<td>I</td>
<td>53.333</td>
<td>1887.183</td>
</tr>
<tr>
<td>31014</td>
<td>14 SQFT</td>
<td>I</td>
<td>540.000</td>
<td>1003.800</td>
</tr>
<tr>
<td>301361</td>
<td>14 SQFT</td>
<td>I</td>
<td>672.000</td>
<td>114.240</td>
</tr>
<tr>
<td>311014</td>
<td>14 SQFT</td>
<td>I</td>
<td>37.333</td>
<td>621.594</td>
</tr>
<tr>
<td>311021</td>
<td>14 SQFT</td>
<td>I</td>
<td>37.333</td>
<td>1887.183</td>
</tr>
<tr>
<td>310081</td>
<td>13 SQFT</td>
<td>I</td>
<td>15.960</td>
<td>25.217</td>
</tr>
<tr>
<td>310061</td>
<td>13 SQFT</td>
<td>I</td>
<td>8.8581</td>
<td>16.564</td>
</tr>
<tr>
<td>313201</td>
<td>13 TON</td>
<td>I</td>
<td>0.984</td>
<td>7.036</td>
</tr>
<tr>
<td>321011</td>
<td>12 TON</td>
<td>I</td>
<td>0.2841</td>
<td>235.721</td>
</tr>
<tr>
<td>222041</td>
<td>12 TON</td>
<td>I</td>
<td>45.300</td>
<td>74.292</td>
</tr>
<tr>
<td>222041</td>
<td>12 TON</td>
<td>I</td>
<td>45.300</td>
<td>32.616</td>
</tr>
<tr>
<td>222051</td>
<td>12 TON</td>
<td>I</td>
<td>0.8671</td>
<td>13.785</td>
</tr>
<tr>
<td>222031</td>
<td>12 TON</td>
<td>I</td>
<td>46.966</td>
<td>123.521</td>
</tr>
<tr>
<td>310091</td>
<td>12 SQFT</td>
<td>I</td>
<td>48.000</td>
<td>185.760</td>
</tr>
<tr>
<td>310061</td>
<td>12 SQFT</td>
<td>I</td>
<td>18.000</td>
<td>33.660</td>
</tr>
<tr>
<td>313201</td>
<td>12 SQFT</td>
<td>I</td>
<td>45.000</td>
<td>7.650</td>
</tr>
<tr>
<td>331201</td>
<td>12 SQFT</td>
<td>I</td>
<td>1.667</td>
<td>11.919</td>
</tr>
<tr>
<td>33100</td>
<td>12 TON</td>
<td>I</td>
<td>1.667</td>
<td>81.516</td>
</tr>
<tr>
<td>321011</td>
<td>11 TON</td>
<td>I</td>
<td>2.1441</td>
<td>1779.520</td>
</tr>
<tr>
<td>310111</td>
<td>11 TON</td>
<td>I</td>
<td>150.000</td>
<td>55.500</td>
</tr>
<tr>
<td>222041</td>
<td>11 TON</td>
<td>I</td>
<td>149.603</td>
<td>245.349</td>
</tr>
<tr>
<td>222051</td>
<td>11 TON</td>
<td>I</td>
<td>149.603</td>
<td>107.714</td>
</tr>
<tr>
<td>2220361</td>
<td>11 TON</td>
<td>I</td>
<td>5.0501</td>
<td>80.295</td>
</tr>
<tr>
<td>310061</td>
<td>11 SQFT</td>
<td>I</td>
<td>190.381</td>
<td>509.072</td>
</tr>
<tr>
<td>3103161</td>
<td>11 SQFT</td>
<td>I</td>
<td>300.000</td>
<td>561.000</td>
</tr>
<tr>
<td>3103161</td>
<td>11 SQFT</td>
<td>I</td>
<td>450.000</td>
<td>76.500</td>
</tr>
</tbody>
</table>
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-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 = 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 == "deduct for 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(quantity-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 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 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 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 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 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 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 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 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(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-method6
if (concrete-placing-method-selection == "column ftg pour concrete w/carts")
then (quantity-id = 33166,
    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-quantity(quantity-id,units,amount)
)
Technical Description of KADBASE

