Appendices

Appendix A. Axioms and Classification Rules for the Specification Meta-Model

This appendix specifies key portions of the specification meta-model described in Chapter 4. Two specifications are included. Section A.1 specifies axioms for the concepts used to represent data/control flow diagrams and specification addenda, as discussed in Chapter 4, sections 4.1 and 4.2. Section A.2 specifies rules for classifying concepts from a data/control flow diagram, as described in Chapter 4, section 4.3.

A.1 Axioms for Semantic Concepts

Chapter 4 describes a specification meta-model for representing data/control flow diagrams and certain specification addenda. The specification meta-model consists, in part, of concepts, with each concept representing some semantic element in the meta-model. Each concept can be constrained by a set of axioms. This appendix specifies the relevant axioms, organized into two groups: axioms for specification elements and axioms for specification addenda. Within each group the concepts are listed in alphabetical order. Any concept not listed should be assumed to require no unique axioms, that is, no axioms beyond any inherited from a parent concept.

A.1.1 Axioms for Specification Elements

Concept: Aperiodic Function

Axiom: **No Timer**

An Aperiodic Function cannot be the sink for a Timer.

Concept: Asynchronous Device Interface Object

Axiom: **One, and Only One, Interrupt**

An Asynchronous Device Interface Object must be the sink for an Interrupt, and cannot be the sink for more than one Interrupt.

Axiom: **No Timer**

An Asynchronous Device Interface Object cannot be the sink for a Timer.

Concept: Control Event Flow

Axiom: **Restricted Name**

A Control Event Flow must be named one of Disable, Enable, or Trigger.

Axiom: **Restricted Source**

The source of a Control Event Flow must be a Control Object.

Axiom: **Restricted Sink**

The sink of a Control Event Flow must be a State-Dependent Function.

Concept: Control Object

Axiom: **Inputs Must Be Signals**

Any Directed Arc with a sink that is a Control Object must also be a Signal.

Axiom: **Uniquely-Named Inputs**

For a given Control Object, let the set S be all Signals whose sink is the Control Object. No two elements in S may have the same label.

Axiom: **Restricted Output Types**

For any Control Object, let the set D be all Directed Arcs whose source is the Control Object. Each element of D must be either a Control Event Flow or a Signal.

Concept: Dashed Transform

Axiom: **Dashed Input Arcs Only**

For any Dashed Transform, let the set I be all Directed Arcs whose sink is the Dashed Transform. Each element of I must be either a Dashed Directed Arc or the descendant of a Dashed Directed Arc.

Axiom: **Dashed Output Arcs Only**

For any Dashed Transform, let the set O be all Directed Arcs whose source is the Dashed Transform. Each element of O must be either a Dashed Directed Arc or the descendant of a Dashed Directed Arc.

Axiom: **No Two-Way Arcs**

There cannot exist a Two-Way Arc that connects (via left or right) to a Dashed Transform.

Concept: Data Store

Axiom: **At Least One Edge With A Transform**

For a given Data Store, at least one of the following must hold: 1) there is a Data-Store Data Flow whose source is the Data Store and whose sink is a Solid Transform, 2) there is a Data-Store Data Flow whose sink is the Data Store and whose source is a Solid Transform, 3) there is a Solid Two-Way Arc whose left is the Data Store and whose right is a Solid Transform, or 4) there is a Solid Two-Way Arc whose right is the Data Store and whose left is a Solid Transform.

Axiom: **Limited Directed Arc Types**

Any Directed Arc whose source is the Data Store must be a Retrieve, and any Directed Arc whose sink is the Data Store must be a Store.

Axiom: **Limited Two-Way Arc Types**

Any Solid Two-Way Arc whose right is the Data Store must be an Update.

Axiom: **Connects Only To Solid Transforms**

There cannot exist any of the following: 1) a Retrieve whose source is the Data Store and whose sink is not a Solid Transform, 2) a Store whose sink is the Data Store and whose source is not a Solid Transform, or 3) an Update whose right is the Data Store and whose left is not a Solid Transform.

Axiom: **At Most One Link Per Transform**

For any pair of Data Store and Solid Transform, let S be the set of Solid Directed Arcs such that either: 1) the Data Store is the sink and the Solid Transform is the source or 2) the Data Store is source and the Solid Transform is the sink. For the same pair of Data Store and Solid Transform, let U be the set of Solid Two-Way Arcs such that either: 1) the Data Store is the right and the Solid Transform is the left or 2) the Data Store is the left and the Solid Transform is the right. The cardinality of S added to the cardinality of U must not exceed one.

Concept: Data-Store Data Flow

Axiom: **Connects Solid Transform With Data Flow**

There must exist a Data Store and a Solid Transform such that either: 1) the source of the Data-Store Data Flow is the Data Store and the sink is the Solid Transform, or 2) the sink of the Data-Store Data Flow is the Data Store and the source is the Solid Transform.

