PROCESS MODELING LANGUAGE SPECIFICATION

Authorization: Subcontract No. P.O. TTM 748357

Revised

Date: 13 January 1995

PROCESS MODELING LANGUAGE SPECIFICATION

Authorization: Subcontract No. P.O. TTM 748357

Prepared By:

Grubel, David E.

Information Requirements and Analysis

Advanced Information Engineering

Approved By:

F. L. Bsharah J. F. Willis (Concurrence)

Project Engineer Manager

Information Requirements and Analysis Advanced Engineering Systems

Advanced Information Engineering

Revised

Date: 13 January 1995

No. of pages: i through ii + 22

+ A-1

Revision Record

Basic 02 Sep 1994 Initial Release

A 13 Jan 1995 Revised, Optimized, and Included Additional Capability.

FOREWORD

This report was prepared by the Information Requirements and Analysis Group, Advanced Information Engineering, Rockwell Aerospace - North American Aircraft, Seal Beach, California, in support of Subcontract Number P.O. TTM 748357.

ABSTRACT

To promote transferability of workflows, a tool independent language is needed for process flow definition. The Process Modeling Language (PML) defined in this specification describes models built using an extended version of the Integrated Computer-Aided Manufacturing DEFinition Number 3 language (IDEF-3x). This specification provides a "template" for importing and exporting process models (workflows) to and from model repositories. The Process Modeling Language (PML) grammar will be expressed in the Backus Naur Form (BNF).

TABLE OF CONTENTS

1.0 Introduction

The Process Modeling Language (PML) is a language for describing models built using an extended version of the Integrated Computer-Aided Manufacturing DEFinition Number 3 language (IDEF-3x). This specification is needed because a "template" is required for importing and exporting process models (workflows) to and from model repositories. The PML will be defined in the Backus-Naur Form (BNF). The unique feature of the PML is that it is a tool independent language. Any robust model repository must be able to seamlessly import or export PML files. This has the benefit of freeing the workflow implementation tools from being dependent upon (held hostage to) any workflow definition tool.

This document contains the formal definition of the PML grammar expressed in Backus-Naur form (BNF). The PML's statements consist of keywords and user-specified values (names, identifiers, descriptions, etc.). Certain statements describe IDEF-3x models and diagrams, and the activities and input, control, output, and mechanism objects (ICOMs) that are depicted on these models and diagrams. Other statements document the definitions of the activities and ICOMs. These two sets of statements are placed in separate files: a model PML file and a glossary PML file. The model PML is the textual file that describes a process flow model, diagrams, activities, junctions, and ICOMs and the links between these different elements. The glossary PML is the textual file that allows for the definition of each element that was defined in the model PML.

This document is intended for model authors who create PML files and for software developers who create tools to read or write PML files.

The BNF grammar statements are grouped in a hierarchical fashion in Section 2 to show the logical decomposition of the language. Section 3 contains an alphabetical listing of the collected BNF grammar statements as defined in Section 2. Section 4 contains several general notes (guidelines) related to specific BNF grammar statements defined in Section 2 and 3. Section 5 discusses the outstanding issues and concerns associated with this release of the PML specification. Section 6 contains a set of example PML statements for a given process flow model and its glossary. While the set does not include every variation of each PML statement, it does provide at least one typical example of most.

_

<u>**2.0 BNF Grammar of PML**</u>

Each BNF grammar statement consists of the name of a language component, followed by the definition of that component. A definition consists of literals or the names of other language components, which are defined in other (usually subsequent) statements. In addition, the following punctuation marks are used to structure the BNF statements:

`::=` separates the name of a language component from its definition.

`[ ]` describes sentences that are either empty or

of the form contained within the brackets (<u>optional</u>).

`{ }` describes sentences that consist of zero or more sentences of the form

contained within the braces (<u>repetition</u>).

`{ }*` If a "*" (asterisk) follows the bracketed group, the repetition is zero or

`{ }+` many occurances. If a "+" (plus) follows the bracketed group, the

repetition is one or many occurances.

`|` separates alternatives within a group (exclusive or).

These punctuation marks are part of BNF syntax, not PML syntax, so they must not appear in any PML statements. However, the parentheses and commas that are shown in some PML statements are part of PML syntax and must be included (this will be apparent). Keywords may be any mixture of upper and lower case when they appear in actual PML statements.

A PML statement is not limited to a single line or logical record. One statement may span several lines, and several statements may be placed on the same line (this is <u>not</u> recommended, however). Keywords and user-specified values in PML statements are separated by white space consisting of one or more blanks, tabs, line feeds, or carriage returns.

