AllFusion® Process Modeler
Process Flow Modeling

Design Guide

r7.2

CA Product References

This document references the following CA products:

- AllFusion® Process Modeler (AllFusion PM)

Contact Technical Support

For online technical assistance and a complete list of locations, primary service hours, and telephone numbers, contact Technical Support at http://ca.com/support.

Contents

Chapter 1: Process Flow Modeling Method 7

Process Flow Modeling 7
IDEF3 Model Structure 8
IDEF3 Process Description Capture 28

Chapter 2: Building Process Flow Models 31

Create an IDEF3 Model 31
Add More Activities to an IDEF3 Diagram 33
Set Activity Properties 34
Add Junctions to an IDEF3 Diagram 35
Connect Junctions to Activity Boxes Using Arrows 36
Add Referents to an IDEF3 Diagram 38
Connect a Referent to an Activity 39
Define the Referent Name 40
Set Referent Properties 41

Chapter 3: Additional IDEF3 Features 43

Organization Visualization 43
Swim Lane Diagrams 44
Organization Charts 45
How To Use Swim Lane Diagrams and Organization Charts in Your Models 46
Create and Manage Role Groups 47
Create and Manage Roles 48
Create and Manage Resources 49
Create a Process Role in the UDP Dictionary 50
Add a Bitmap to the Bitmap Dictionary 50
Create a Swim Lane Diagram 51
Create an Organization Chart 52
Create an IDEF3 Scenario Diagram 57
Simulation 59
Sources and Destinations in a Simulation Model 60
Queues in a Simulation Model 60
Facilities in a Simulation Model 60
Simulation Examples 61
Simulation Formulas 64
Simulation Experiments 69

IDEF3 Process Flow Networks to Arena BE Mappings 70
Base Mapping Table 71
Arena Create Module and AllFusion PM Referent (Link From) 72
Arena Process Module and AllFusion PM UOB (IDEF3 Activity) 73
Arena Decide Module and AllFusion PM Junction (XOR or Sync OR) 74
Arena Batch Module and AllFusion PM Junction (AND or Async OR) 76
Arena Separate Module and AllFusion PM Junction (AND or Async OR) 76
Arena Assign Module and AllFusion PM UOB 77
Arena Resource Module and AllFusion PM UOB 78
Arena Resource Repeat Group and AllFusion PM Arrow 79
AllFusion PM to Arena Sample Models 80

Index **81**

Chapter 1: Process Flow Modeling Method

This section contains the following topics:

Process Flow Modeling (see page 7)

Process Flow Modeling

Process flow modeling (IDEF3) provides a structure for depicting a sequence of events. It graphically represents the flow, relationships, and objects of a process.

Use IDEF3 to accomplish the following tasks:

- Model incomplete processes for further analysis.
- Reengineer business processes.
- Develop deliverable measurements.
- Collect policy and procedure information.
- Model real-life scenarios (like emergency procedures or contingency plans).

IDEF3 Model Structure

The business scenario is the basic organizing structure of an IDEF3 model. The IDEF3 model outlines a sequence of activities, or a process within a given setting. Because a scenario describes the purpose and scope of a model, it is important that the main activity be appropriately named with a verb, gerund (verb ending in *ing*), or a verb phrase (verb + direct object). Process Customer Order, Implement New Design, and Develop Customer Response Profile are all examples of scenario names.

Include the following items in the structure of an IDEF3 model:

Viewpoint

Identifies the role or job title of the person providing the description.

Purpose

Identifies questions the model should answer.

Scope

Identifies what to include and exclude from the model.

Audience

Identifies who will use model.

IDEF3 Diagrams

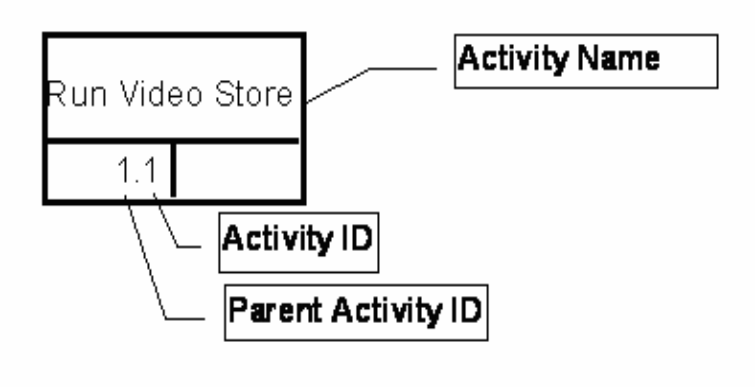
The IDEF3 diagram is the basic organizing unit of an IDEF3 model, as in every activity modeling technique described in this guide. The organization of the diagrams in an IDEF3 model is more important if the model is to be published or read by others, as would be the case for most design models. The modeler needs to make sure to determine what information is included in a particular IDEF3 diagram, in order to ensure that the diagram is comprehensive, and clear to the reader.

Unit of Work (UOW)/Activity

As in all activity modeling techniques, the activity, also called unit of work (UOW), is the central component of the model. It refers to a process, action, decision, or other procedure performed in a system or business within an IDEF3 model. UOWs in IDEF3 modeling are equivalent to Activities in IDEF0 (business process) modeling. An IDEF3 diagram depicts an activity as a box with square corners. In an IDEF3 diagram, the activity is identified with a verb or verb phrase (verb plus direct object), and a unique number identifier.

The noun part of the verb phrase, that is, the direct object of the verb, usually describes the major input to the activity (such as *Gather Data*), the major output of the activity (such as *Write Book*), or the name of the system (*Run Video Store*). Sometimes, the noun is changed during modeling because a different noun becomes the more accepted version, or the verb might be replaced with something more precise. When an activity is first created, it is given a unique number. Even if the activity is subsequently deleted, the number is not reused.

In an IDEF3 diagram, the activity number is usually preceded by the number of its parent activity, as shown in the following illustration:



Links in IDEF3

Links denote significant constraining relationships among activities. All links in IDEF3 are unidirectional (except when you choose the option to use bi-directional links in IDEF3), and although an arrow may originate or terminate at any side of an activity box, IDEF3 diagrams are generally organized from left to right so that links normally originate from the right side and terminate at the left side of activity boxes. The following describes the three types of links available in IDEF3 and their usage in IDEF3 modeling:

Precedence Link

Defines that a source activity must complete before the destination activity can begin.

Object Flow Link

Defines that the output of the source activity is input to the destination activity. This implies that the source activity must complete before the destination activity can begin.

Relational Link

Defines that the constraining relationship between the source and destination activities must be user-defined for each instance of a relational link.

Bi-directional Link

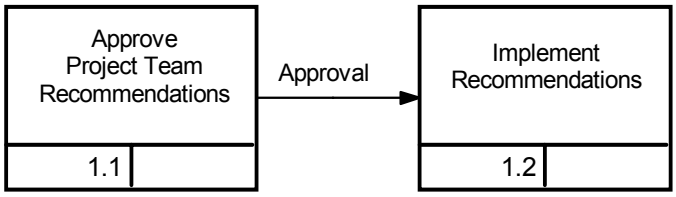
Defines bi-directional data flow between DFD objects, however you can choose to allow bi-directional arrows in IDEF3 modeling.

Referent

Connects outputs of UOWs to junctions as inputs and connects referents to UOWs.

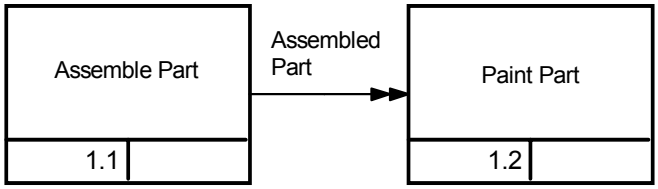
IDEF3 Precedence Link

A precedence link denotes that the source activity must complete before the destination activity can begin. A precedence link appears in IDEF3 diagrams as an arrow, with the arrowhead pointing from the source activity to the destination (triggered) activity. The link should be labeled so that the reader can understand why the link exists. In many cases, the completion of one activity enables or triggers the activation of another, as shown in the illustration that follows. In this example, management must approve the project team's recommendations before they can be implemented.



IDEF3 Object Flow Link

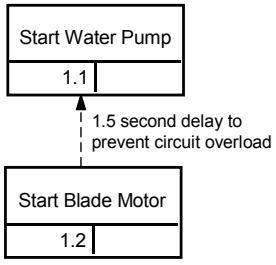
One of the most common reasons for a temporal precedence link between two activities is that some object, produced by the source activity, is required by the destination activity. The object flow link is distinguished from the generic temporal precedence link by its double arrowhead. These links should be named to clearly identify the object that flows along this link. Object flow links have the same temporal semantics as the precedence link; that is, the activity from which the object flow link originates must complete before the activity to which the object flow link points can start, as shown in the illustration that follows. In this example, the assembled part is the object that is produced by the source activity. It must be assembled before it can be painted.



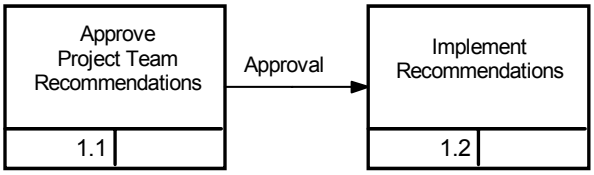
IDEF3 Relational Link

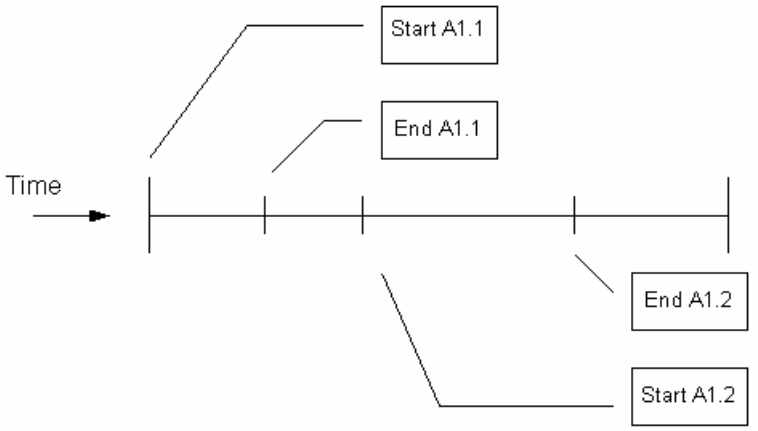
Relational links denote relationships that do not imply temporal precedence or object flow. The meaning of each relational link must be defined, because the relational link imposes no constraints of its own.

You can also use relational links to denote relationships between parallel activities. The figure that follows illustrates part of the process of starting a water-cooled saw and the special relationship between the activities *Start Blade Motor* and *Start Water Pump*. An arrow label can be used to describe the nature of the relationship, and a more thorough description could be captured as supplemental text.