Concept: Device

Axiom: **Outgoing Solid Arc Restrictions**

Any Solid Directed Arc whose source is a Device must: 1) be an Input and 2) have a sink that is a Device Interface Object.

Axiom: **Incoming Solid Arc Restrictions**

Any Solid Directed Arc whose sink is a Device must: 1) be an Output and 2) have a source that is a Device Interface Object.

Axiom: **Outgoing Dashed Arc Restrictions**

Any Dashed Directed Arc whose source is a Device must: 1) be an Interrupt and 2) have a sink that is a Device Interface Object.

Concept: Device Input Object

Axiom: **Input Only**

There cannot exist an Output whose source is a Device Input Object.

Concept: Device Interface Object

Axiom: **Requires Input, Output, Or Interrupt**

For any Device Interface Object, there must exist at least one of the following: 1) an Input whose sink is the Device Interface Object, 2) an Output whose source is the Device Interface Object, or 3) an Interrupt whose sink is the Device Interface Object.

Concept: Device IO Object

Axiom: **Requires Input And Output**

For any Device IO Object, there must exist both of the following: 1) an Input whose sink is the Device IO Object and 2) an Output whose source is the Device IO Object.

Concept: Device Output Object

Axiom: **Output Only**

For any Device Output Object, there cannot exist an Input such that the sink of the Input is the Device Output Object.

Axiom: **Cannot Send Stimulus Or Signal**

For any Device Output Object, there cannot exist: 1) a Stimulus whose source is the Device Output Object or 2) a Signal whose source is the Device Output Object.

Concept: Directed Arc

Axiom: **Distinct Source And Sink**

A Directed Arc cannot have the same source and sink.

Axiom: **Source Is A Node Name**

A Directed Arc must have a source that identifies a Node or that identifies the System.

Axiom: **Sink Is A Node Name**

A Directed Arc must have a sink that identifies a Node.

Concept: Disable

Axiom: **Restricted Name**

A Disable must have a label that is Disable.

Axiom: **Requires Corresponding Enable**

There must exist an Enable with the same source and sink as the Disable.

Concept: Enable

Axiom: **Restricted Name**

An Enable must have a label that is Enable.

Axiom: **Requires Corresponding Disable**

There must exist a Disable with the same source and sink as the Enable.

Axiom: **Restricted Sink**

There must exist an Enabled Function that is the sink of the Enable.

Concept: Enabled Asynchronous Function

Axiom: **No Timer**

An Enabled Asynchronous Function cannot be the sink for a Timer.

Concept: Enabled Function

Axiom: **Requires Incoming Enable**

There must exist an Enable whose sink is the Enabled Function.

Concept: Enabled Periodic Function

Axiom: **One, And Only One, Timer**

Let T be the set of all Timers such that the sink of the Timer is the Enabled Periodic Function. The cardinality of T must be one.

Concept: Event Flow

Axiom: **Requires A Label**

An Event Flow must have a label that is not nil.

Axiom: **No Exchanges With A Data Store**

An Event Flow cannot have a source or a sink that is a Data Store.

Concept: External Data Flow

Axiom: **Requires A Label**

An External Data Flow must have a label that is not nil.

Axiom: **No Exchanges With A Data Store**

An External Data Flow cannot have a source or a sink that is a Data Store.

Axiom: **Connects Interface Object With Terminator**

There must exist an Interface Object and a Terminator such that either: 1) the source of the External Data Flow is an Interface Object and the sink is a Terminator or 2) the sink of the External Data Flow is an Interface Object and the source is a Terminator.

Axiom: **Distinct Names**

There cannot be two, distinct External Data Flows with the same label.

Axiom: **Names Distinct From Interrupt Names**

There cannot be an Interrupt with the same label as the External Data Flow.

Concept: External Subsystem

Axiom: **Outgoing Solid Arc Restrictions**

Any Solid Directed Arc whose source is an External Subsystem must be an Input and must have a sink that is a Subsystem Interface Object.

Axiom: **Incoming Solid Arc Restrictions**

Any Solid Directed Arc whose sink is an External Subsystem must be an Output and must have a source that is a Subsystem Interface Object.

Axiom: **No Outgoing Dashed Arc**

There cannot exist a Dashed Directed Arc whose source is an External Subsystem.

Concept: Function

Axiom: **No Data Exchange With Terminator**

A Function cannot be the source or sink of an External Data Flow.

Axiom: **No Interrupt From Terminator**

A Function cannot be the sink of an Interrupt.

Axiom: **No Control Event Flows Out**

A Function cannot be the source of a Control Event Flow.

Axiom: **Requires Activator**

A Function must be the sink for at least one of the following: 1) a Control Event Flow, 2) a Signal, 3) a Timer, or 4) a Stimulus.

Concept: Input

Axiom: **From Terminator To Interface Object**

The must exist a Terminator and Interface Object such that the source of an Input is the Terminator and the sink of that Input is the Interface Object.