Comment lines may be interspersed anywhere among the lines that contain PML statements. A comment is any group of characters starting with a forward slash (/) and an asterisk (*) and ending with an asterisk (*) and a forward slash (/). This is shown in the example on the following page.

```
ACTIVITY 1 "Conduct Credit Check"

/* this section describes the activities ICOMs */

INPUT
New*Loan-Evaluation-Worksheet
Unapproved*Loan-Application
END
CONTROL
Approved*Application-Procedures
END
MECHANISM
Qualified*Loan-Officer
END
OUTPUT
In-Work*Loan-Application
Completed*Loan-Evaluation-Worksheet
END
END /* end of activity */
```

Comment lines are not part of the PML description of IDEF-3x models, so PML software tools can ignore them.

_

2.1 PML File

```
<pml-file> ::= <model> [ <glossary> ] [ <end-keyword> ]
```

2.2 Model

```
<model> ::= <model-stmt> [ <model-body> ]

<model-body> ::= <gen-info-stmt> <model-gen-info>
{ <diagram-block> } +

<model-gen-info> ::= <creation-date-stmt> [ <revision-date-stmt> ]
[ <author-stmt> ] [ <description-stmt> ]
[ <scope-stmt> ] [ <purpose-stmt> ]
[ <viewpoint-stmt> ]
```

2.3 Diagram

```
<diagram-block> ::= <diagram-stmt> <diagram-body> [ <end-keyword> ]

<diagram-body> ::= <function-block> { <diagram-body> } *

<function-block> ::= <activity-block>
| <junction-block>
| <notif-block>
```

2.4 Activity Block

```
<activity-block> ::= <activity-stmt> [ <activity-location-stmt> ]
[ <activity-body> ] [ <end-keyword> ]
```

```
<activity-body> ::= ( <activity-icom-block> | <destination-block> )
{ <activity-body> } *

<activity-icom-block> ::= <activity-input-block>
| <activity-control-block>
| <activity-output-block>
| <activity-mech-block>
| <activity-notif-block>

<destination-block> ::= <destination-keyword> ( <activity-stmt>
| <junction-stmt> )

<activity-input-block> ::= <input-keyword> { <activity-icom> } +
[ <end-keyword> ]

<activity-control-block> ::= <control-keyword> { <activity-icom> } +
[ <end-keyword> ]

<activity-output-block> ::= <output-keyword>
{ <activity-icom> [ <destination-block> ] } +
[ <end-keyword> ]

<activity-mech-block> ::= <mech-keyword> { <activity-icom> } +
[ <end-keyword> ]

<activity-icom> ::= <icom-id> [ <timing-stmt> ]

<activity-notif-block> ::= <notif-keyword> <activity-notif>
[ <end-keyword> ]

<activity-notif> ::= <activity-notif-stmt> [ <timing-stmt> ]
{ <activity-notif> } *
```

2.5 Junction Block

```
<junction-block> ::= <junction-stmt> [ <junction-location-stmt> ]
[ <junction-body> ] [ <end-keyword> ]

<junction-body> ::= ( <junction-icom-block> | <destination-block> )
{ <junction-body> } *

<junction-icom-block> ::= <junction-input-block>
| <junction-output-block>
| <function-notif-block>

<junction-input-block> ::= <input-keyword> { <icom-id> } +
[ <end-keyword> ]

<junction-output-block> ::= <output-keyword>
{ <icom-id> [ <destination-block> ] } +
[ <end-keyword> ]

<junction-notif-block> ::= <notif-keyword> <junction-notif>
[ <end-keyword> ]

<junction-notif> ::= <junction-notif-stmt> { <junction-notif> } *
```

2.6 Notification Block

```
<notif-block> ::= <notiflist-stmt> [ <notif-location-stmt> ]
[ <notif-body> ] [ <end-keyword> ]

<notif-body> ::= <notif-message-block> <notif-receiver-block>
```

```
<notif-message-block> ::= <notif-message-stmt> [ <end-keyword> ]

<notif-receiver-block> ::= <receiver-keyword> <notif-receiver>
[ <end-keyword> ]

<notif-receiver> ::= { <receiver-name> } +
```

_

2.7 Glossary

```
<glossary> ::= { <element-def> } +

<element-def> ::= <activity-def> [ <icom-def> ]
```

2.8 Activity Definitions

```
<activity-def> ::= <glossary-activity-stmt> [ <description-stmt> ]
```

2.9 Object Definitions

```
<icom-def> ::= <icom-def-body> { <icom-def> } +

<icom-def-body> ::= <glossary-icom-stmt> <icom-type-stmt>
[ <description-stmt> ]
```