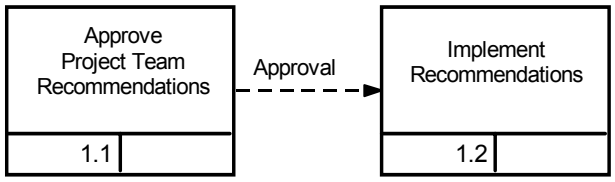


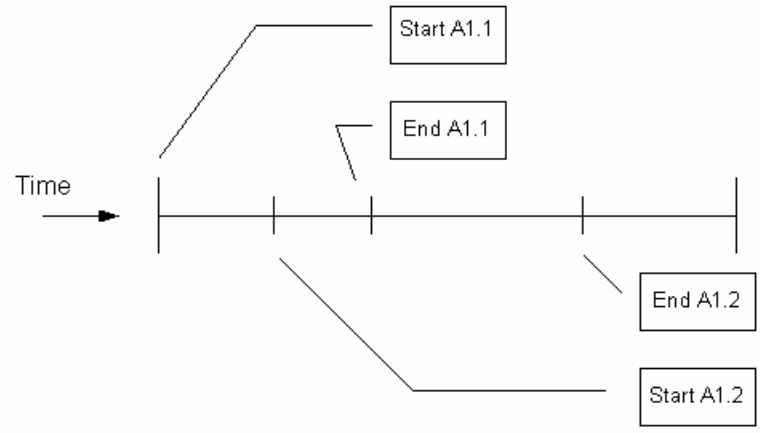
A common use of the relational link is to describe special cases of the precedence link, that is, alternate temporal relationships between activities. In the following two examples, the first is a typical temporal precedence link. In the second example, the vertical bars show the relationship between the start and end times for the two activities in the first example, implied by a temporal precedence link. In this example, as time moves from left to right, the start of *Implement Recommendations* occurs AFTER the completion of *Approve Project Team Recommendations*:



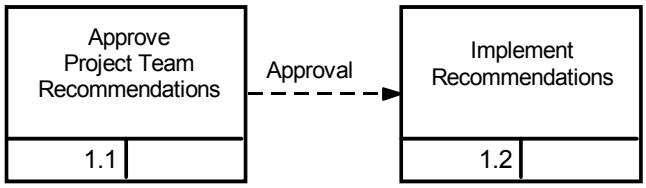


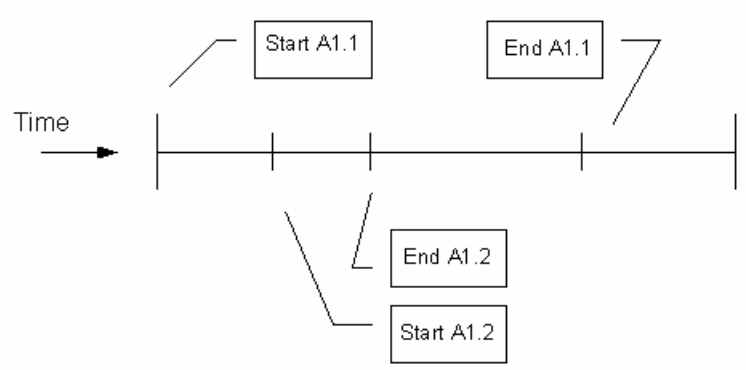
An alternative temporal constraint between the two activities in the first figure is shown in the second figure. In this example, the project team starts to *Implement Recommendations* before *Approve Project Team Recommendations* ends. A relational link is shown in the first figure:





It is important to clearly document the temporal constraint between two activities connected by a relational link. Consider another possible temporal constraint, shown in the second figure. Applied to the example shown in the first figure, the project team would start to *Implement Recommendations* after the activity *Approve Project Team Recommendations* begins, but *Implement Recommendations* would finish before *Approve Project Team Recommendations* finishes.





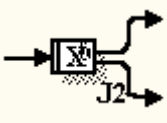
The temporal constraints shown in the previous figure and the figure above are both plausible, and the correct interpretation must be documented. It is important to emphasize that *correct* in this case means the interpretation that accurately reflects the situation being documented, and not the interpretation that in the analyst's view would lead to a more effective process.

Junctions in IDEF3

Junctions are used in process flow diagrams to show branching or joining in the process logic. The completion of one activity may enable several other activities to start, or an activity may have to wait for several activities to finish before it can start. Junctions either distribute or consolidate process flow and are used to describe process branching. There are two classifications of junctions to describe distribution or consolidation of process flow:

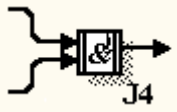
Fan-out Junction

Branches one arrow into multiple arrows to distribute process flow; the completion of one activity causes the activation of other activities.

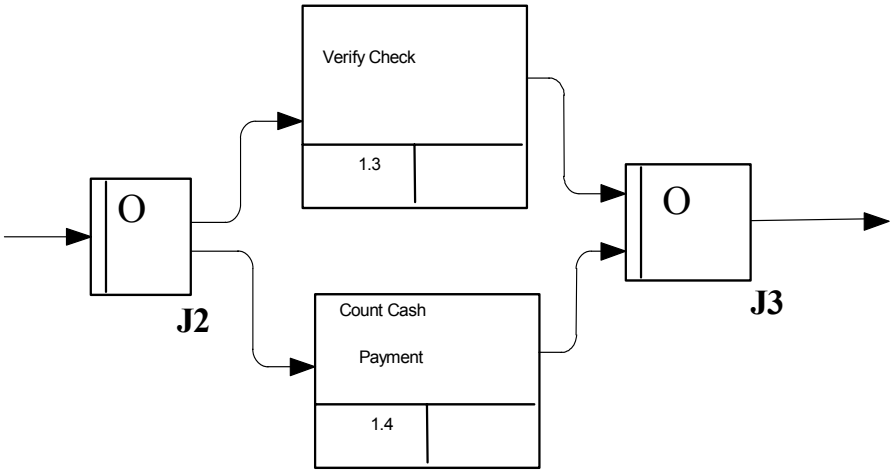


Fan-in Junction

Consolidates multiple arrows into a single arrow to consolidate process flow; the completion of one or more activities causes the activation of a single activity.



Fan-out (J2) and fan-in (J3) junctions are shown in the following illustration:



Note: A junction cannot be both fan-in and fan-out at the same time.

The following describes the three types of junctions available in IDEF3:

AND Junction

Activates every destination activity to which it connects.

Exclusive-OR Junction

Activates only one activity at any one time, regardless of the number of activities attached to an Exclusive-OR fan-in or fan-out junction, and therefore, only one completes before any activity following an Exclusive-OR fan-in junction can begin.

OR Junction

Captures activation combinations that cannot be described by AND (all) and Exclusive-OR (one and only one) junctions.

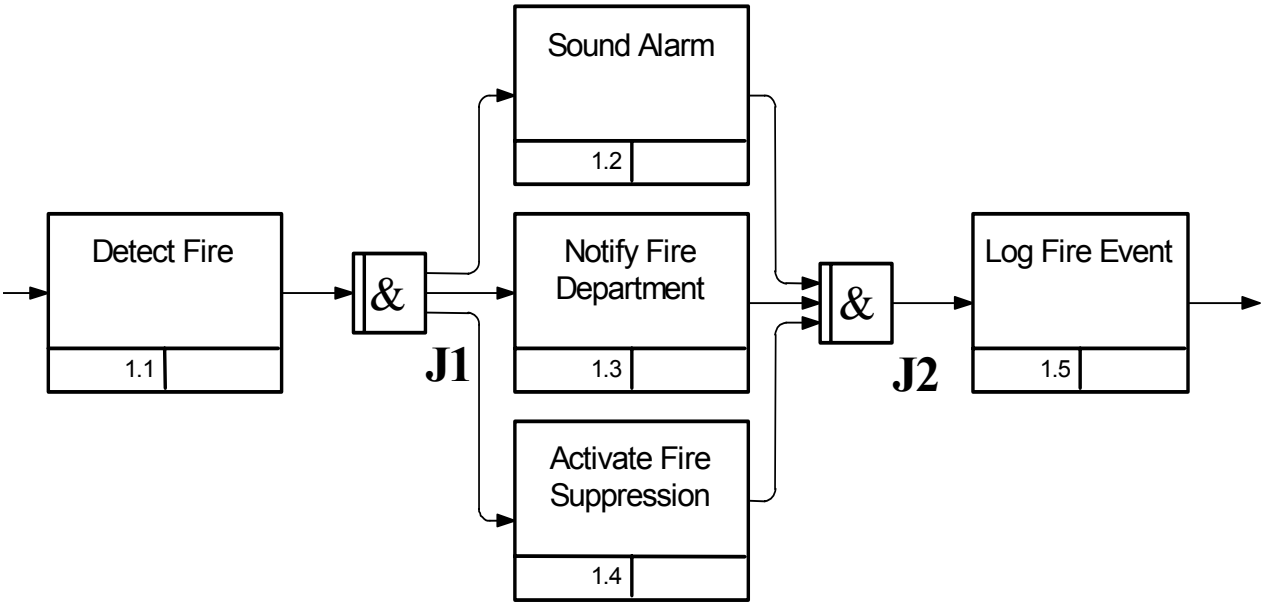
IDEF3 AND Junction

The AND junction always activates every destination activity to which it connects. All activities that connect to an AND fan-in junction must complete before the next activity can begin.

The following activation rules apply to the AND junction:

- For a fan-out AND junction, every destination activity connected to the AND junction is always activated.
- For a fan-in AND junction, every source activity connected to the AND junction must always complete.

In the following illustration, when *Detect Fire* completes, *Sound Alarm*, *Notify Fire Department*, AND *Activate Fire Suppression* are activated. When and only when all three of these activities complete, *Log Fire Event* is activated.



IDEF3 Exclusive-OR Junction

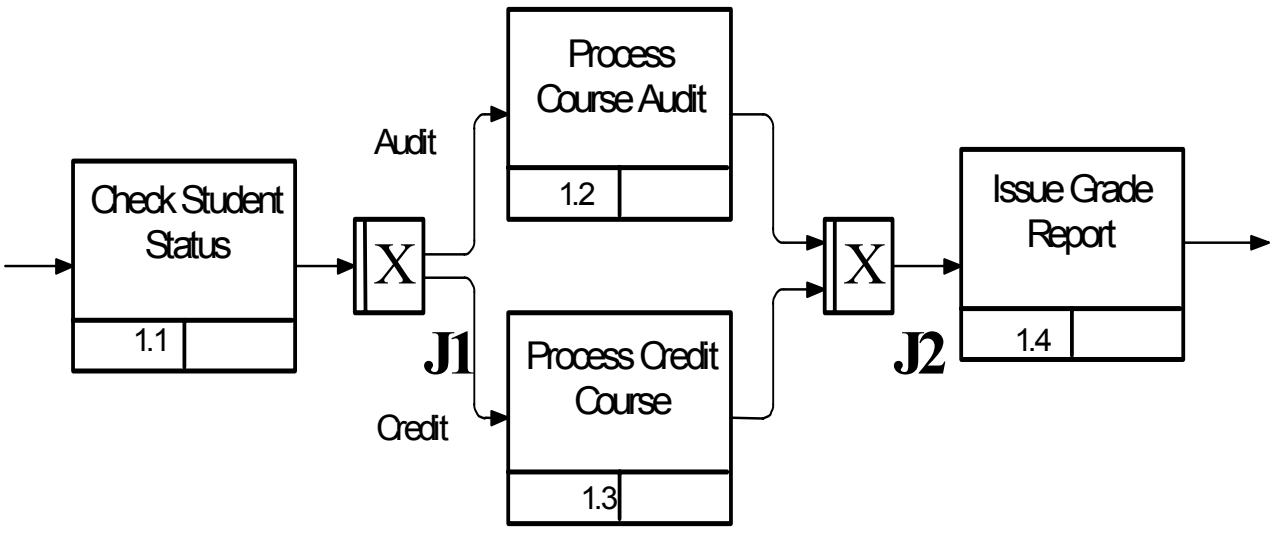
Regardless of the number of activities attached to an Exclusive-OR fan-in or fan-out junction, only one is activated at any one time, and therefore, only one completes before any activity following an Exclusive-OR fan-in junction can begin.

The following activation rules apply to the Exclusive-OR junction:

- For a fan-out Exclusive-OR junction, one and only one destination activity connected to the Exclusive-OR Junction is activated.
- For a fan-in Exclusive-OR junction, one and only one source activity connected to the Exclusive-OR junction must complete.

If the junction activation rules are known, then they should be captured either in the junction's elaboration (description), in a junction referent, or by labeling the arrows that emanate from the fan-out junction, as illustrated in the figure that follows.

This figure illustrates an Exclusive-OR junction is used to show that *Process Course Audit* and *Process Credit Course* are never activated at the same time. One and only one of these two activities are activated by *Check Student Status* because a student can either take a course for credit or audit a course, but never both.

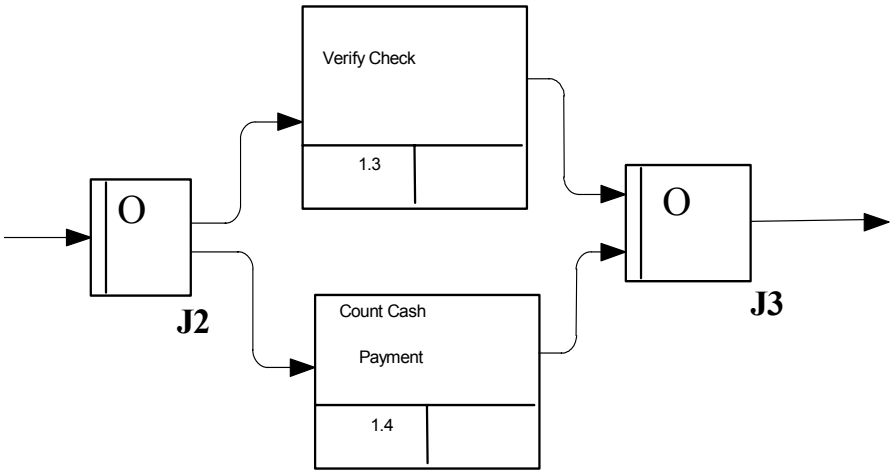


IDEF3 OR Junction

The OR junction captures activation combinations that cannot be described by AND (all) and Exclusive-OR (one and only one) junctions. Like the relational link, the OR junction is primarily user-defined. In the figure that follows, the OR junction J2 can activate Verify Check, or Count Cash Payment, or both. Verify Check is activated if the customer hands the teller a check, Count Cash Payment is activated if the customer hands the teller cash, and both are activated if the customer hands the teller both cash and a check.