Concept: Interface Object

Axiom: **Interface To One, And Only One, Terminator**

Let T be the set of Terminators such that there exists a Directed Arc where either: 1) the source of the Directed Arc is the Interface Object and the sink is a Terminator or 2) the sink of the Directed Arc is the Interface Object and the source is a Terminator. The cardinality of T must be one.

Concept: Internal Data Flow

Axiom: **Requires A Label**

The label of an Internal Data Flow cannot equal nil.

Axiom: **No Exchanges With Data Store**

Neither the source nor the sink of an Internal Data Flow can be a Data Store.

Axiom: **Connects Two Solid Transforms**

There must exist two, distinct Solid Transforms such that the sink of an Internal Data Flow is one of the Solid Transforms and the source of that Internal Data Flow is the other Solid Transform.

Concept: Interrupt

Axiom: **Distinct Name**

There cannot exist two, distinct Interrupts with the same label.

Axiom: **Name Distinct From External Data Flow**

There cannot exist an External Data Flow with the same label as the Interrupt.

Axiom: **From Device To Asynchronous Interface Object**

The source of the Interrupt must be a Device and the sink must be an Asynchronous Interface Object.

Concept: Node

Axiom: **Name Required**

The label of a Node cannot be nil.

Axiom: **Name "System" Reserved**

The label of a Node cannot be System.

Axiom: **Distinct Name**

There cannot exist two, distinct Nodes with the same label.

Concept: Normally-Named Event Flow

Axiom: **Reserved Names Excluded**

The label of a Normally-Named Event Flow cannot be one of the reserved event names, that is, cannot be any of Disable, Enable, Timer, or Trigger.

Concept: Output

Axiom: **From Interface Object To Terminator**

The must exist a Terminator and Interface Object such that the sink of an Output is the Terminator and the source of that Output is the Interface Object.

Concept: Passive Device Input Object

Axiom: **Must Emit A Response**

The must exist a Response for which the Passive Device Input Object is the source.

Concept: Passive Device Interface Object

Axiom: **Must Receive Stimulus Or Signal**

There must exist a Stimulus or a Signal whose sink is the Passive Device Interface Object.

Axiom: **No Timer**

There must not exist a Timer whose sink is the Passive Device Interface Object.

Axiom: **No Interrupt**

There must not exist an Interrupt whose sink is the Passive Device Interface Object.

Concept: Passive Device IO Object

Axiom: **Must Emit A Response**

The must exist a Response for which the Passive Device IO Object is the source.

Concept: Periodic Device Interface Object

Axiom: **One, And Only One, Timer**

Let T be the set of all Timers whose sink is the Periodic Device Interface Object. The cardinality of T must be one.

Axiom: **No Interrupt**

There must not exist an Interrupt whose sink is the Periodic Device Interface Object.

Concept: Periodic Function

Axiom: **One, And Only One, Timer**

Let T be the set of all Timers whose sink is the Periodic Function. The cardinality of T must be one.

Concept: Response

Axiom: **Requires Corresponding Stimulus**

There must exist a Stimulus whose source is the same as the sink of the Response and whose sink is the same as the source of the Response.

Concept: Retrieve

Axiom: **From Data Store To Solid Transform**

The source of a Retrieve must be a Data Store and the sink of a Retrieve must be a Solid Transform.

Concept: Signal

Axiom: **Flows Between Transforms**

Both the source and sink of a Signal must be Transforms.

Concept: Solid Directed Arc

Axiom: **No Exchanges With Control Objects**

Neither the source nor the sink of a Solid Directed Arc can be a Control Object.

Concept: Solid Transform

Axiom: **No Redundant Data Flows**

Two, distinct Internal Data Flows cannot have the same source and sink when both the source and the sink are Solid Transforms.

Concept: Specially-Named Event Flow

Axiom: **Restricted Name**

The label of a Specially-Named Event Flow must be one of the following: Enable, Disable, or Trigger.

Concept: State-Dependent Function

Axiom: **Requires Event Flow From Control Object**

There must exist an Event Flow whose sink is the State-Dependent Function and whose source is a Control Object.

Axiom: **Trigger Restricts Input Event Flows**

If a State-Dependent Function is the sink for a Trigger, then that State-Dependent Function cannot be the sink for another Event Flow, unless that other Event Flow is a Timer.

Axiom: **Enable Pairs Only With Disable And Timer**

If a State-Dependent Function is the sink for an Enable, then that State-Dependent Function cannot be the sink for another Event Flow, unless that other Event Flow is a Timer or a Disable.

Axiom: **Disable Pairs Only With Enable And Timer**

If a State-Dependent Function is the sink for a Disable, then that State-Dependent Function cannot be the sink for another Event Flow, unless that other Event Flow is a Timer or an Enable.

Axiom: **Incoming Signal XOR With Incoming Control Event Flow**

If a State-Dependent Function is the sink for a Signal, then that State-Dependent Function cannot be the sink for a Control Event Flow.

Concept: State-Independent Function

Axiom: **No Events From A Control Object**

There cannot exist an Event Flow whose sink is a State-Independent Function and whose source is a Control Object.