2.10 PML Statements

```
<activity-location-stmt> ::= <coordinates-keyword> <x-id> , <y-id>

<activity-notif-stmt> ::= <notif-name>

<activity-stmt> ::= <activity-keyword> <activity-no> <activity-name>

<author-stmt> ::= <author-keyword> <text-line>

<creation-date-stmt> ::= <creation-keyword> [ <date-keyword> ] [ <date> ]

<description-stmt> ::= <description-keyword> <text-line>

<diagram-stmt> ::= <diagram-keyword> <diagram-no> <diagram-name>

<gen-info-stmt> ::= <general-keyword> [ <information-keyword> ]

<glossary-activity-stmt> ::= <activity-keyword> <activity-no> <activity-name>

<glossary-icom-stmt> ::= <icom-keyword> <icom-id>

<icom-type-stmt> ::= <input-keyword>
| <output-keyword>
| <control-keyword>
| <mech-keyword>

<junction-location-stmt> ::= <coordinates-keyword> <x-id> , <y-id>

<junction-notif-stmt> ::= <notif-name>

<junction-stmt> ::= <junction-keyword> <junction-no> <junction-name>
```

```
<model-stmt> ::= <model-keyword> <model-name>

<notiflist-stmt> ::= <notiflist-keyword> <notiflist-no>

<notif-location-stmt> ::= <coordinates-keyword> <x-id> , <y-id>

<notif-message-stmt> ::= <notif-keyword> <notif-name>

<purpose-stmt> ::= <purpose-keyword> <text-line>

<revision-date-stmt> ::= <revision-keyword> [ <date-keyword> ] [ <date> ]

<scope-stmt> ::= <scope-keyword> <text-line>

<timing-stmt> ::= <intermediate-keyword> | <optional-keyword>

<viewpoint-stmt> ::= <viewpoint-keyword> <text-line>
```

2.11 PML Names, Numbers, and IDs

```
<activity-name> ::= <quoted-name>

<activity-no> ::= <integer> | ( <activity-no> . <integer> )

<diagram-name> ::= <quoted-name>

<diagram-no> ::= <integer> | ( <diagram-no> . <integer> )

<icom-id> ::= <state-name> * <icom-name>

<icom-name> ::= <long-name>
```

```
<state-name> ::= <long-name>

<junction-name> ::= <exclusive-or>
| <asynchronous-and>
| <asynchronous-or>
| <synchronous-and>
| <synchronous-or>

<exclusive-or> ::= XO

<asynchronous-and> ::= A&

<asynchronous-or> ::= AO

<synchronous-and> ::= S&

<synchronous-or> ::= SO

<junction-no> ::= <integer> { . <integer> } *

<model-name> ::= <quoted-name>

<notif-name> ::= <long-name>

<notiflist-no> ::= <integer> { . <integer> } *

<receiver-name> ::= <long-name>

<x-id> ::= <integer>

<y-id> ::= <integer>
```

2.12 PML Primitives

```
<date> ::= <month> - <day> - <year> [ <hour> : <minute> ]
```

```
<month> ::= <integer> where 01 ≤ month ≤ 12
```

```
<day> ::= <integer> where 01 ≤ day ≤ 31
```

```
<year> ::= <integer> where 0000 ≤ day ≤ 9999
```

```
<hour> ::= <integer> where 00 ≤ hour ≤ 23
```

```
<minute> ::= <integer> where 00 ≤ minute ≤ 59
```

```
<integer> ::= { <digit> } +
```

```
<digit> ::= <zero-digit> | <nonzero-digit>
```

```
<long-name> ::= <letter> { <long-name-character> } *
```

NOTE: `long-name` must not exceed 65 characters and must not be the same as any of the keywords.

```
<long-name-character> ::= <letter>
| <digit>
| <ampersand-sign>
| <at-sign>
| <backward-slash>
| <comma>
| <dollar-sign>
| <forward-slash>
| <hyphen>
| <period>
| <underscore>
```

```
<letter> ::= <upper-case-letter> | <lower-case-letter>
```

```
<quoted-name> ::= " <text-wo-double-quote> ( " | <end-of-line> )
```

```
<text-wo-double-quote> ::= <any-char-but-dbl-quote>
{ <text-wo-double-quote> } *
```

NOTE: `text-wo-double-quote` must not exceed 65 characters.

```
<text-line> ::= <text-value> { <end-of-line> - <text-value> } *
<end-of-line>
```

```
<text-value> ::= <any-char> { <text-value> } *
```

2.13 PML Literals

NOTE: Some of the definitions in this area are italicized. They describe values that were inconvenient to list. Some of them include names of other language components, which are shown in plain text. The unitalicized definitions consist of actual values rather than names of other language components.