The following activation rules apply to the OR junction:

- For a fan-out OR junction, one or more destination activities connected to the OR Junction are activated.
- For a fan-in OR junction, one or more source activities connected to the OR junction must complete.



Synchronous and Asynchronous Junctions

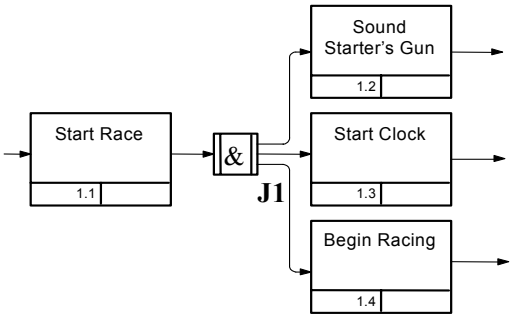
In the AND and OR junction examples, the relationship between the beginning and ending of the activities activated by the fan-out junctions was not discussed. In these examples, the activities were asynchronous, meaning that they did not have to begin or end at the same time. However, there are occasions when the start or end times (or both the start and end times) of parallel activities must be synchronous, meaning that they must occur at the same time. Synchronous junctions are used to model this behavior. The following table shows the proper interpretations for synchronous junctions:

Junction Type	Description
Fan-out AND Junction	All of the activities that fan out of the junction begin simultaneously.
Fan-in AND Junction	All of the activities that fan in to the junction complete simultaneously.
Fan-out OR Junction	One or more of the activities that fan out of the junction begin simultaneously.
Fan-in OR Junction	One or more of the activities that fan in to the junction complete simultaneously.
Fan-out Exclusive-OR Junction	Exactly one following process starts. Because one and only one activity connected to an Exclusive-OR fan-out junction is activated, synchronicity with other activities is impossible.
Fan-in Exclusive-OR Junction	Exactly one preceding process completes. Because one and only one activity connected to an Exclusive-OR fan-in junction completes, synchronicity with other activities is impossible.

A synchronous junction is denoted by the two vertical bars inside the junction box, as opposed to the single vertical bar indicative of an asynchronous junction.

In many kinds of competitive races, the starter's gun must sound, the clock must start, and the participants must begin racing at the same time. Otherwise, the race is not considered to be a fair contest.

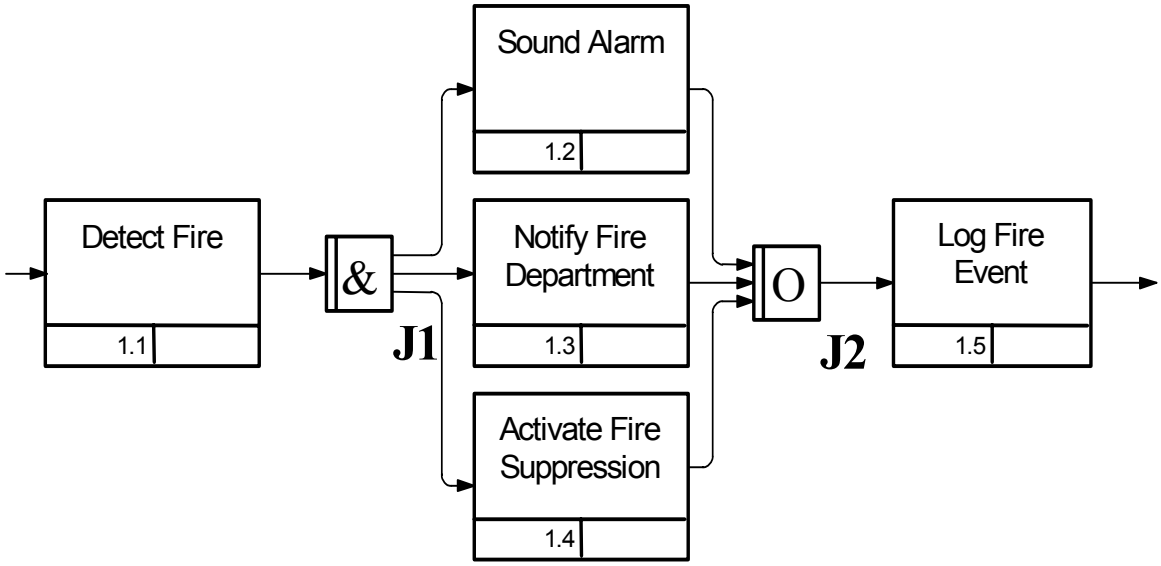
The following example illustrates this by using a synchronous AND junction:



If a junction is synchronous, then the modeler should note the tolerance in the timing constraint; that is, how closely to one another the activities have to begin or end. Also, it is not required that a synchronous fan-out junction be paired with a matching synchronous fan-in junction. Certainly, activities can begin together but not end together, such as in the race example, where the main attraction is the decidedly asynchronous finish. It is also possible for activities to begin asynchronously and end synchronously.

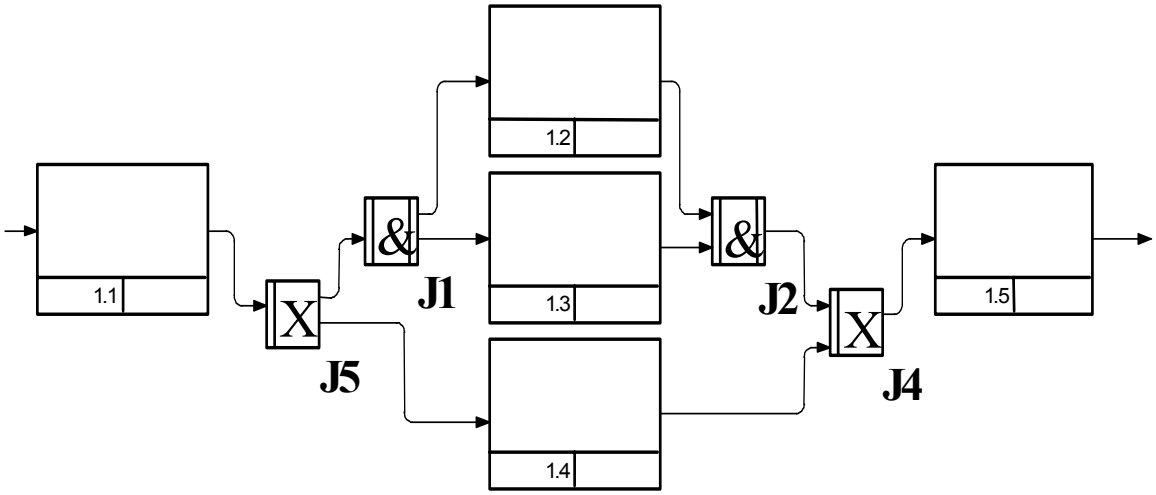
Junction Pairs

On a diagram, junctions should be paired, that is, every fan-out junction has a paired fan-in junction. However, it is not required that the junction type be the same. In the following illustration, an AND fan-out junction is matched with an OR fan-in junction. The AND junction (J1) is interpreted to mean the same as in the figure. When *Detect Fire* completes, *Sound Alarm*, *Notify Fire Department*, AND *Activate Fire Suppression* are activated. The OR junction (J2) is interpreted as follows: When any or all of the *Sound Alarm*, *Notify Fire Department*, and *Activate Fire Suppression* complete, Log Fire Event activates.



Junction Combinations

Junctions can be combined to create more complex branching rules, as shown in the figure below. Junction combinations must be used judiciously, with a clear understanding of the purpose of the document being the ultimate guideline for deciding whether a particular junction combination will clarify or merely clutter a diagram. Complicated junction structures can be nested inside activity boxes.



Referents in IDEF3

A referent is a term used to describe an object in an IDEF3 diagram where additional information is stored outside the process flow. For example, if a credit check were processed and a determination was made to set the credit rating as low, the information from that credit check would reside in a Bad Credit List. In this case, the Bad Credit List is considered a referent.

Referents are used in IDEF3 modeling to support junctions and other process flow objects, or to represent repeating UOWs. They are special symbols that refer to other parts of a process description. They are added to a diagram to direct the reader's attention to something important. The following describes the types of referents and their purpose:

OBJECT

Describes the participation of an important object in an activity.

GOTO

Implements looping (repeating a sequence of activities), possibly on the same diagram but not necessarily so. If the activities are all on the same diagram, looping can also be depicted by drawing an arrow back to the starting activity. A GOTO referent can also refer to a junction.

UOB (Unit of Behavior)

Includes another instance of an activity without looping. For example, if the activity Count Cash occurs several times within a process, the first occurrence of Count Cash can be created as an activity and subsequent occurrences drawn as UOB referents. The use of this referent type is normally not required when using automated tools.

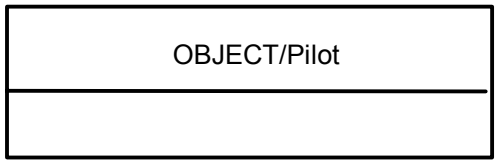
NOTE

Documents any important but general information that relates to some graphic on the diagram. In this regard, NOTE referents serve as an alternative to recording text notes directly on a diagram.

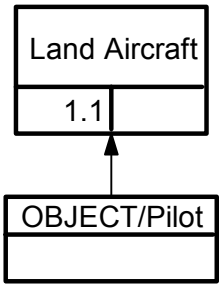
ELAB (elaboration)

Elaborates on a graphic, or describes it in more detail. Elaboration referents are commonly used to describe the branching logic of a junction.

A referent is depicted as a box (much like an activity). The referent name usually includes the referent type and an identifier. The following figure illustrates an Object referent:



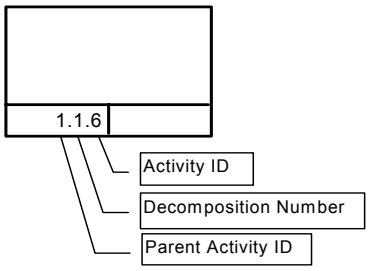
The following figure illustrates an object/activity relationship:



Activity Decomposition

IDEF3 activities may be decomposed to express more detail. The IDEF3 method allows an activity to be decomposed *multiple* times; that is, have multiple children. This allows a single model to document alternative process flows.

To properly track activities in a multiple decomposition model, the numbering scheme must be extended to include the decomposition number as well as the activity ID and parent activity ID. This is illustrated in the following figure, which shows an activity identifier that includes the decomposition number.



Activity Decomposition Diagrams

You can use activity decomposition diagrams in business modeling to break an activity down into its constituent parts. For example, the activity *Run Video Store* can be decomposed into activities such as *Open Store*, *Receive Payment*, *Rent Video*, and *Close Store*. Each of these activities can also be decomposed into their constituent activities.

You can decompose activities (in IDEF0 and DFD modeling) or Units of Work (in IDEF3 modeling). Units of Work (UOWs) indicate an event, process, decision, or action. Although the methodologies differ, the basic premise of decomposition is the same. You can decompose IDEF0 models into IDEF3 and DFD constituent activities and Units of Work.

Note: The IDEF0 diagram must always be the parent activity of any mixed model decomposition diagram. It is a good idea to have at least two levels of IDEF0 activities before decomposing into another methodology. That way, you have enough activities to create child decomposition diagrams.

Create an Activity Decomposition Diagram

You can decompose a context activity when you create a decomposition diagram. This is helpful when you need greater detail for a more accurate representation of the system you are modeling.

To create an activity decomposition diagram

1. Select the activity that you want to decompose and then click the Go to Child Diagram tool ▼ on the AllFusion PM toolbar.

   The Activity Box Count dialog opens.

2. Select one of the following decomposition types and any additional options and click OK:

   IDEF0

   Specifies to create an IDEF0 decomposition diagram.

   DFD

   Specifies to create a DFD decomposition diagram.