Concept: Store

Axiom: **From Solid Transform To Data Store**

The source of a Store must be a Solid Transform and the sink of that Store must be a Data Store.

Concept: Subsystem Interface Object

Axiom: **No Interrupt**

There cannot exist a Dashed Directed Arc whose source is a Terminator and whose sink is a Subsystem Interface Object.

Axiom: **Requires Input Or Output**

There must exist either: 1) an Input whose source is an External Interface Object and whose sink is the Subsystem Interface Object or 2) an Output whose sink is an External Interface Object and whose source is the Subsystem Interface Object.

Concept: Synchronous Function

Axiom: **Identically-Named Incoming Stimuli**

A Synchronous Function cannot be the sink for two Stimuli such that the labels of the Stimuli are not the same.

Axiom: **Identically-Named Outgoing Responses**

A Synchronous Function cannot be the source for two Responses such that the labels of the Responses are not the same.

Axiom: **Incoming Signal XOR With Incoming Stimulus**

A Synchronous Function cannot be the sink for both a Stimulus and a Signal.

Axiom: **No Return Signal**

A Synchronous Function cannot be the source for a Signal whose sink is the same as a Signal received by the Synchronous Function.

Concept: Terminator

Axiom: **Edge With Interface Object**

There must exist a Directed Arc and an Interface Object such that either: 1) the Terminator is the sink of the Directed Arc and the Interface Object is the source or 2) the Terminator is the source of the Directed Arc and the Interface Object is the sink.

Axiom: **At Most One Input**

A Terminator cannot be the source for more than one Solid Directed Arc..

Axiom: **At Most One Output**

A Terminator cannot be the sink for more than one Solid Directed Arc.

Axiom: **At Most One Interrupt**

A Terminator cannot be the source for more that one Dashed Directed Arc.

Axiom: **No Sink For Dashed Arcs**

A Terminator cannot be the sink for a Dashed Directed Arc.

Axiom: **One Transform Per Terminator**

Both of the following must hold: 1) there cannot exist two, distinct Directed Arcs whose source is the Terminator and whose sinks are not identical and 2) there cannot exist two, distinct Directed Arcs whose sink is the Terminator and whose sources are not identical.

Axiom: **Like Cardinality**

Given a Transform that is connected to the Terminator with a Directed Arc, that is, either the Transform is the source and the Terminator is the sink of the Directed Arc or the Transform is the sink and the Terminator is the source of the Directed Arc, then the Transform and the Terminator must have identical cardinality.

Concept: Timer

Axiom: **Restricted Source**

The source of a Timer must be System.

Axiom: **Restricted Sink**

The sink of a Timer must be a Solid Transform.

Axiom: **Positive Period**

A Timer must have a period that exceeds zero seconds.

Concept: Transform

Axiom: **Distinct Number**

There cannot exist two, distinct Transforms, where the number for each Transform is not nil, that both have identical numbers.

Axiom: **At Least One Input**

A Transform must be the sink of at least one Directed Arc.

Axiom: **At Least One Output**

A Transform must be the source of at least one Directed Arc.

Concept: Trigger

Axiom: **Restricted Name**

The label of a Trigger must be Trigger.

Axiom: **Restricted Sink**

The sink of a Trigger must be a Triggered Function.

Concept: Triggered Asynchronous Function

Axiom: **No Timer**

There cannot exist a Timer whose sink is a Triggered Asynchronous Function.

Concept: Triggered Function

Axiom: **Requires Incoming Signal Or Trigger**

There must exist a Signal or a Trigger whose sink is the Triggered Function.

Concept: Triggered Periodic Function

Axiom: **One, And Only One, Timer**

Let T be the set of Timers whose sink is the Triggered Periodic Function. The cardinality of T must be one.

Concept: Triggered Synchronous Function

Axiom: **No Timer**

There cannot exist a Timer whose sink is the Triggered Synchronous Function.

Concept: Two-Way Arc

Axiom: **Distinct Left And Right**

The left and right of the Two-Way Arc cannot be identical.

Axiom: **Left Is Node Name**

There must exist a Node whose label is equal to the left of the Two-Way Arc.

Axiom: **Right Is Node Name**

There must exist a Node whose label is equal to the right of the Two-Way Arc.

Concept: Update

Axiom: **Connects Data Store With Solid Transform**

For every Update, the following must hold: 1) the left identifies a Solid Transform and 2) the right identifies a Data Store.

Concept: User Role

Axiom: **Outgoing Solid Arc Restrictions**

Any Solid Directed Arc whose source is a User Role must be an Input and must have a sink that is a User-Role Interface Object.

Axiom: **Incoming Solid Arc Restrictions**

Any Solid Directed Arc whose sink is a User Role must be an Output and must have a source that is a User-Role Interface Object.

Axiom: **No Outgoing Dashed Arcs**

There cannot exist a Dashed Directed Arc whose source is a User Role.

Concept: User-Role Interface Object

Axiom: **No Interrupt**

A User-Role Interface Object cannot be the sink of a Dashed Directed Arc whose source is a Terminator.