`<any-char> ::=` *`any ASCII or EBCDIC value, except`* `end-of-line`

`<any-char-but-dbl-quote> ::=` *`any ASCII or EBCDIC value, except`* `end-of-line` *`or`*
`double-quote`

```
<ampersand-sign> ::= &
```

```
<asterisk> ::= *
```

```
<at-sign> ::= @
```

```
<backward-slash> ::= \
<colon> ::= :

<comma> ::= ,

<dollar-sign> ::= $

<double-quote> ::= "

<end-of-line> ::= any ASCII or EBCDIC value that denotes the end
of a line, e.g., carriage return (CR) or line
feed (LF)

<forward-slash> ::= /

<hyphen> ::= -

<lower-case-letter> ::= a | b | c | d | e | f | g | h | i | j | k | l
| m | n | o | p | q | r | s | t | u | v | w
| x | y | z

<nonzero-digit> ::= 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9

<number-sign> ::= #

<single-quote> ::= '

<underscore> ::= _

<upper-case-letter> ::= a | b | c | d | e | F | g | h | i | j | k | l
| m | n | o | p | q | r | s | t | u | v | w
| x | y | z

<zero-digit> ::= 0
```

_

2.14 PML Keywords

NOTE: These definitions consist of actual values (not case sensitive), and are not the names of other language components.

```
<activity-keyword> ::= ACTIVITY

<author-keyword> ::= AUTHOR

<control-keyword> ::= CONTROL

<coordinates-keyword> ::= COORDINATES

<creation-keyword> ::= CREATION

<date-keyword> ::= DATE

<description-keyword> ::= DESCRIPTION

<diagram-keyword> ::= DIAGRAM

<end-keyword> ::= END

<general-keyword> ::= GENERAL

<icom-keyword> ::= ICOM

<information-keyword> ::= INFORMATION

<input-keyword> ::= INPUT
```

```
<intermediate-keyword> ::= INTERMEDIATE

<junction-keyword> ::= JUNCTION

<mech-keyword> ::= MECHANISM

<model-keyword> ::= MODEL

<notif-keyword> ::= NOTIF

<notiflist-keyword> ::= NOTIFLIST

<optional-keyword> ::= OPTIONAL

<output-keyword> ::= OUTPUT

<purpose-keyword> ::= PURPOSE

<receiver-keyword> ::= RECEIVER

<revision-keyword> ::= REVISION

<scope-keyword> ::= SCOPE

<viewpoint-keyword> ::= VIEWPOINT
```

_

3.0 Collected BNF Grammar of PML

```
<activity-block> ::= <activity-stmt> [ <activity-location-stmt> ]
[ <activity-body> ] [ <end-keyword> ]
<activity-body> ::= ( <activity-icom-block> | <destination-block> )
{ <activity-body> } *
```

```
<activity-control-block> ::= <control-keyword> { <activity-icom> } +

[ <end-keyword> ]

<activity-def> ::= <glossary-activity-stmt> [ <description-stmt> ]

<activity-icom> ::= <icom-id> [ <timing-stmt> ]

<activity-icom-block> ::= <activity-input-block>

| <activity-control-block>

| <activity-output-block>

| <activity-mech-block>

| <activity-notif-block>

<activity-input-block> ::= <input-keyword> { <activity-icom> } +

[ <end-keyword> ]

<activity-keyword> ::= ACTIVITY

<activity-location-stmt> ::= <coordinates-keyword> <x-id> , <y-id>

<activity-mech-block> ::= <mech-keyword> { <activity-icom> } +

[ <end-keyword> ]

<activity-name> ::= <quoted-name>

<activity-no> ::= <integer> | ( <activity-no> . <integer> )

<activity-notif> ::= <activity-notif-stmt> [ <timing-stmt> ]

{ <activity-notif> } *

<activity-notif-block> ::= <notif-keyword> <activity-notif>

[ <end-keyword> ]

<activity-notif-stmt> ::= <notif-name>

<activity-output-block> ::= <output-keyword>

{ <activity-icom> [ <destination-block> ] } +

[ <end-keyword> ]

<activity-stmt> ::= <activity-keyword> <activity-no> <activity-name>

<ampersand-sign> ::= &

<any-char> ::= any ASCII or EBCDIC value, except  end-of-line

<any-char-but-dbl-quote> ::= any ASCII or EBCDIC value, except  end-of-line or
double-quote

<asterisk> ::= *

<asynchronous-and> ::= A&

<asynchronous-or> ::= AO

<at-sign> ::= @
```

<author-keyword> ::= AUTHOR

<author-stmt> ::= <author-keyword> <text-line>

<backward-slash> ::= \

<colon> ::= :