   If you select to create a DFD decomposition, you can select the following:

   Include Externals and Data Stores

   (Optional) Specifies to include externals and data stores in the decomposition diagram.

IDEF3

Specifies to create an IDEF3 decomposition diagram.

Number of Activities in this Decomposition (0-8)

Specifies the number of activities between zero and eight to create in the decomposition diagram.

The Activity Box Count dialog closes and the decomposition diagram opens.

3. Double-click one of activity boxes that you just created.

   The Activity Properties dialog opens at the Name tab.

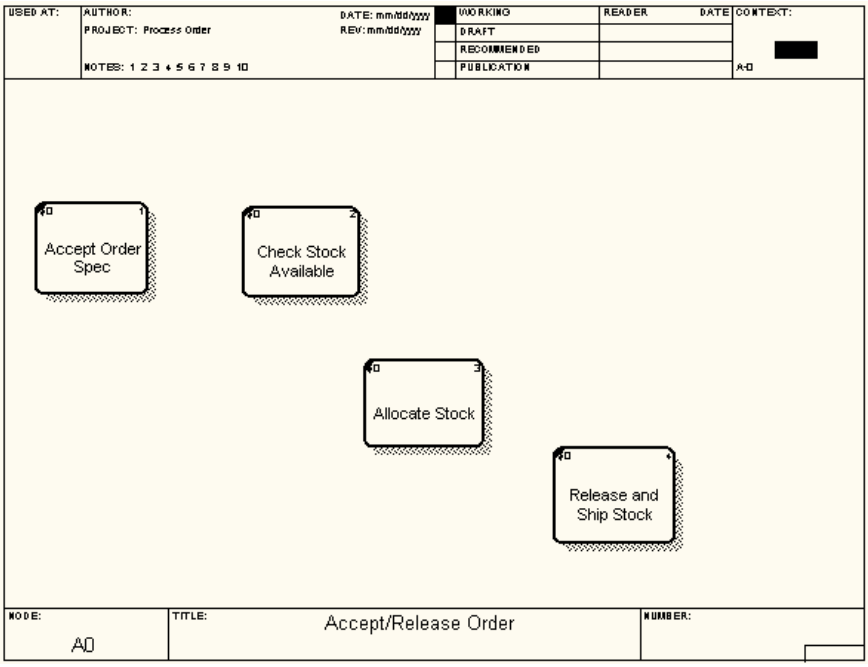
4. Enter a name for the activity or select an existing name from the Unused Activity Names list, and click OK.

   The Activity Properties dialog closes and the new name for that activity displays in the activity box.

   Note: Double-click each activity box for which you need to enter a name. You can reposition the activity boxes on the diagram as needed. You will also need to connect the existing arrow objects to the activities in the decomposition.

Example: Activity Decomposition Diagram

The following is an example of an activity decomposition diagram in the beginning stage, showing a decomposition of four activities:



IDEF3 Process Description Capture

IDEF3 process description capture is the process of building an IDEF3 diagram from a text-based process description. A modeler or analyst creates the IDEF3 diagram, and one or more subject experts provide the information from which to derive the process description.

Definition of Activities and Objects

The subject experts will typically provide a textual narrative that describes the scenario of interest. In addition, written documentation may already exist to aid in understanding the current process. Regardless of whether the information is written or verbal, it is analyzed and separated into parts of speech in order to identify a candidate list of activities (verbs and verb phrases) that constitute the process and objects (nouns and noun phrases) that participate in the process. Although the process sounds straightforward, there are certain pitfalls, such as adjectives that can be verb forms as well (at least in English). An example of this is *test procedure*, where it is unclear whether *test* is an adjective describing procedure, or a verb.

In some cases it is also possible to create the graphic model in the presence of the subject experts. The graphic model can also be designed after the information gathering session so that the details of the diagram formatting do not distract the participants.

Because many IDEF3 models may be developed at the same time by separate teams, IDEF3 supports a simple scheme for allocating activity numbers across all models. Different modelers, each assigned a different range of activity numbers, can work independently. The following table illustrates how activity IDs are allocated in large blocks to each modeler. In this example, Tom exhausted his original supply of numbers and was given a second allocation.

Modeler	IDEF3 Number Range
Tom	1-999
Lynn	1000-1999
Dan	2000-2999
Tom	3000-3999

Activity Sequencing and Concurrency

If the diagram is created after the interview, the modeler will need to make some decisions with respect to the diagram's hierarchy, for example, how much detail will be included on a single diagram page. If activity sequencing and concurrency are not clear, the experts can be interviewed again (perhaps having a copy of the incomplete diagram for reference) to fill in missing information. It is important, however, to distinguish between implied concurrency (concurrency that is implied by the absence of links) and explicit concurrency (concurrency clearly stated in the expert's description).

Activity, Junction, and Object Elaboration Documents

IDEF3 allows for information to be captured in a variety of ways. For example, complicated junction logic can be described graphically using a combination of junctions. That same information can also be captured in an elaboration referent, or even as part of the junction's definition. This allows the modeler to capture information in the form most convenient at the time (if the diagram is built in the presence of the expert). However, it is important that the models be reorganized, if necessary, to make them suitable for presentations. The choice of presentation format often has a drastic impact on the organization of a model, since elaborate junction combinations take considerable space on a diagram, and using hierarchies of junctions complicates activity placement.

IDEF3 can also be used to build design models and can be employed in this way as a follow-up activity to IDEF0 and DFD modeling.

Chapter 2: Building Process Flow Models

This section contains the following topics:

Create an IDEF3 Model (see page 31)

Create an IDEF3 Model

Create an IDEF3 model to describe and document the flow of a process.

To create an IDEF3 model

1. Select New from the File menu.

   The AllFusion Process Modeler r7 dialog opens.

2. Enter a name for the model you are creating in the Name text box, select the Process Flow (IDEF3) option for model type, and click OK.

   The Properties for New Models dialog opens.

3. Complete the following fields in the General tab of the Properties for New Models dialog, and then click OK:

   Author

   Specifies the name of the model author.

   Author initials

   Specifies the initials of the model author.

   Apply CRUD/IRUN restrictions

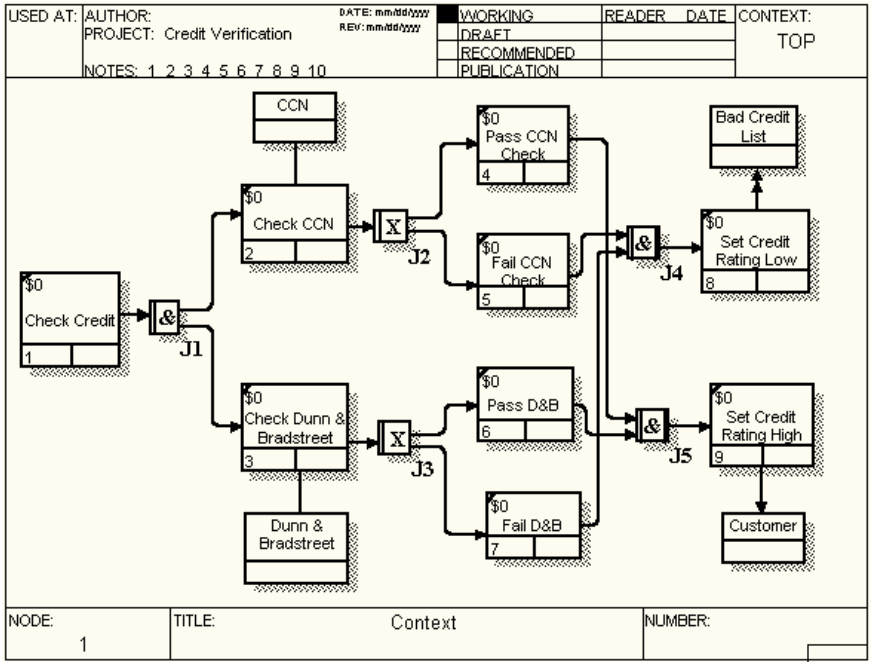
   Specifies to enforce CRUD and IRUN restrictions on Call arrows and Mechanism arrows. When you clear this check box, CRUD and IRUN restrictions are not enforced so that you can specify CRUD and IRUN data to Mechanism arrows and Call arrows.

   The model opens and displays the Activity Box that is your UOW.

   Note: To add more activities you must decompose the context activity into a new activity decomposition diagram.

Example: IDEF3 Diagram

The following is an example of a typical IDEF3 diagram after it is populated with activities, junctions, referents, and arrows.



Add More Activities to an IDEF3 Diagram

When you create an IDEF3, you also create your context activity. The next step is to decompose that activity into its related activities. If you need to add more activities to the diagram, you can do this using the Activity Box Tool in any open IDEF3 diagram.

To add activities

1. Click the Activity Box Tool then click the area on the diagram where you want to add the activity.

   The activity box is displayed on the diagram.

2. Double-click the activity box you just added.

   The Activity Properties dialog opens at the Name tab.

3. Enter a name for the activity or select an existing name from the Unused Activity Names list, and click OK.

   The Activity Properties dialog closes and the name for that activity displays in the activity box.

   Note: You can move the Activity Box within the diagram by clicking and dragging the box to the location that you want.

More information:

Create an Activity Decomposition Diagram (see page 26)

Set Activity Properties

After you create an activity box, you can set or change other activity box properties using the Activity Properties dialog.

To set activity properties

1. Double-click the activity box for which you want to add or edit properties.

   The Activity Properties dialog opens.

2. Click the one of the following tabs for the property that you want to add or edit, add or edit the values as necessary, and then click OK.

   Name

   Defines the activity name and the author name.

   Definition

   Provides the activity definition.

   Status

   Specifies the activity status option.

   Font

   Defines the activity font, font size, and font style.

   Color

   Defines the activity color and create custom color palettes.

   Costs

   Specifies the activity cost values based on cost centers defined in the Cost Center Dictionary. Frequency and duration cost values are also specified.

   UDP Values

   Specifies user-defined property (UDP) values based on UDPs created in the UDP Dictionary.

   UOW

   Defines objects, facts, description, and constraint information.

   Source

   Defines the source of the activity information.

   Roles

   Specifies role values based on the roles defined in the Role Dictionary.

   Box Style

   Defines activity box style options such as standard boxes, specific shapes, or bitmaps.

The Activity Properties dialog closes and the activity box properties you added or edited are set.

Add Junctions to an IDEF3 Diagram

After you have added all the necessary activity boxes, you need to determine which types of junctions, if any, you will need to connect your activities.

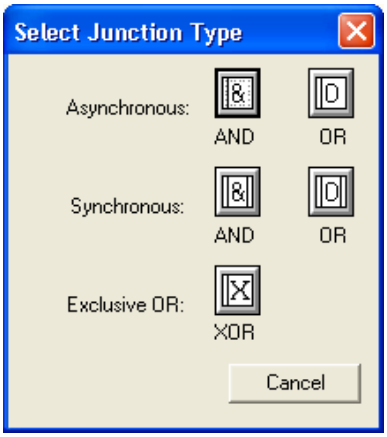
To add a junction

1. Click the Junction Tool button on the AllFusion PM toolbar.

   The Junction cursor appears.

2. Click where you want the junction to appear on the diagram.

   The Select Junction Type dialog opens.



3. Click the type of junction that you need to insert into the diagram.

   The junction appears in the diagram.

Connect Junctions to Activity Boxes Using Arrows

When you have added all the junctions required for your diagram, you must then connect the junctions to the activity boxes.

To connect a junction to an activity box

1. Select the Arrow Tool button on the AllFusion PM toolbar, and click the side of the activity box to which you want to connect the junction.

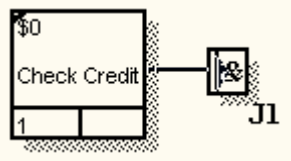
   A large highlight triangle appears.

2. Click the highlight triangle and move the cursor from the destination box to the junction that must be connected.

   A large highlight triangle appears.

3. Click the highlight triangle.

   The arrow is created.



4. Click a side of the junction box.

   A large highlight triangle appears.

5. Click the highlight triangle.

   The junction is connected to the activities that follow it.