Axiom: **Requires Input And Output**

Both of the following must hold: 1) there exists an Input whose source is a User Role and whose sink is the User-Role Interface Object and 2) there exists an Output whose sink is a User Role and whose source is the User-Role Interface Object.

A.1.2 Axioms for Specification Addenda

Concept: Aggregation Group

Axiom: **Control Object Exists**

The Control Object associated with the Aggregation Group must exist.

Axiom: **Controls One Group**

Any Control Object in a specification can belong to no more than one Aggregation Group.

Axiom: **Limited To One Group**

Any Device in a specification can belong to no more than one Aggregation Group.

Concept: Exclusion Group

Axiom: **Must Be Periodic Or Asynchronous Function**

If the Exclusion Group is not associated with a Control Object, then each member of the Exclusion Group must be either a Periodic Function or an Asynchronous Function.

Axiom: **Requires Control Object**

If the Exclusion Group is associated with a Control Object, then the Control Object must exist.

Axiom: **Requires Enable**

If the Exclusion Group is associated with a Control Object, then each member of the Exclusion Group must be a State-Dependent Function and must be the sink for an Enable whose source is the Control Object associated with the Exclusion Group.

Axiom: **Limited To One Group**

A Function can be a member of at most one Exclusion Group.

Concept: Locked-State Events

Axiom: **Requires Control Object**

The Control Object associated with the Locked-State Events must exist.

Axiom: **Control Object Sink For All Signals**

The Control Object associated with the Locked-State Events must be the sink for each Signal that is a member of the Locked-State Events.

A.2 Rules for Classifying Semantic Concepts

Chapter 4, section 4.3, describes a concept classifier for inferring the presence of semantic concepts within the specification meta-model. The concept classifier uses four sets of classification rules, each set corresponding to a stage in the classification process. This appendix specifies each set of rules in four sections, as follows.

- Stage One Rules: Arc Classification (A.2.1)
- Stage Two Rules: Transformation Classification (A.2.2)
- Stage Three Rules: Stimulus-Response Classification (A.2.3)
- Stage Four Rules: Ambiguous-Function Classification (A.2.4)

To understand the purpose and form of these rules, as well as the relationship between classification stages and concepts within the specification meta-model, the reader should refer to Chapter 4, section 4.3.

A.2.1 Rules for Arc Classification

Rule: All Terminators Are Devices (First Preference)

```
if
        the concept is a Terminator and
        the classification stage is one and
        this question has not already been asked
then
        ask user if all Terminators in the specification are devices
        if      the user says yes or (the user is inexperienced and says no)
        then
                classify each Terminator as a Device
        fi
fi
```

Rule: User Classifies Terminator

```
if
        the concept is a Terminator
then
        ask user to classify the Terminator, if possible
        if      the user makes a classification
        then
                classify the concept as the user directs
        else
                classify the concept as a Device
        fi
fi
```

Rule: Classify Control Object

```
if
        the concept is a Dashed Transform and
        the Dashed Transform has a name that is not "System"
then
        classify the concept as a Control Object
fi
```

Rule: Classify Interface Object

if
the concept is a Solid Transform and
(the concept is the sink of a Directed Arc that has a
Terminator as its source or
the concept is the source of a Directed Arc that has a
Terminator as its sink)
then
classify the concept as an Interface Object
fi

Rule: Classify User Role Interface Object

if
the concept is an Interface Object and
the concept is the source of an output and
the concept is the sink of an input and
the sink of the output and the source of the input is the same
User Role
then
classify the concept as a User Role Interface Object
fi

Rule: Classify Subsystem Interface Object

if
the concept is an Interface Object and
(the concept is the source of an output to an External Subsystem or
the concept is the sink of an input to an External Subsystem)
then
classify the concept as a Subsystem Interface Object
fi

Rule: Classify Device Interface Object

if
the concept is an Interface Object and
(the concept is the sink of an Input whose source is a Device or
the concept is the source of an Output whose sink is a Device or
the concept is the sink of an Interrupt whose source is a Device)
then
classify the concept as a Device Interface Object
fi

Rule: Classify Event Flow

if
the concept is a Dashed Directed Arc and
the Dashed Directed Arc has a name
then
classify the concept as an Event Flow
fi

Rule: Classify Normally-Named Event Flow

if
the concept is an Event Flow and
the concept name is not Trigger or Enable or Disable
then
classify the concept as a Normally-Named Event Flow
fi

Rule: Classify Interrupt

if
the concept is a Normally-Named Event Flow and
the source of the Normally-Named Event Flow is a Device
then
classify the concept as an Interrupt
fi

Rule: Classify Signal

if
the concept is a Normally-Named Event Flow and
the source of the Normally-Named Event Flow is a Solid Transform
then
classify the concept as a Signal
fi

Rule: Classify Specially-Named Event Flow

if
the concept is an Event Flow and
the name of the Event Flow is Enable or Disable or Trigger
then
classify the concept as a Specially-Named Event Flow
fi