<comma> ::= ,

<control-keyword> ::= CONTROL

<coordinates-keyword> ::= COORDINATES

<creation-date-stmt> ::= <creation-keyword> [<date-keyword>] [<date>]

<creation-keyword> ::= CREATION

<date> ::= <month> - <day> - <year> [<hour> : <minute>]

<date-keyword> ::= DATE

<day> ::= <integer> where 01 ≤ day ≤ 31

<destination-block> ::= <destination-keyword> (<activity-stmt>

| <junction-stmt>)

<description-keyword> ::= DESCRIPTION

<description-stmt> ::= <description-keyword> <text-line>

<diagram-block> ::= <diagram-stmt> <diagram-body> [<end-keyword>]

<diagram-body> ::= <function-block> { <diagram-body> } *

<diagram-keyword> ::= DIAGRAM

<diagram-name> ::= <quoted-name>

<diagram-no> ::= <integer> | (<diagram-no> . <integer>)

<diagram-stmt> ::= <diagram-keyword> <diagram-no> <diagram-name>

<digit> ::= <zero-digit> | <nonzero-digit>

<dollar-sign> ::= $

<double-quote> ::= "

<element-def> ::= <activity-def> [<icom-def>]

<end-keyword> ::= END

<end-of-line> ::= *any ASCII or EBCDIC value that denotes the end*

of a line, e.g., carriage return (CR) or line

feed (LF)

<exclusive-or> ::= XO

<forward-slash> ::= /

<function-block> ::= <activity-block>

```
| <junction-block>
| <notif-block>
<general-keyword> ::= GENERAL
<gen-info-stmt> ::= <general-keyword> [ <information-keyword> ]
<glossary> ::= { <element-def> } +
<glossary-activity-stmt> ::= <activity-keyword> <activity-no> <activity-name>
<glossary-icom-stmt> ::= <icom-keyword> <icom-id>
<hour> ::= <integer> where 00 ≤ hour ≤ 23
<hyphen> ::= -
<icom-def> ::= <icom-def-body> { <icom-def> } +
<icom-def-body> ::= <glossary-icom-stmt> <icom-type-stmt>
[ <description-stmt> ]
<icom-id> ::= <state-name> * <icom-name>
<icom-keyword> ::= ICOM
<icom-name> ::= <long-name>
<icom-type-stmt> ::= <input-keyword>
| <output-keyword>
| <control-keyword>
| <mech-keyword>
<information-keyword> ::= INFORMATION
<input-keyword> ::= INPUT
<integer> ::= { <digit> } +
<intermediate-keyword> ::= INTERMEDIATE
<junction-block> ::= <junction-stmt> [ <junction-location-stmt> ]
[ <junction-body> ] [ <end-keyword> ]
<junction-body> ::= ( <junction-icom-block> | <destination-block> )
{ <junction-body> } *
<junction-icom-block> ::= <junction-input-block>
| <junction-output-block>
| <function-notif-block>
<junction-input-block> ::= <input-keyword> { <icom-id> } +
[ <end-keyword> ]
<junction-keyword> ::= JUNCTION
<junction-location-stmt> ::= <coordinates-keyword> <x-id> , <y-id>
```

```
<junction-name> ::= <exclusive-or>
| <asynchronous-and>
| <asynchronous-or>
| <synchronous-and>
| <synchronous-or>
<junction-no> ::= <integer> { . <integer> } *
<junction-notif> ::= <junction-notif-stmt> { <junction-notif> } *
<junction-notif-block> ::= <notif-keyword> <junction-notif>
[ <end-keyword> ]
<junction-notif-stmt> ::= <notif-name>
<junction-output-block> ::= <output-keyword>
{ <icom-id> [ <destination-block> ] } +
[ <end-keyword> ]
<junction-stmt> ::= <junction-keyword> <junction-no> <junction-name>
<letter> ::= <upper-case-letter> | <lower-case-letter>
<long-name> ::= <letter> { <long-name-character> } *
<long-name-character> ::= <letter>
| <digit>
| <ampersand-sign>
| <at-sign>
| <backward-slash>
| <comma>
| <dollar-sign>
| <forward-slash>
| <hyphen>
| <period>
| <underscore>
<lower-case-letter> ::= a | b | c | d | e | f | g | h | i | j | k | l
| m | n | o | p | q | r | s | t | u | v | w
| x | y | z
<mech-keyword> ::= MECHANISM
<minute> ::= <integer> where 00 ≤ minute ≤ 59
<model> ::= <model-stmt> [ <model-body> ]
```

```
<model-body> ::= <gen-info-stmt> <model-gen-info>
{ <diagram-block> } +
<model-gen-info> ::= <creation-date-stmt> [ <revision-date-stmt> ]
[ <author-stmt> ] [ <description-stmt> ]
[ <scope-stmt> ] [ <purpose-stmt> ]
[ <viewpoint-stmt> ]
<model-keyword> ::= MODEL
<model-name> ::= <quoted-name>
<model-stmt> ::= <model-keyword> <model-name>
<month> ::= <integer> where 01 ≤ month ≤ 12
<nonzero-digit> ::= 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9
<notif-block> ::= <notiflist-stmt> [ <notif-location-stmt> ]
[ <notif-body> ] [ <end-keyword> ]
<notif-body> ::= <notif-message-block> <notif-receiver-block>
<notif-keyword> ::= NOTIF
<notif-location-stmt> ::= <coordinates-keyword> <x-id> , <y-id>
<notif-message-stmt> ::= <notif-keyword> <notif-name>
<notif-message-block> ::= <notif-message-stmt> [ <end-keyword> ]
<notif-name> ::= <long-name>
<notif-receiver> ::= { <receiver-name> } +
<notif-receiver-block> ::= <receiver-keyword> <notif-receiver>
[ <end-keyword> ]
<notiflist-keyword> ::= NOTIFLIST
<notiflist-no> ::= <integer> { . <integer> } *
<notiflist-stmt> ::= <notiflist-keyword> <notiflist-no>
<number-sign> ::= #
<optional-keyword> ::= OPTIONAL
<output-keyword> ::= OUTPUT
<pml-file> ::= <model> [ <glossary> ] [ <end-keyword> ]
<purpose-keyword> ::= PURPOSE
<purpose-stmt> ::= <purpose-keyword> <text-line>
<quoted-name> ::= " <text-wo-double-quote> ( " | <end-of-line> )
<receiver-keyword> ::= RECEIVER
<receiver-name> ::= <long-name>
```

```
<revision-date-stmt> ::= <revision-keyword> [ <date-keyword> ] [ <date> ]
<revision-keyword> ::= REVISION
<scope-keyword> ::= SCOPE
<scope-stmt> ::= <scope-keyword> <text-line>
<single-quote> ::= '
<state-name> ::= <long-name>
<synchronous-and> ::= S&
<synchronous-or> ::= SO
<text-line> ::= <text-value> { <end-of-line> - <text-value> } *
<end-of-line>
<text-value> ::= <any-char> { <text-value> } *
<text-wo-double-quote> ::= <any-char-but-dbl-quote>
{ <text-wo-double-quote> } *
<timing-stmt> ::= <intermediate-keyword> | <optional-keyword>
<underscore> ::= _
<upper-case-letter> ::= a | b | c | d | e | F | g | h | i | j | k | l
| m | n | o | p | q | r | s | t | u | v | w
| x | y | z
<viewpoint-keyword> ::= VIEWPOINT
<viewpoint-stmt> ::= <viewpoint-keyword> <text-line>
<x-id> ::= <integer>
<y-id> ::= <integer>
<year> ::= <integer> where 0000 ≤ day ≤ 9999
<zero-digit> ::= 0
```