6. Move the cursor over to the side of the destination activity.

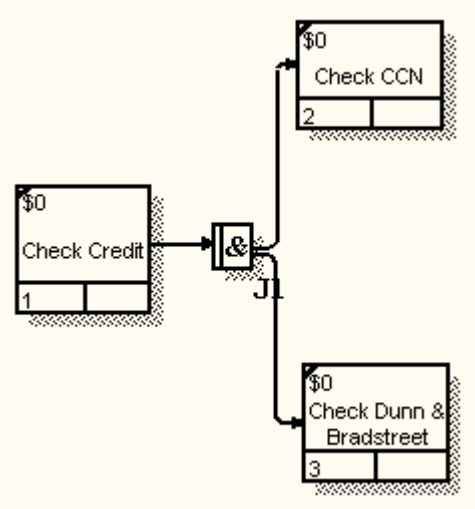
   A large highlight triangle appears.

7. Click the large highlight triangle.

   The arrow is created.

Example: Connected Junctions

The following example illustrates a typical result from the procedure to connect junctions to activity boxes using arrows.



Add Referents to an IDEF3 Diagram

Add referents to a diagram to represent the external sources of information you check in order to complete specific tasks in the process flow.

To add a referent to a diagram

1. Click the Referent Tool button  on the AllFusion PM toolbar, then click inside the diagram where you want the referent to appear.

   The Referent dialog opens.

2. Select either an existing referent name from the Reusable Referent dropdown list, or select one of the following options to assign a name to the referent, then click OK.

   Arrow

   Lets you select an existing arrow name from the list.

   Entity

   Lets you select an existing entity name from the list.

   Other

   Lets you enter a new referent name in the text box if you select this option.

   The referent appears in the diagram.

Note: You can connect referents to other objects in the IDEF3 diagram using lines instead of arrows. First draw the arrow, and then change it to a line in the Style tab in the Arrow Properties dialog.

Connect a Referent to an Activity

After you have added referents to the IDEF3 diagram, you must connect them to an activity. When you connect a referent to an activity, you should use a referent line.

To connect a referent to an activity

1. Select Default Arrow Types, Referent from the Model menu, click the Referent Arrow Tool button on the AllFusion PM toolbar, and then click inside the diagram where you want to add the source of the referent.

   A large highlight triangle appears.

2. Click the highlighted triangle, and move the cursor over the destination box that must be connected.

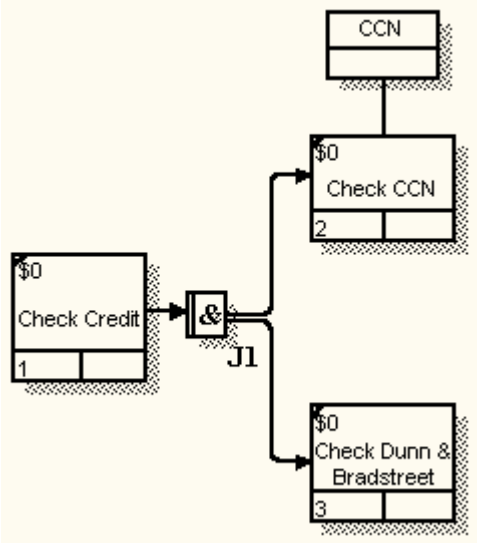
   A large highlight triangle appears.

3. Click the highlight triangle.

   The referent is connected to the activity.

Example: Connected Referents

The following example illustrates a typical result from the procedure to connect a referent to an activity.



Define the Referent Name

You can add or edit the name of a referent for your IDEF3 diagram using the Referent Properties dialog.

To define the referent name

1. Double-click the referent for which you must define a name.

   The Referent Properties dialog opens at the Name tab.

2. Select one of the following options to define the referent name and then click OK:

   Arrow

   Lets you select an existing name from the Referent Names dropdown list.

   Entity

   Lets you select an existing name from the Referent Names dropdown list.

   Other

   Lets you select an existing name from the Referent Names dropdown list or type a new referent name in the text box.

   The referent name is defined.

Note: You can modify all occurrences of a referent name in the Referent Dictionary.

Set Referent Properties

After you create a referent, you can set or change other referent properties using the Referent Properties dialog.

To set referent properties

1. Double-click the referent for which you want to add or edit properties.

   The Referent Properties dialog opens.

2. Click one of the following tabs based on the property that you want to add or edit, add or edit the values as necessary, then click OK.

   Name

   Defines the referent name and the author name.

   Box Style

   Defines referent box style options such as standard boxes, specific shapes, or bitmaps.

   Definition

   Provides the referent definition.

   Status

   Defines a referent status option.

   UDP Values

   Assigns user-defined property (UDP) values based on UDPs you create in the UDP Dictionary.

   Font

   Defines the referent font, font size, and font style.

   Color

   Defines the referent color and create custom color palettes.

   UOW

   Defines objects, facts, description, and constraint information.

   Source

   Defines the source of the referent information.

   Roles

   Assigns role values based on the roles you define in the Role Dictionary.

   The Referent Properties dialog closes and the referent properties you added or edited are set.

Chapter 3: Additional IDEF3 Features

This section contains the following topics:

Organization Visualization (see page 43)
Create an IDEF3 Scenario Diagram (see page 57)
Simulation (see page 59)
IDEF3 Process Flow Networks to Arena BE Mappings (see page 70)

Organization Visualization

Complex business processes often cut across a number of organizational boundaries and disciplines. Understanding and optimizing these types of processes requires companies to extend their thinking beyond traditional hierarchical models and to visualize operations from the perspective of their customers and partners.

With Swim Lane diagrams and organization charts you have the tools to visualize the structure and process flows of your organization. Swim Lane diagrams enable you to quickly assess and improve complex business process flows across organizational groups. Organization charts graphically help you to understand your organization's structure and its impact on your business optimization effort.

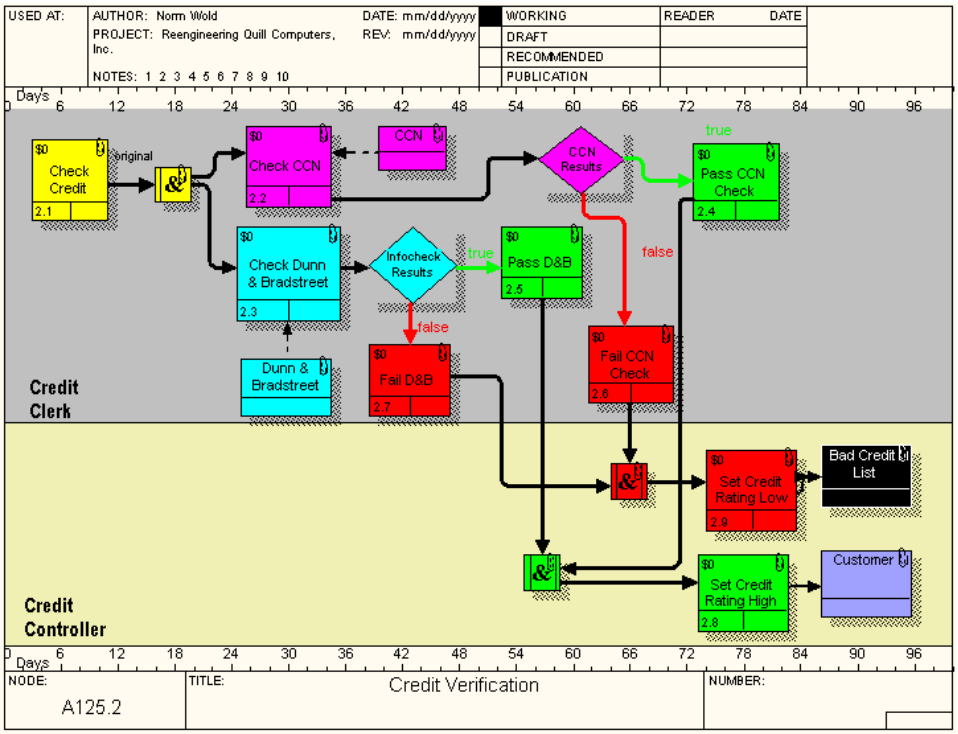
Swim Lane Diagrams

Swim Lane diagrams can provide your organization with an efficient mechanism for visualizing and optimizing processes. Swim Lane diagrams organize complex processes across functional boundaries, and help you to conveniently view processes, roles, and responsibilities, and their flow. You can build a new diagram or use one based on existing Process Flow (IDEF3) diagrams.

You can add Swim Lane diagrams to any model to better visualize process flow. Swim Lane diagrams display graphical horizontal lanes that represent process dependencies called *roles*. For example, you could create a Swim Lane diagram to display all activities with the Shipping role in the Shipping swim lane. You can also add bitmaps and a diagram scale or timeline to any Swim Lane diagram.

Example: Swim Lane Diagram

The following is an example of a Swim Lane diagram:



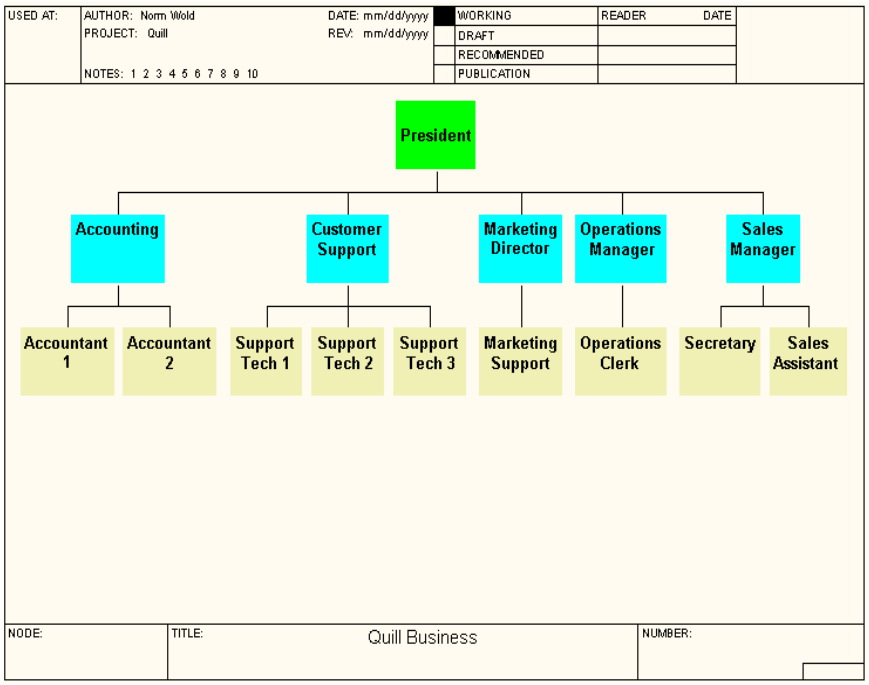
Organization Charts

Organization structures have an immense impact on how business processes are defined and carried out. Without a clear understanding of roles, relationships, and responsibilities, it is often impossible to successfully optimize business operations.

Organization charts are based on user-defined roles and provide a graphical view of an organization's structure. Use them to quickly clarify the business process optimization effort.

Example: Organization Chart

The following is an example of a typical organization chart:



How To Use Swim Lane Diagrams and Organization Charts in Your Models

To use Swim Lane diagrams or organization charts in AllFusion Process Modeler models, complete the following tasks:

- Create process roles using the following procedures:
  - Create and manage role group names and properties in the Role Group Dictionary.
  - Create and manage role names and properties in the Role Dictionary.
  - Create and manage resource names and their role associations in the Resource Dictionary.

  Note: For Swim Lane diagrams, you can create process roles in the UDP Dictionary by adding list items to a text list UDP.
- (Optional) Add bitmaps to the Bitmap Dictionary for enhancing Swim Lane diagrams and organization charts.
- Create Swim Lane diagrams or organization charts.

Create and Manage Role Groups

The Role Group Dictionary utilizes a grid to provide a way to create and manage role group names and property information. Create a new role group in the Role Group Dictionary so it can be used in Swim Lane diagrams or organization charts, or manage the properties of existing role groups.

Note: You must have at least one role group defined in the Role Group Dictionary before you can add either a role or a resource.

To create or manage a role group

1. Select Role Group from the Dictionary menu.

   The Role Group Dictionary opens.

2. Add or edit the information in the following Role Group Dictionary columns, select Save from the Dictionary menu, and then select Close from the Dictionary menu.

   Name

   Displays the role group name.

   Click the cell in the last row and enter the role group name, or select a name to edit.

   Note: Use a role group name that best describes how you intend to use the role group. For example, if you want to create a company organizational chart, you can use "WIDGET INC. MANAGEMENT" as a role group.

   Definition

   Displays the role group definition information.