{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 concrete-footing-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-footing-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-footing-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-footing-wall-and-column-footing-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)
  )
}

103
Technical Description of KADBASE

(take-off-rule concrete-footing-wall-and-column-footing-forms)
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-footing-forms)
if (footing-forms-selection == "stepped ftg forms")
then (quantity-id = 31006,
units = "SQFT",
amount = number * (footing-thickness * (1 + footing-thickness / 2)),
make-quantity(quantity-id, units, amount),
quantity-id = 31008,
units = "SQFT",
amount = number * (width * footing-thickness),
make-quantity(quantity-id, units, amount))

(take-off-rule concrete-footing-wall-and-column-footing-forms)
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-footing-forms)
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-footing-forms)
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-footing-wall-and-column-footing-forms)
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-excavation

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(quantity-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-excavation5

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 = "CUYD";
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)
)
Technical Description of KADBASE

(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-footing-trench-bracing-selection)
if (trench-bracing-selection == "solid wood col. ftg")
then (quantity-id = 24101,
units = "CUYD",
amount = number * (2 * (length +2) + 2 * (width +2)
* (grade-elevation - top-elevation + footing-thickness)),
make-quantity(quantity-id,units,amount)
)

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

;;; Rules for backfill

(take-off-rule concrete-footing-backfill-selection)
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) )


Technical Description of KADBAS

{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-footing-backfill-selection4
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)
)
Technical Description of KADBASE

(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-selection7
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)
}
Technical Description of KADBASE

(take-off-rule concrete-footing-backfill-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(quantity-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-quantity(quantity-id,units,amount),
)

(take-off-rule concrete-footing-backfill-selection10

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)
)
Technical Description of KADBASE

;;; 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)
)
}
### Technical Description of KADBASE

#### Rules for reinforcing steel

```plaintext
;; 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-footing-reinforcing-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-reinforcing-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-footing-reinforcing-steel-selection-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)
)
```

Technical Description of KADBASE

{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-footing-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-footing-reinforcing-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-reinforcing-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) }

}
Technical Description of KADBASE

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

{take-off-rule concrete-footing-reinforcing-steel-selection10
  if (reinforcing-steel-selection == "lbs/footing column footing with j dim")
  then (quantity-id = 32101,
         units = "TONT",
         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-footing-anchor-bolts-and-base-plates-selection1
  if (anchor-bolts-and-base-plates-selection == "set 2ab")
  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))
}

{take-off-rule concrete-footing-anchor-bolts-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-footing-anchor-bolts-and-base-plates-selection3
  if (anchor-bolts-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-footing-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-bolts-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-footing-anchor-bolts-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-amount1
  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-wall-quality-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-concrete0
  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-concrete1
  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-concrete1
  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) )
)
Technical Description of KADBAS

(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-footing-form-and-finish-selection1
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))
)
Technical Description of KADBASE

(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-excavation1
  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)
  )
)
Technical Description of KADBASE

(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 
               * (footing-depth + footing-thickness) / 3.0) 
               + 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-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 
               * (footing-depth + footing-thickness) / 3.0) 
               + 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 
               * (footing-depth + footing-thickness) / 3.0) 
               + 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
    + 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
    + 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
    + 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-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-wall-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-wall-trench-backfill-selection
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)
  )
)
### Technical Description of KADBASE

```plaintext
;; Rules for retained backfill

(take-off-rule retaining-wall-retained-backfill-selection1
  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)
  )
```

129
Technical Description of KADBASE

{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) ,
      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-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-selection1
  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) }
Technical Description of KADBASE