4.0 General Notes

Model PML

1. The `Activity Sections` in a diagram may be in any sequence.

2. The `Junction Sections` in a diagram may be in any sequence.

3. The `Notification Sections` in a diagram may be in any sequence.

4. The `Input`, `Output`, `Control`, `Mechanism`, and `Notification` portions of an `Activity Section` in a diagram may be in any sequence and each portion should appear no more than once per `activity`.

5. The `Input`, `Output`, and `Notification` portions of a `Junction Section` in a diagram may be in any sequence and each portion should appear no more than once per `junction`.

6. The `Notification` and `Receiver` portions of a `Notification Section` in any diagram must be in the sequence `notif-message` then `receiver`.

7. `text-line` may be any number of characters. A `text-line` may contain upper-case or lower-case letters, numbers, spaces, and any punctuation. It begins with a first non-blank character and ends with the last non-blank character on the line, and it may be continued on subsequent lines. Each continuation line must have a hyphen (-) as its first non blank character. The continuation text begins with the first character (either blank or non blank) following a hyphen and ends with the last non blank character of the line. There is no limit on the number of continuation lines that may be used.

8. When the `month` is "4", "6", "8", or "11", the `day` shall be less than or equal to "30"; when the `month` is "2", and the `year` is a leap year, the `day` shall be less than or equal to "29"; when the `month` is "2", and the `year` is not a leap year, the `day` shall be less than or equal to "28"; otherwise, the `day` shall be less than or equal to "31".