   Bitmap

   Displays the default bitmap associated with the role group.

   Importance

   Displays the importance level of the role group.

   Shape

   Displays the default shape for the roles associated with the role group.

   The role group is added to the Role Group Dictionary and the Role Group Dictionary closes.

Create and Manage Roles

The Role Dictionary utilizes a grid to provide a way to create and manage role group names and property information. Create a new role in the Role Dictionary so it can be used in Swim Lane diagrams or organization charts, or manage the properties of existing roles.

Note: Before you can add a new role, you must have at least one role group defined in the Role Group Dictionary.

To create or manage a role

1. Select Role from the Dictionary menu.

   The Role Dictionary opens.

2. Add or edit the information in the following Role Dictionary columns, select Save from the Dictionary menu, and then select Close from the Dictionary menu.

   Name

   Displays the role name.

   Click the cell in the last row and enter the role name, or select a name to edit.

   Definition

   Displays the role definition information.

   Role Group

   Displays the role group associated with the role.

   Bitmap

   Displays the default bitmap associated with the role.

   Importance

   Displays the importance level of the role.

   Shape

   Displays the default shape associated with the role.

   The role is added to the Role Dictionary and the Role Dictionary closes.

Create and Manage Resources

The Resource Dictionary utilizes a grid to provide a way to create and manage resource names and their associations to roles. Create a new resource in the Resource Dictionary so it can be used in Swim Lane diagrams or organization charts, or manage the properties of existing resources.

Note: Before you can add a new resource, you must have at least one role group defined in the Role Group Dictionary, and roles must be defined in the Role Dictionary.

To create or manage a resource

1. Select Resource from the Dictionary menu.

   The Resource Dictionary opens.

2. Add or edit the information in the following Resource Dictionary columns, select Save from the Dictionary menu, and then select Close from the Dictionary menu.

   Name

   Displays the resource name.

   Click the cell in the last row and enter the resource name, or select a name to edit.

   Definition

   Displays the resource definition information.

   Associations

   Displays the role group/role associations.

   The resource is added to the Resource Dictionary and the Resource Dictionary closes.

Create a Process Role in the UDP Dictionary

Use the UDP Dictionary to add a UDP list item to represent a process role, so that it can be used in a Swim Lane diagram. These list items appear as selections in the UDP list boxes in diagram object dictionaries.

To add a process role to a UDP Dictionary

1. Select UDP from the Dictionary menu.

   The UDP Dictionary opens.

2. Click a cell in the UDP Datatype column, select Text List (Single selection) from the dropdown selection, and then click a cell in the Settings column.

   The Text UDP List Editor opens.

3. Type the UDP list item name on a new line in the list area and click OK.

   The process role is added and the Text UDP List Editor closes.

Add a Bitmap to the Bitmap Dictionary

You can use bitmaps (*.bmp) to enhance the appearance of any Swim Lane diagram or organization chart. To use bitmaps in diagrams, you must add them to the Bitmap Dictionary by importing them from an external source, such as your computer hard drive.

Note: You can import any size bitmap. After you import a bitmap, the original bitmap file is not referenced or required. When you save the model, all imported bitmaps are saved with the model file.

To add a bitmap to the Bitmap Dictionary

1. Select Bitmaps from the Dictionary menu.

   The Bitmap Dictionary opens.

2. Click the Import button.

   The Open dialog opens.

3. Select the path and file name of the bitmap (*.bmp) you want to add and click Open.

   The Open dialog closes, you are returned to the Bitmap Dictionary, and your selected bitmap is displayed in the Model bitmaps list.

4. Click OK.

   The bitmap is added and the Bitmap Dictionary closes.

Create a Swim Lane Diagram

Add a Swim Lane diagram to any IDEF3 model to display all activities with a specific role and better visualize process flow. Once you have created the necessary role groups, roles, and resources, you are ready to create a Swim Lane diagram.

Note: You must have either a role group or a single-selection text list UDP defined in order to create a Swim Lane diagram.

To create a Swim Lane diagram

1. Select Add Swim Lane diagram from the Diagram menu.

   The Swim Lane Diagram Wizard - Step 1 of 2 dialog opens.

2. Complete the following information in this dialog, and then click the Next button:

   Role Group

   Bases the Swim Lane diagram on a role group and lets you select the specific role group from the dropdown list.

   Text List UDP

   Bases the Swim Lane diagram on a text list UDP and lets you select the specific text list UDP from the dropdown list.

   Yes

   Specifies to copy the objects from an IDEF3 diagram into the Swim Lane diagram and lets you select the specific IDEF3 diagram from the dropdown list.

   No

   Specifies not to copy the objects from an IDEF3 diagram into the Swim Lane diagram.

   If you select this option, you can add diagram objects such as activities and arrows later.

   Diagram name

   Specifies a name for the Swim Lane diagram.

   The Swim Lane Diagram Wizard - Step 2 of 2 dialog opens.

3. Complete the following information in this dialog, and then click the Finish button:

   Swim Lanes on Diagram

   Specifies the swim lanes to display in the Swim Lane diagram.

   Swim Lanes NOT on Diagram

   Specifies the swim lanes to not display in the Swim Lane diagram.

Add All

Specifies to move all the displayed swim lanes into the Swim Lanes on Diagram list.

Remove All

Specifies to move all the displayed swim lanes into the Swim Lanes NOT on Diagram list.

Bitmap display

Lets you specify each swim lane that you want to display as a bitmap.

Note: Swim Lane bitmaps apply only to Swim Lane diagrams based on a Role Group. Bitmaps do not apply to Swim Lane diagrams that you base on a text list UDP.

The Swim Lane diagram is added to the model using the diagram properties you defined in this wizard, and is opened in the diagram workspace.

Note: You can also drag and drop swim lanes between the Swim Lanes on Diagram tree and the Swim Lanes NOT on Diagram tree, and place them in any order that you need.

Movement of Objects in a Swim Lane Diagram

You can move objects in a Swim Lane diagram by clicking and dragging objects to the location that you want. You can change the order of swim lanes on your diagram from the Display tab of the Diagram Properties dialog.

You can also resize objects when you move the cursor over the object until you see a double-headed arrow, and then click and drag to resize.

Create an Organization Chart

Add an organization chart to any model to provide a graphical view of an organization's structure. You must first create the necessary role groups, roles, and resources in order to create an organization chart.

To create an organization chart

1. Click Add Organization Chart from the Diagram menu.

   The Organization Chart Wizard - Step 1 of 3 dialog opens.

2. Complete the following information in this dialog, and then click the Next button:

   Name

   Specifies the name of the new organization chart.

   Role Group

   Specifies the role group.

   Click the Role Group Dictionary button to either add or edit a role group.

   Note: When you open the Role Group Dictionary, the Organization Chart Wizard closes.

   Role

   Specifies the role.

   Click the Role Dictionary to add or edit a role.

   Note: When you open the Role Dictionary, the Organization Chart Wizard closes.

   Resource

   Specifies the resource.

   Click the Resource Dictionary button to add or edit a resource.

   Note: When you open the Resource Dictionary, the Organization Chart Wizard closes.

   Author

   (Optional) Specifies the organization chart author.

   The Organization Chart Wizard - Step 2 of 3 opens.

3. Complete the following information in the dialog:

   Role Group for the next level

   Specifies the role group containing the roles for the second level of the organization chart.

   Available Role / Resources

   Specifies the role and resource combinations available for selection.

   Click the role and resource combinations you require, and click Add to move it into the Selected Group/Role/Resources list.

   Selected Group/Role/Resources

   Specifies all the role and resource combinations that you moved from the Available Role / Resources list.

   Select a role and resource combination in this list and click Remove to move the role and resource combination back into the Available Role / Resources list. To change the order of this list use the up and down arrow buttons.

   Click Next.

   The Organization Chart Wizard - Step 3 of 3 opens.

4. Select the following organization chart display and style options as needed for your environment, and then click Finish:

   Drawing

   Specifies the following drawing options:

   Show boxes

   Specifies to display roles as graphical boxes in the organization chart.

   Show Role Group Name

   Specifies to display the role group name in each organization chart box.

   Show Role Name

   Specifies to display the role name in each organization chart box.

   Show Resource Name

   Specifies to display the resource name in each organization chart box.

   Border

   Specifies the following border options:

   Include Kit

   Specifies to include the diagram kit in the organization chart.

   Include title

   Specifies to include the diagram title area in the organization chart.

Box Size

Specifies the following box size options:

Fit each box to text

Enables automatic adjustment of each box to the amount of text in the box.

One size per row

Enables automatic adjustment of each box within any row to be the same size.

The box with the most amount of text is used to size all boxes in a row.

All one size

Enables automatic adjustment of each box in the organization chart to be the same size.

The box with the most amount of text is used to size each box in the organization chart.

Draw Style

Specifies the following draw style options:

Standard

Enables boxes in the organization chart to be drawn according to the options selected in the Drawing group box.

Bitmap

Enables role bitmaps in the organization chart boxes to be drawn.

You can view bitmaps in the organization chart only if you first assign bitmaps to roles in the Role Dictionary.

Shape

Enables role shapes in the organization chart boxes to be drawn.

You can view shapes in the organization chart only if you first assign shapes to roles in the Role Dictionary.

Your organization chart opens in the diagram workspace.

Note: You can change any of these options later in the Style tab in the Organization Chart Properties dialog.

Move Organization Chart Objects

You can move objects in an organization chart by clicking and dragging objects right on the diagram workspace.

To move objects in an organization chart

1. Click Open from the File menu and select the model that contains the organization chart that you need to edit.

   The model opens in the diagram workspace.

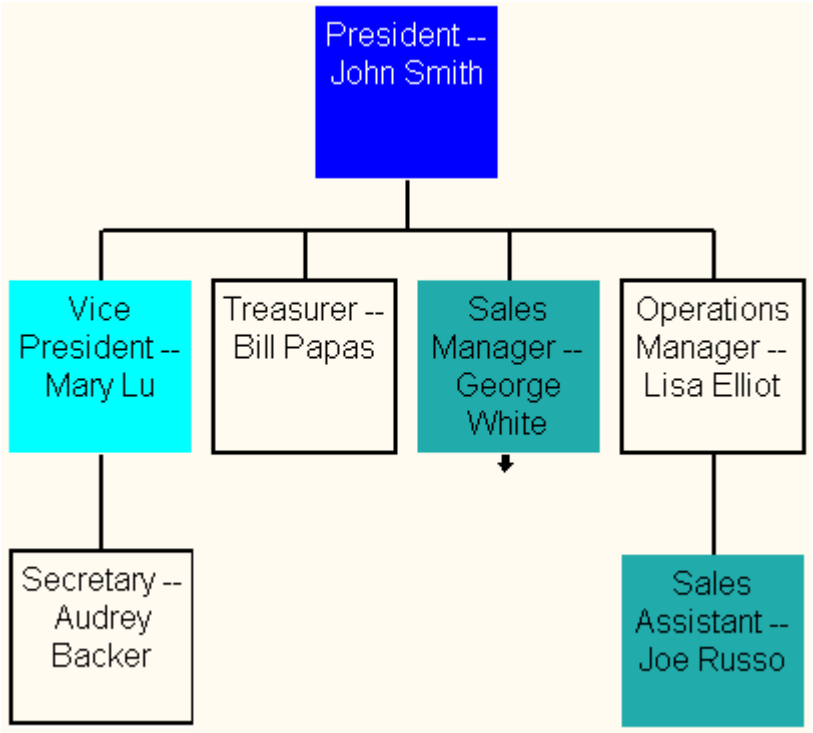
2. Open the organization chart that you need to edit, single-click each object, drag it to its new location in the organization chart, and then release the mouse button.

   Note: The cursor changes to a thick black arrow to show you when you can release the mouse button.

   The object appears in its new location.