Rule: Classify Timer

if
the concept is a Normally-Named Event Flow and
the source of the Normally-Named Event Flow is System and
the sink of the Normally-Named Event Flow is a Solid Transform
then
classify the concept as a Timer
fi

Rule: Classify Control Event Flow

if
the concept is a Specially-Named Event Flow
then
classify the concept as a Control Event Flow
fi

Rule: Classify Enable

if
the concept is a Control Event Flow and
the name is Enable and
the sink of the Control Event Flow is a Solid Transform and
the source of the Control Event Flow is a Control Object
then
classify the concept as an Enable
fi

Rule: Classify Disable

if
the concept is a Control Event Flow and
the name is Disable and
the sink of the Control Event Flow is a Solid Transform and
the source of the Control Event Flow is a Control Object
then
classify the concept as a Disable
fi

Rule: Classify Trigger

if
the concept is a Control Event Flow and
the name is Trigger and
the sink of the Control Event Flow is a Solid Transform and
the source of the Control Event Flow is a Control Object
then
classify the concept as a Trigger
fi

Rule: Classify Data-Store Data Flow

if
the concept is a Solid Directed Arc and
the source or sink of the Solid Directed Arc is a Data Store
then
classify the concept as an Data-Store Data Flow
fi

Rule: Classify Store

if
the concept is an Data-Store Data Flow and
the source of the Data-Store Data Flow is a Solid Transform and
the sink of the Data-Store Data Flow is a Data Store
then
classify the concept as a Store
fi

Rule: Classify Retrieve

if
the concept is an Data-Store Data Flow and
the sink of the Data-Store Data Flow is a Solid Transform and
the source of the Data-Store Data Flow is a Data Store
then
classify the concept as a Retrieve
fi

Rule: Classify Update1

if
the concept is a Solid Two-Way Arc and
the left end connects to a Solid Transform and
the right end connects to a Data Store
then
classify the concept as an Update
fi

Rule: Classify Update2

if
the concept is a Solid Two-Way Arc and
the right end connects to a Solid Transform and
the left end connects to a Data Store
then
classify the concept as an Update, reversing the right end and left end in the process
fi

Rule: Classify External Data Flow

if
the concept is a Solid Directed Arc and
the Solid Directed Arc has a name and
the Solid Directed Arc has a source or a sink that is a Terminator
then
classify the concept as an External Data Flow
fi

Rule: Classify Input

if
the concept is an External Data Flow and
the source of the External Data Flow is a Terminator and
the sink of the External Data Flow is an Interface Object
then
classify the concept as an Input
fi

Rule: Classify Output

if
the concept is an External Data Flow and
the sink of the External Data Flow is a Terminator and
the source of the External Data Flow is an Interface Object
then
classify the concept as an Output
fi

Rule: Classify Internal Data Flow

if
the concept is a Solid Directed Arc and
the Solid Directed Arc has a name and
the source and sink of the Solid Directed Arc are both Solid Transforms
then
classify the concepts as an Internal Data Flow
fi

A.2.2 Rules for Transformation Classification

Rule: Classify Function

if
the concept is a Solid Transform and not an Interface Object and
the Solid Transform has a name that is not "System"
then
classify the concept as a Function
fi

Rule: Classify State-Dependent Function

if
the concept is a Function and
(the Function is the sink for a Control Event Flow or
the Function is the sink for a Signal that has a Control Object as its source)
then
classify the concept as a State-Dependent Function
fi

Rule: Classify Enabled Function

if
the concept is a State-Dependent Function and
the State-Dependent Function is the sink for an Enable
then
classify the concept as an Enabled Function
fi

Rule: Classify Enabled-Periodic Function

if
the concept is an Enabled Function and
the Enabled Function is the sink for a Timer
then
classify the concept as an Enabled-Periodic Function
fi

Rule: Classify Enabled-Asynchronous Function

if
the concept is an Enabled Function and
the Enabled Function is not the sink for a Timer
then
classify the concept as an Enabled-Asynchronous Function
fi

Rule: Classify Triggered Function

if
the concept is a State-Dependent Function and
(the concept is the sink for a Trigger or
the concept is the sink for a Signal whose source is a Control Object)
then
classify the concept as a Triggered Function
fi

Rule: Classify Triggered-Periodic Function

if
the concept is a Triggered Function and
the concept is the sink for a Timer
then
classify the concept as a Triggered-Periodic Function
fi

Rule: Classify State-Independent Function

if
the concept is a Function and
the Function is not the sink for a Control Event Flow and
the Function is not the sink for a Signal whose source is a Control Object
then
classify the concept as a State-Independent Function
fi

Rule: Classify Periodic Function

if
the concept is a State-Independent Function and
the State-Independent Function is the sink for a Timer
then
classify the concept as a Periodic Function
fi

Rule: Classify Aperiodic Function

if
the concept is a State-Independent Function and
the State-Independent Function is not the sink for a Timer
then
classify the concept as an Aperiodic Function
fi