5.0 Outstanding Issues

1. Decision Points

Currently, an exclusive-or junction (XO) is utilized as a decision point. An enhancement to the PML should specify how a decision is made within an exclusive-or junction.

_

6.0 PML Examples

This section contains an example of a model and glossary PML file for a given workflow. The following conventions are used in both to improve the readability of the examples:

Keywords are in upper case.

Names are a mixture of upper and lower case (initial caps).

Blank lines are used to separate major blocks of PML statements.

Indention is used to show nesting of minor blocks of PML statements.

The italized PML statements on the right side of the page are not part of the PML text

file. These PML statement are provided for traceability to the BNF definition of PML.

All of these conventions conform to the PML syntax, but none of them are required.

_

6.1 Model PML Example

```
MODEL "Commercial Loan Process" model-stmt
```

```
GENERAL INFORMATION gen-info-stmt
AUTHOR Dave Grubel author-stmt
CREATION DATE 5-15-1993 creation-date-stmt
REVISION DATE 9-1-1993 revision-date-stmt
DESCRIPTION This model describes how loans description-stmt
- are currently handled at the text-line
- Commercial National Bank. text-line
PURPOSE This model will be used to help purpose-stmt
- identify improvements that can be text-line
- made to correlate loan accounts text-line
- with checking and savings accounts. text-line
SCOPE Existing commercial loan generation scope-stmt
- and administration procedures. text-line
VIEWPOINT Branch Manager viewpoint-stmt

DIAGRAM 0 "Conduct Commercial Loan Process" diagram-stmt

ACTIVITY 1 "Conduct Credit Check" activity-stmt
INPUT input-keyword
New*Loan-Evaluation-Worksheet icom-id
Unapproved*Loan-Application icom-id
END end-keyword
CONTROL control-keyword
Approved*Application-Procedures icom-id
END end-keyword
MECHANISM mechanism-keyword
Qualified*Loan-Officer icom-id
END end-keyword
OUTPUT output-keyword
In-Work*Loan-Application icom-id
Completed*Loan-Evaluation-Workshet icom-id
END end-keyword
END end-keyword
```