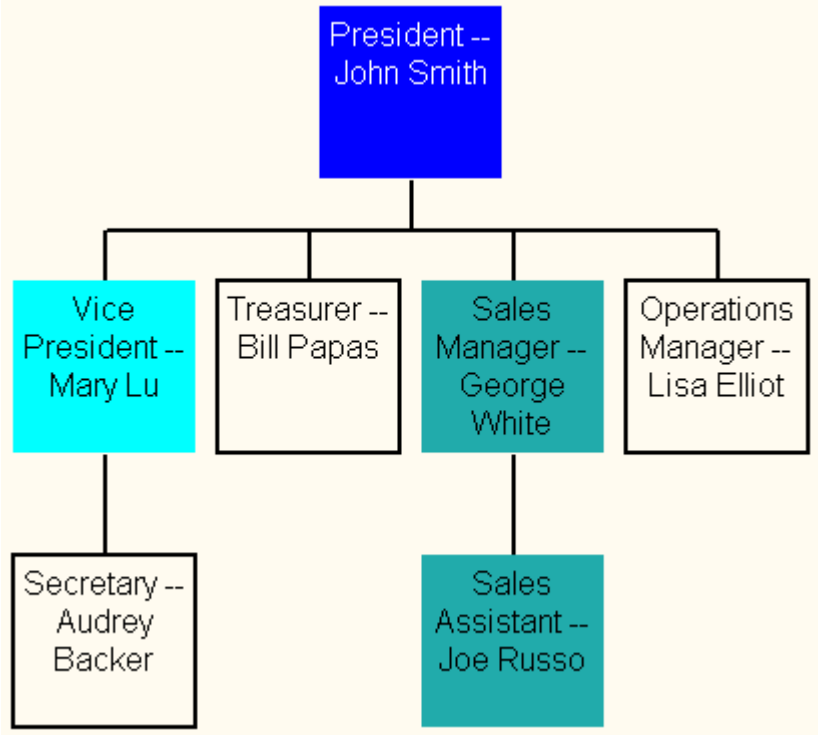
Example: Organization Chart Object Movement

In the following illustration, the organization chart object *Sales Assistant--Joe Russo* needs to be moved from *Operations Manager--Lisa Elliot* to *Sales Manager--George White*:



Example: Organization Chart Object Movement Completion

In the following illustration, the movement of the organization chart object *Sales Assistant--Joe Russo* from *Operations Manager--Lisa Elliot* to *Sales Manager--George White* was completed:



Create an IDEF3 Scenario Diagram

Create an IDEF3 Scenario diagram to capture many different viewpoints of a process in IDEF3 scenarios and then make a composite as the final standard diagram or use them as the basis for the IDEF0 view of the process.

To create an IDEF3 Scenario diagram

1. Click Add IDEF3 Scenario from the Diagram menu.

   The Add New IDEF3 Scenario diagram dialog opens.

2. Complete the following fields:

Name of new diagram

Specifies the name of the IDEF3 Scenario diagram.

IDEF3 Scenario Of

Specifies the type of diagram upon which to create the new FEO diagram.

Select one of the following options:

Context Diagram

Specifies to use the model context diagram.

Select a context diagram from the Source Diagram Name list.

Decomposition Diagram

Specifies to use a decomposition diagram.

Select a decomposition diagram from the Source Diagram Name list.

Source Diagram Name

Specifies the source context or decomposition diagram upon which to create the IDEF3 Scenario diagram.

Copy contents of source diagram

Specifies to copy all contents of the source diagram into the new IDEF3 Scenario diagram.

Click OK.

The Add New IDEF3 Scenario diagram dialog closes and the IDEF3 Scenario diagram displays in the diagram area.

Simulation

Simulation is a modeling technique used to study changes in the system that occur as a function of time. As a simulation advances with time, pertinent statistics are gathered about the simulated system in much the same way as it is performed in real life.

When activity models are simulated, statistically relevant changes are associated with events. Simulation is performed, in effect, by jumping forward in time from one event to the next. This type of simulation is commonly known as *discrete event simulation*.

Simulation is usually associated with *operations research*, a decision science that attempts to determine the best course of action with limited resources. However, the objectives and constraints of many real-world systems are difficult to express either quantitatively or mathematically. Alternatively, simulation models the system under study as a collection of elemental modules linked by well-defined logical relationships rather than as a (usually complex) mathematical formula. Compared to mathematical models, simulation models usually provide greater flexibility in defining objectives and constraints.

Simulation modeling does have two notable drawbacks:

- Minute details can have significant impact on the outcome, which necessitates a rather expensive and detailed modeling effort.
- Simulations can execute for considerable periods of time even on high-performance computer systems.

The relationship between simulation models and activity models is synergistic. Activity models can be translated into a skeleton, or incomplete, simulation model, and simulation models greatly increase our understanding of the performance of some system, perhaps resulting in changes that are made to the original activity models.

A simulation model includes the following major components:

- Sources and destinations
- Queues
- Facilities

Sources and Destinations in a Simulation Model

Sources model the arrival of inputs and are similar in concept to external entities in DFDs. The rate of time between arrivals is recorded as a mathematical expression, usually a statistical distribution function such as *normal (mean, standard deviation)*.

Destinations are also like external entities but in simulation, their role is generally that of an information-gathering device. Destinations capture information related to the journey of an object through the system, which can include the total time to transform the object from input to output, the total cost to produce the output, and so on.

Queues in a Simulation Model

Queues in a simulation model are similar to data stores in DFDs. A *queue* is a way to model the characteristics of one or more objects waiting to be processed. Activity cycle time or *duration* is the time it takes an activity to execute. Activity duration may vary from one iteration to the next, causing fluctuations in output. For example, consider two activities that normally take about the same length of time. Occasionally an earlier activity executes faster than the later activity, which may cause partially finished goods to pile up in the queue.

Queue behavior must be encoded in the simulation. For example, a queue can function as a stack: the last item placed on the stack is the first item that is retrieved. This is known as last-in-first-out (LIFO). Alternatively, a waiting line generally operates on a first-in-first-out (FIFO) basis. A queue can also operate randomly, or according to some other formula.

When IDEF0 and IDEF3 models are simulated, a queue for each input and control is assumed.

Facilities in a Simulation Model

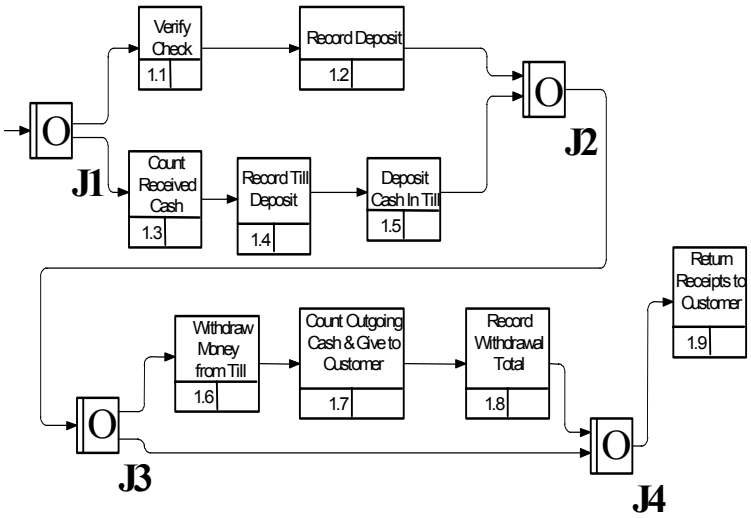
Facilities model the activities of the system. The term *facility* reflects the manufacturing roots of simulation in which an activity is performed at some work station (usually a dedicated piece of machinery) and the materials move from one station to the next in assembly line fashion. The processing time is recorded as a mathematical (usually statistical) expression, and rules for the use of resources (mechanisms) are specified.

Simulation Examples

The following simulation examples describe the various activities that occur in a bank lobby. Bank tellers must perform a variety of transactions, each one requiring the teller to perform a different set of activities. When you use simulation, you can understand how to best allocate bank tellers.

Example: Deposit into Customer Account

This simulation model illustrates the activity Deposit into Customer Account:

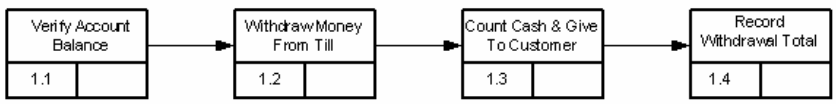


The following is an outline describing the scenarios involved in the previous illustration:

- Deposit check:
  - Verify check and record deposit.
- Deposit cash:
  - Count the cash, record till deposit, and deposit the cash in till. Use the cash counter if a large amount of cash is being deposited.
- Return cash to customer:
  - Withdraw money from the till, count it, and give to customer. Record the withdrawal total.
- Return receipts to customer.

Example: Withdrawal from Customer Account

This simulation model illustrates the activity Withdrawal from Customer Account:

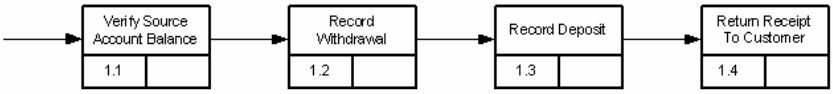


The following is an outline describing the scenarios involved in the previous illustration:

- Verify account balance.
- Withdraw money from the till, count it, and give it to the customer.
- Record the withdrawal total.

Example: Transfer among Customer Accounts

This simulation model illustrates the activity Transfer among Customer Accounts:

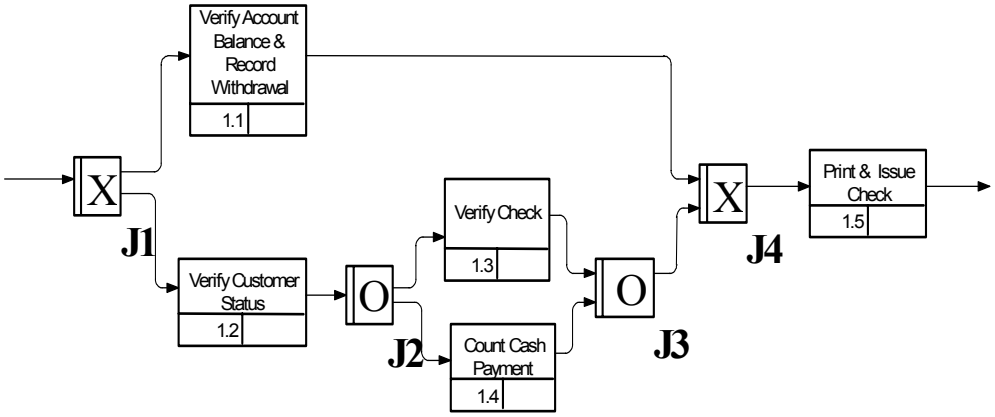


The following is an outline describing the scenarios involved in the previous illustration:

- Verify source account balance.
- Record withdrawal.
- Record deposit.
- Return receipt to the customer.

Example: Issue Cashier's Check

This simulation model illustrates the activity Issue Cashier's Check:



The following is an outline describing the scenarios involved in the previous illustration:

- Withdrawal:
  - Verify account balance and record withdrawal.
- Cash transaction:
  - Verify customer status, verify checks, and count cash.
- Print and issue check.

In the previous illustration, there are two general transaction types based on how the funds are supplied for a cashier's check, either a withdrawal of money from an account or with cash. As shown by the OR junction, funds can be supplied using cash, check, or both. If there are significant differences between the three types of cash transactions in the amount of time and cost to perform them, then it is important to model all cases for the simulation to be accurate. Likewise, if transaction times for business accounts vary considerably from personal accounts, this also needs to be incorporated into the model.

Simulation Formulas

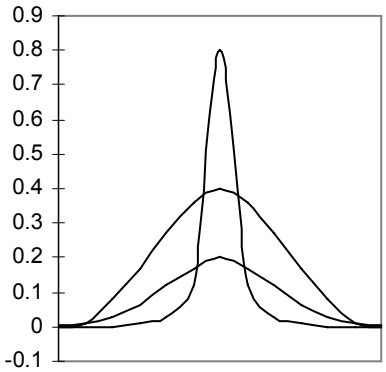
Instead of using a simulation model for activities, you can use a formula. It is important to remember that it is not always necessary to model all activities to the same level of detail, but to create a model in sufficient detail to fulfill the model's purpose.

The important simulation variables are described in the following table, along with the necessary resources (IDEF0 mechanisms) to perform the activities:

Resource Name	Number Available	Cost per Minute	MTBF	Downtime
Teller	3	$0.20	0	0
Cash Counting Machine	1	$0.05	Normal (6000,25)	Normal (16,3)

In the previous table, Mean Time Between Failure (MTBF) is used to record situations where something is periodically unavailable due to failure. Downtime measures how long the resource is typically out of commission. If we needed a more accurate model of the availability of individual tellers, such as would be required to simulate several days worth of activity, then a resource calendar can be used to document the availability schedule.