Rule: Classify Device Input Object

if
the concept is a Device Interface Object and
the Device Interface Object is the sink for an Input or Interrupt and
the Device Interface Object is not the source for an Output
then
classify the concept as a Device Input Object
fi

Rule: Classify Device Output Object

if
the concept is a Device Interface Object and
the Device Interface Object is the source for an Output and
the Device Interface Object is not the sink for an Input
then
classify the concept as a Device Output Object
fi

Rule: Classify Device IO Object

if
the concept is a Device Interface Object and
the Device Interface Object is the source for an Output and
the Device Interface Object is the sink for an Input
then
classify the concept as a Device IO Object
fi

Rule: Classify Periodic Device Interface Object

if
the concept is a Device Interface Object and
the Device Interface Object is the sink for a Timer and
the Device Interface Object is not the sink for an Interrupt
then
classify the concept as a Periodic Device Interface Object
fi

Rule: Classify Asynchronous Device Interface Object

if
the concept is a Device Interface Object and
the Device Interface Object is the sink for an Interrupt and
the Device Interface Object is not the sink for a Timer
then
classify the concept as an Asynchronous Device Interface Object
fi

Rule: Classify Passive Device Interface Object

if
the concept is a Device Interface Object and
the Device Interface Object is not the sink for an Interrupt and
the Device Interface Object is not the sink for a Timer
then
classify the concept as a Passive Device Interface Object
fi

Rule: Classify Periodic Device Input Object Path1

if
the concept is a Device Input Object and
the Device Input Object is the sink for a Timer and
the Device Input Object is not the sink for an Interrupt
then
classify the concept as a Periodic Device Input Object
fi

Rule: Classify Asynchronous Device Input Object Path 1

if
the concept is a Device Input Object and
the Device Input Object is not the sink for a Timer and
the Device Input Object is the sink for an Interrupt
then
classify the concept as an Asynchronous Device Input Object
fi

Rule: Classify Passive Device Input Object Path 1

if
the concept is a Device Input Object and
the Device Input Object is not the sink for a Timer and
the Device Input Object is not the sink for an Interrupt
then
classify the concept as a Passive Input Device
fi

Rule: Classify Periodic Device Output Object Path1

if
the concept is a Device Output Object and
the Device Output Object is the sink for a Timer and
the Device Output Object is not the sink for an Interrupt
then
classify the concept as a Periodic Device Output Object
fi

Rule: Classify Asynchronous Device Output Object Path 1

if
the concept is a Device Output Object and
the Device Output Object is not the sink for a Timer and
the Device Output Object is the sink for an Interrupt
then
classify the concept as an Asynchronous Device Output Object
fi

Rule: Classify Passive Device Output Object Path 1

if
the concept is a Device Output Object and
the Device Output Object is not the sink for a Timer and
the Device Output Object is not the sink for an Interrupt
then
classify the concept as a Passive Output Device
fi

Rule: Classify Periodic Device IO Object Path1

if
the concept is a Device IO Object and
the Device IO Object is the sink for a Timer and
the Device IO Object is not the sink for an Interrupt
then
classify the concept as a Periodic Device IO Object
fi

Rule: Classify Asynchronous Device IO Object Path 1

if
the concept is a Device IO Object and
the Device IO Object is not the sink for a Timer and
the Device IO Object is the sink for an Interrupt
then
classify the concept as an Asynchronous Device IO Object
fi

Rule: Classify Passive Device IO Object Path 1

if
the concept is a Device IO Object and
the Device IO Object is not the sink for a Timer and
the Device IO Object is not the sink for an Interrupt
then
classify the concept as a Passive IO Device
fi

Rule: Classify Periodic Device Input Object Path 2

if
the concept is a Periodic Device Object and
the Periodic Device Object is the sink for an Input and
the Periodic Device Object is not the source for an Output
then
classify the concept as a Periodic Device Input Object
fi

Rule: Classify Asynchronous Device Input Object Path 2

if
the concept is an Asynchronous Device Object and
the Asynchronous Device Object is the sink for an Input and
the Asynchronous Device Object is not the source for an Output
then
classify the concept as an Asynchronous Device Input Object
fi

Rule: Classify Passive Device Input Object Path 2

if
the concept is a Passive Device Object and
the Passive Device Object is the sink for an Input and
the Passive Device Object is not the source for an Output
then
classify the concept as a Passive Device Input Object
fi

Rule: Classify Periodic Device Output Object Path 2

if
the concept is a Periodic Device Object and
the Periodic Device Object is not the sink for an Input and
the Periodic Device Object is not he source for an Output
then
classify the concept as a Periodic Device Output Object
fi

Rule: Classify Asynchronous Device Output Object Path 2

if
the concept is an Asynchronous Device Object and
the Asynchronous Device Object is not the sink for an Input and
the Asynchronous Device Object is the source for an Output
then
classify the concept as an Asynchronous Device Output Object
fi