ACTIVITY 2 "Make Loan Determination" _activity-stmt_

INPUT _input-keyword_

In-Work*Loan-Application _icom-id_

END _end-keyword_

CONTROL _control-keyword_

Completed*Loan-Evaluation-Worksheet _icom-id_

Approved*Qualification-Procedures _icom-id_

END _end-keyword_

MECHANISM _mechanism-keyword_

Qualified*Loan-Officer _icom-id_

Qualified*Branch-Manager _icom-id_

END _end-keyword_

OUTPUT _output-keyword_

Reviewed*Loan-Application _icom-id_

Generated*Internal-Order Optional _icom-id_

timing-stmt

END _end-keyword_

NOTIF _notif-keyword_

rejection-notification _activity-notif-stmt_

END _end-keyword_

END _end-keyword_

ACTIVITY 3 "Book Loan" _activity-stmt_

INPUT _input-keyword_

Generated*Internal-Order _icom-id_

END _end-keyword_

CONTROL _control-keyword_

Approved*Loan-Application _icom-id_

Approved*Loan-Booking-Procedures _icom-id_

END _end-keyword_

MECHANISM _mechanism-keyword_

Trained*Note-Department _icom-id_

END _end-keyword_

OUTPUT *output-keyword*

Approved*Loan-File *icom-id*

Approved*Loan-Dollars *icom-id*

END *end-keyword*

END *end-keyword*

JUNCTION 4 AO *junction-stmt*

INPUT *input-keyword*

Received*Loan-Application *icom-id*

Rejected*Loan-Application *icom-id*

END *end-keyword*

OUTPUT *output-keyword*

Unapproved*Loan-Application *icom-id*

END *end-keyword*

END *end-keyword*

JUNCTION 6 XO *junction-stmt*

INPUT *input-keyword*

Reviewed*Loan-Application *icom-id*

END *end-keyword*

OUTPUT *output-keyword*

Rejected*Loan-Application *icom-id*

Approved*Loan-Application *icom-id*

END *end-keyword*

END *end-keyword*

JUNCTION 7 A& *junction-stmt*

INPUT *input-keyword*

Approved*Loan-Generation-Procedures *icom-id*

END *end-keyword*

OUTPUT *output-keyword*

Approved*Application-Procedures *icom-id*

Approved*Loan-Generation-Procedures *icom-id*

END *end-keyword*

END *end-keyword*

JUNCTION 8 A& *junction-stmt*

INPUT *input-keyword*

Approved*Loan-Generation-Procedures *icom-id*

END *end-keyword*

OUTPUT *output-keyword*

Approved*Qualification-Procedures *icom-id*

Approved*Loan-Booking-Procedures *icom-id*

END *end-keyword*

END *end-keyword*

JUNCTION 9 A& *junction-stmt*

INPUT *input-keyword*

Qualified*Loan-Officer *icom-id*

END *end-keyword*

OUTPUT *output-keyword*

Qualified*Loan-Officer *icom-id*

Qualified*Loan-Officer *icom-id*

END *end-keyword*

END *end-keyword*

NOTIFLIST 10 *notiflist-stmt*

NOTIF *notif-keyword*

Rejection-Notification *notif-name*

END *end-keyword*

RECEIVERS *receiver-keyword*

Customer *receiver-name*

Loan-Officer *receiver-name*

END *end-keyword*

END *end-keyword*

END *end-keyword*

```
ACTIVITY 1 "Conduct Credit Check" glossary-activity-stmt
DESCRIPTION When the completed application has been accepted by the
- bank, it is placed in an application file. This file is
- identified by the customer's name and contains the
- information gathered from inside the bank and from one or
- more credit reporting agencies. This information includes:

- - Information on the applicant's checking and saving
- accounts,
-
- - Information about current and past loans. Credit agency
- information may come from TRW or Dun & Bradstreet. These
- reports include information about accounts with other
- banks and businesses.

ICOM Received*Loan-Application INPUT glossary-icom-stmt icom-type-stmt
DESCRIPTION Formal request to the bank to establish commercial loan
- accounts.

ACTIVITY 2 "Make Loan Determination" glossary-activity-stmt
DESCRIPTION Once the supporting information has been assembled in the
- application file, it is reviewed by a loan officer. The
- loan officer determines the appropriate Ts & Cs for the
- application, and decides whether to approve the loan. The
- loan may be approved by the loan officer or the branch
- manager, depending on the amount requested. The loan may be
- declined by the loan officer or the manager. At any time,
- more information may be requested.

ICOM Qualified*Loan-Officer MECHANISM glossary-icom-stmt icom-type-stmt
DESCRIPTION A specific bank employee responsible for generating and
- managing loans.
```

ACTIVITY 3 "Book Loan" *glossary-activity-stmt*

DESCRIPTION The actual booking of a commercial loan and the distibution

- of laon dollars.

ICOM Approved*Loan-Dollars OUTPUT *glossary-icom-stmt icom-type-stmt*

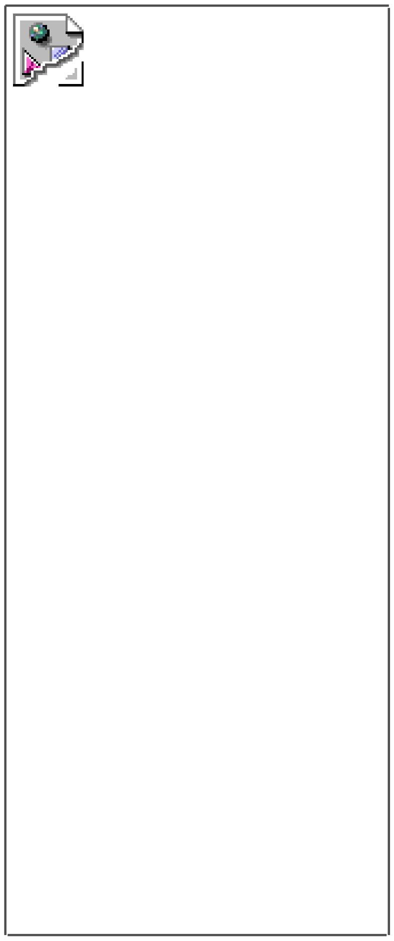
DESCRIPTION The result of an approved application for a specific amount

- of money loaned to a customer under a single set of terms

- and conditions.

END *end-keyword*

6.2 Process Model Diagram Example



APPENDIX A - Acronym and Abbreviations

A& - Asyncronous And

AIE - Advanced Information Engineering

AO - Asyncronous Or

ASCII - American Standard Code for Information Interchange

BNF - Backus Naur Form

CR - Carriage Return

EBCDIC - Extended Binary Coded Decimal Interchange Code

ICAM - Integrated Computer Aided Manufacturing

ICOM - Input, Control, Output, Mechanism

ID - Identification

IDEF-3x - ICAM DEFinition language 3 extended

LF - Line Feed

NAA - North American Aircraft

No - Number

PFM - Process Flow Model

PML - Process Modeling Language

PO - Purchase Order

RASSP - Rapid Prototyping of Application Specific Signal Processors

RI - Rockwell International

S& - Syncronous And

SO - Syncronous Or

WF - Workflow

XO - Exclusive Or