(take-off-rule retaining-wall-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-reinforcing-steel-selection4
  if (reinforcing-steel-selection == "index crsi level bf- railway 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-wall-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-foothing-key-and-constr-joints-selection1
  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-foothing-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)
  )
)}
Technical Description of KADBASE

(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-selection7
  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-selection1
  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-selection7
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-selection1
  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-metal-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-selection4
  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.)"
  )
  then (amount = number * pieces-per-location * (length * 2 * (first-dimension + second-dimension)
                           * thickness * 0.0034)
    )
)

(take-off-rule structural-metal-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-strength 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-strength 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 structural-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-selection11
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-selection12
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-selection1
  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-metal-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.)"
  )
  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)
  )
)
Technical Description of KADBASE

{take-off-rule structural-metal-type-of-metal-selection9
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-selection10
if (type-of-metal-selection == "add hi-strength 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-selection11
if (type-of-metal-selection == "add hi-strength 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-selection12
if (type-of-metal-selection == "add hi-strength 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-strength 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-selection14
if (type-of-metal-selection == "add hi-strength 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)
)
}
Technical Description of KADBASE

{take-off-rule structural-metal-type-of-metal-selection15
 if (  type-of-metal-selection == "add hi-strength 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-selection16
 if (  type-of-metal-selection == "add hi-strength 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-selection17
 if (  type-of-metal-selection == "add hi-strength 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-selection18
 if (  type-of-metal-selection == "add hi-strength steel"
 and type-of-member-selection == "round rod"")
 then (quantity-id = 51050 ,
 units = "TON",
 make-quantity(quantity-id,units,amount) ,
 quantity-id = 51050 ,
 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) 
 )
}
Technical Description of KADBASE

```plaintext
(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-metal-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-selection26
  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-metal-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 structural-metal-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 structural-metal-anchor-bolts-selection1
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-metal-anchor-bolts-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) ,
)}
This chapter is intended to supplement the description of the JADE-KADB BASE 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 joint1
      slot-type relationship
      data-type symbol
      lfbm (member . joint1)
      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 shear1
      slot-type attribute
      data-type number
      lfbm (member . shear1)
      lim FY1
      units kips
    slot axial1
      slot-type attribute
      data-type number
      lfbm (member . axial1)
      lim FX1
      units kips
    slot moment1
      slot-type attribute
      data-type number
      lfbm (member . moment1)
      lim MZ1
      units kip-inches
    slot fixity1
      slot-type attribute
      data-type string
      lfbm (member . fixity1)
    slot connection1
      slot-type relationship
      data-type symbol
      lfbm (member . connection1)
      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 joint1
  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
Technical Description of KADBASE

data-type integer
lfbm (member . mid)
lim member-id
slot joint1
  slot-type relationship
data-type symbol
lfbm (member . joint1)
relationship-type is-a
slot fixity1
  slot-type attribute
data-type string
lfbm (member . fixity1)
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 joint1
  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
Technical Description of KADBASE

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 x1
  slot-type attribute
data-type number
  units feet
slot y1
  slot-type attribute
data-type number
  units feet
slot z1
  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
Technical Description of KADBASE

slot-type attribute
data-type integer
sources (jad scd ard)

slot joint1
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 joint1
sources (jad)

slot yl
slot-type attribute
data-type number
units feet
inherited-through joint1
sources (jad)

slot zl
slot-type attribute
data-type number
units feet
inherited-through joint1
sources (jad)

slot fixity1
slot-type attribute
data-type string
sources (jad scd)

slot connection1
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
Technical Description of KADBAS

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
Technical Description of KADBASE

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 joint
  range joint
slot joint2
  range joint

entity joint
slot joined-members
  range member

entity joint
slot x1
  constraint (joint.x1 == joint.x)
slot y1
  constraint (joint.y1 == joint.y)
slot z1
  constraint (joint.z1 == 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 ==
sqrt1 (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.z1) / member.length)
;
References