Rule: Classify Passive Device Output Object Path 2

if
the concept is a Passive Device Object and
the Passive Device Object is not the sink for an Input and
the Passive Device Object is the source for an Output
then
classify the concept as a Passive Device Output Object
fi

Rule: Classify Periodic Device IO Object Path 2

if
the concept is a Periodic Device Object and
the Periodic Device Object is the sink for an Input and
the Periodic Device Object is the source for an Output
then
classify the concept as a Periodic Device IO Object
fi

Rule: Classify Asynchronous Device IO Object Path 2

if
the concept is an Asynchronous Device Object and
the Asynchronous Device Object is the sink for an Input and
the Asynchronous Device Object is the source for an Output
then
classify the concept as an Asynchronous Device IO Object
fi

Rule: Classify Passive Device IO Object Path 2

if
the concept is a Passive Device Object and
the Passive Device Object is the sink for an Input and
the Passive Device Object is the source for an Output
then
classify the concept as a Passive Device IO Object
fi

A.2.3 Rules for Stimulus-Response Classification

Rule: Classify Stimulus 1

if
    the concept is an Internal Data Flow and
    a Passive Device Interface Object is the sink for the Internal Data Flow
then
    classify the concept as a Stimulus
fi

Rule: Classify Stimulus 2

if
    the concept is an Internal Data Flow and
    there exists no Internal Data Flow or Signal that flows in the reverse
        direction
then
    classify the concept as a Stimulus
fi

Rule: Classify Stimulus 3

if
    the concept is an Internal Data Flow and
    no other Internal Data Flow flows in the reverse direction from a
        Passive Device Interface Object and
    (no Signal has the same sink as the Internal Data Flow or
    no other Internal Data Flow has the same sink as, and a different name from,
        the Internal Data Flow)
then
    classify the concept as a Stimulus
fi

Rule: Classify Stimulus 4

if
the concept is an Internal Data Flow and
a Response flows in the reverse direction
then
classify the concept as a Stimulus
fi

Rule: Classify Response 1

if
the concept is an Internal Data Flow and
a Stimulus or Signal flows in the reverse direction
then
classify the concept as a Response
fi

Rule: User Classifies Internal Data Flow (Last Preference)

if
the concept is an Internal Data Flow and
another Internal Data Flow flows in the reverse direction
then
ask the user to indicate, if possible, whether one of the two Internal Data Flows is a Response or whether each is a Stimulus
if the user provides guidance
then classify each Internal Data Flow as the user requests
else classify each Internal Data Flow as a Stimulus
fi

A.2.4 Rules for Ambiguous-Function Classification

Rule: Classify Asynchronous Function (First Preference)

if
the concept is an Aperiodic Function and
(the Aperiodic Function receives Signals from different sources or
receives a Stimulus and a Signal from different sources or
receives Stimuli with different names from different sources or
sends Responses with different names to different sinks)
then
classify the concept as an Asynchronous Function
fi

Rule: Classify Triggered Synchronous Function (First Preference)

if
the concept is a Triggered Function and
the Triggered Function does not receive a Stimulus or Signal from a Solid Transform and
the Triggered Function does not send a Signal to its triggering Control Object
then
classify the concept as a Triggered Synchronous Function
fi

Rule: Classify Triggered Asynchronous Function (First Preference)

if
the concept is a Triggered Function and
the Triggered Function sends a Signal to its triggering Control Object
then
classify the concept as a Triggered Asynchronous Function
fi

Rule: User Classifies Triggered Function (Second Preference)

if
the concept is a Triggered Function
then
ask user whether the Triggered Function completes during the
triggering state transition
if the Triggered Function completes during the triggering state
transition or the user doesn't know
then classify the concept as a Triggered Synchronous Function
else classify the concept as a Triggered Asynchronous Function
fi
fi

Rule: Classify Synchronous Function1 (Second Preference)

if
the concept is an Aperiodic Function and
the Aperiodic Function receives a Stimulus from a Transformation and
the Aperiodic Function sends a Response to the same Transformation
then
classify the concept as a Synchronous Function
fi

Rule: Classify Synchronous Function2 (Second Preference)

if
the designer is experienced
the concept is an Aperiodic Function and
((the Aperiodic Function sends no Stimulus and sends no Signal) or
(the Aperiodic Function sends a Stimulus or Signal to no Transformation
that is not a Passive-Device-Interface-Object))
then
ask the designer to confirm this classification as a Synchronous Function
if the designer confirms this classification or doesn't know
then classify the concept as a Synchronous Function
else classify the concept as an Asynchronous Function
fi

Rule: User Classifies Aperiodic Function (Last Preference)

```
if
        the concept is an Aperiodic Function
then
        ask the user whether another function stops execution waiting
                for the results of this Aperiodic Function
        if      another function waits
        then    classify the concept as a Synchronous Function
        else    ask the user whether this Aperiodic Function executes quickly or not
                if      the function executes quickly
                then    classify the concept as a Synchronous Function
                else    classify the concept as an Asynchronous Function
                fi
        fi
fi
```