In statistics, certain formulas are developed to describe the probability of some phenomenon. For example, on average, how many customers enter a particular bank lobby between 12:00 PM and 1:00 PM on Friday? The average, called the *mean,* is represented on the three distributions in the following illustration by the peak of each curve. The margin of error, that is, how close together the values of all the samples are, is called the standard deviation, and is represented by the width of each curve. There are many probability curves other than the normal distribution that together describe a wide variety of situations. Graphically, the shapes of these other probability curves will vary substantially from that shown in the following illustration.



These probability formulas are at the heart of simulation. In addition to representing the range of execution times of activities, statistical probability formulas are also used to represent the range of time between arrivals and the availability of resources. When the simulation engine is run, the software that actually executes the simulation continuously uses these formulas to determine activity execution times, arrival rates, and resource availability.

As the following table shows, the time to perform each activity is represented as a statistical probability formula. For example, the time to perform Verify Check and Record Deposit is distributed normally and takes an average of .5 minute to perform. This is the first number in the formula. It does not always take .5 minute to complete this activity, this is just the average time. This transaction is completed within the range of .4 minute and .6 minute, except on rare occasions. This is calculated by taking the second number in the formula, the standard deviation, and adding it to the mean to get .6, and subtracting it from the mean to get .4.

Note: In the following table, CCM = Cash-counting Machine.

Activity Name	Resource Name	Resource Quantity	Time
Deposit into Customer Account			
Verify Check and Record Deposit	Teller	1	Normal(.5,.1)
Count Small Cash Deposit	Teller	1	Normal(.5,.5)
Count Large Cash Deposit	Teller	1	Normal(1,.5)
	CCM	1	
Record Till Deposit and Deposit Cash in Till	Teller	1	Normal(1,.5)
Withdraw Money from Till	Teller	1	Normal(.3,.1)
Count Small Cash Return	Teller	1	Normal(.5,.5)
Count Large Cash Return	Teller	1	Normal(1,.5)
	CCM	1	
Give to Customer	Teller	1	Normal(.3,.1)
Record Withdrawal Total	Teller	1	Normal(.3,.1)
Return Receipt to Customer	Teller	1	Normal(.3,.1)
Withdrawal from Customer Account			
Verify Account Balance	Teller	1	Normal(.5,.3)
Withdraw Money from Till	Teller	1	Normal(.3,.1)
Count Small Cash Return	Teller	1	Normal(.5,.5)

Activity Name	Resource Name	Resource Quantity	Time
Count Large Cash Return	Teller	1	Normal(1,.5)
	CCM	1	
Give to Customer	Teller	1	Normal(.3,.1)
Record Withdrawal Total	Teller	1	Normal(.3,.1)
Transfer among Customer Accounts			
Verify Source Account Balance	Teller	1	Normal(.5,.3)
Record Withdrawal	Teller	1	Normal(.3,.1)
Record Deposit	Teller	1	Normal(.3,.1)
Return Receipt to Customer	Teller	1	Normal(.3,.1)
Issue Cashier's Check			
Verify Account Balance	Teller	1	Normal(.5,.3)
Record Withdrawal	Teller	1	Normal(.3,.1)
Verify Customer Status	Teller	1	Normal(.3,.1)
Verify Checks	Teller	1	Normal(1,.5)
Count Cash	Teller	1	Normal(.5,.5)
Print and Issue Check	Teller	1	Normal(1,.5)
Open New Account	Teller	1	Normal(15,6)

Arrivals (sources) can be modeled using either of the following two methods:

- A separate arrival node for each customer type.
- One arrival node with the different activity types represented as probabilities.

Each representation has advantages with respect to complexity and accuracy, however, in the following table, the second method of modeling arrivals is described. Arrivals are represented as a single node with probabilities for each transaction type.

First, the arrival rate is defined as Normal (1.2, 0.3). This means that some customer arrives on the average of once every 1.2 minutes. If the arrival rate varies over time, you can create a table of arrival rates, each valid for a certain time period, similar to how the availability of resources is defined.

According to the following table, 40% (.4) of all transactions are deposits. Of these, half are check-only deposits, 30% are small deposits, and 20% are large deposits. Similar values indicate the probabilities of the other activity types.

General Activity Type	Special Type	Probability
Deposit into Customer Account		**.40**
	Check Only	.50
	Small Cash Deposit	.30
	Large Cash Deposit	.20
Withdraw from Customer Account		**.25**
	Small Cash Return	.75
	Large Cash Return	.25
Transfer among Customer Accounts		**.15**
Issue Cashier's Check		**.15**
	Withdrawal	.80
	Cash	.20
Create New Accounts		**.05**

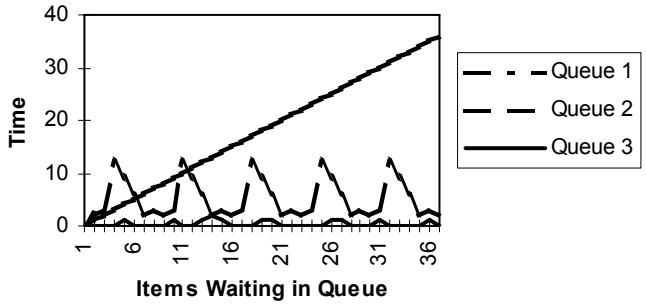
To define queue behavior, the modeler must make some choices about the physical implementation. For example, each General Activity Type can have its own queue, or you can use only one queue.

In the following table that describes queue behavior, the environment is one queue with unlimited capacity. However, it might be useful to put limits on the queue and see if any customers are lost because the queue is too long (that is, if the line looks too long, customers will not wait).

Queue Name	Behavior	Capacity
Incoming Queue	FIFO	0

Finally, we need to identify what information we want to capture about each of the important components. The choice of output variables to monitor depends on the purpose for building the model. For our model, the most obvious measures are the wait times for customers in line and the queue length. However, simply recording the mean and standard deviations is not good enough. We need to see how the queue size and wait time vary over time.

The process that consumes queue 1 in the following illustration is not keeping up; the queue gets longer and longer. The size of queue 2 varies within a range. If the range is not acceptable, that is, the queue is getting too long at certain times, further analysis is needed to understand why. Perhaps a better resource allocation scheme is required to keep up with peak periods. Queue 3 and its activities is a candidate for elimination if the supplier and consumer can coordinate their schedules better.



In this case, resource costs are also important because the smallest number of tellers and cash-counting machines that can handle the workload must be identified. The amount of time that each of the resources was idle must be understood in order to calculate the idle cost (cost for the resource to do nothing), as well as the amount of time the teller was occupied performing each transaction type. The amount of time a teller was waiting for a cash-counting machine can also be captured. All of this information can be captured by developing formulas in the simulation to compute the results.

Simulation Experiments

A simulation is not run just once. The same model may be simulated many times in order to obtain a more accurate representation of performance. In addition, numerous experiments are run on the same basic model, each time manipulating one variable.

The following describes a typical simulation experiment, and is also displayed in the following table.

- Begin with the smallest number of tellers and run a simulation experiment.
- Increase the number of tellers by one.
- Increase the number of cash-counting machines and run another simulation experiment to explore the effect.
- Repeat the process, increasing the number of tellers and the number of cash-counting machines until a simulation experiment is run with the same amount of tellers and cash-counting machines (in the following table, this is four).

Experiment	Tellers	Cash-counting Machines
1	1	1
2	2	1
3	2	2
4	3	1
5	3	2
6	3	3
7	4	1
8	4	2
9	4	3
10	4	4

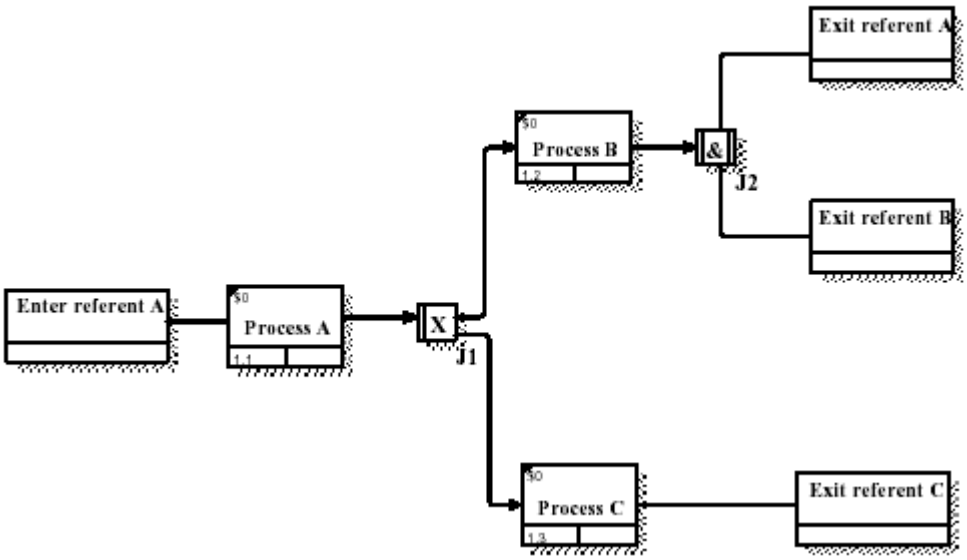
IDEF3 Process Flow Networks to Arena BE Mappings

This section describes the potential content of an AllFusion PM IDEF3 flow network and how this may map to a Rockwell Software, Inc. Arena® BE model.

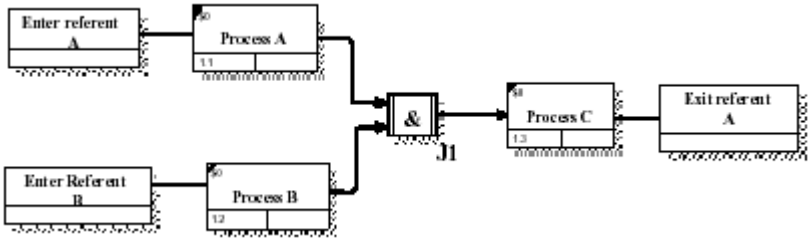
The following items are important points to note:

- The interface maps a single IDEF3 diagram to a single Arena BE model (hierarchy is not supported).
- Not all objects in an IDEF3 diagram will necessarily map to an Arena BE model.
- Not all objects in an Arena BE model can be modeled in an IDEF3 model.

The objective is to include what makes best sense in terms of both tools. The following diagram shows the basic set of IDEF3 objects that may be exchanged:



The following diagram shows a fan-in to a junction. In an actual model, a mixture of fan-in and fan-out situations can occur. Arena BE has specific objects that handle these situations. **Note:** For more information on the specific objects, see the mapping tables in the following sections.



Base Mapping Table

The following table describes the base mapping for AllFusion PM and Arena BE objects:

AllFusion PM Object	Arena BE Object	Comments
Referent (Link From)	Create Module	AllFusion PM Link (arrow) style is modified so that the arrow head can display
Referent (Link To)	Dispose Module	AllFusion PM Link (arrow) style is modified so that the arrow head can display
UOB (IDEF3 Activity)	Process Module	None
Junction (XOR or Sync OR)	Decide Module	Only allows two outgoing links (all others are rejected by the interface) but greater than two incoming
Junction (AND or Async OR) where links (arrows) fan-in	Batch Module	Can be considered a special activity if needed by using UDPs to designate
Junction (AND or Async OR) where links (arrows) fan-out	Separate Module	Only allows two outgoing links (all others are rejected by the interface)
Any diagram object dictionary object	Resource Module	Dictionary objects are used only to store resource rows and there is no association between the data and the object on which it is stored
Arrows (inputs to UOBs representing Process Modules)	Resource or Set Repeat Group	Data is associated with the Process Module into which the arrow sinks; sets are not supported
Not required; see comments	Simulate Module	Created by the interface

No direct equivalent	Assign Module	Can be considered a special activity if needed by using UDPs to designate
No direct equivalent	Record Module	No equivalent possible

The methodology of UDPs attached to the appropriate object is used to set parameters in the AllFusion PM model before interchange. The following sections define the requirements by object type.

Arena Create Module and AllFusion PM Referent (Link From)

The following table describes the requirements for the Arena Create Module and the AllFusion PM Referent (Link From):

Prompt Name	**Description**	**AllFusion PM Meta Property**	**UDP Name**	**UDP Datatype**	**UDP Value List**
Name	Unique name	Referent Name	Not applicable	Not applicable	Not applicable
Entity type	Name of the entity type that will be generated	UDP	Arena_C_ EntityType	Text	Not applicable
Type	Arrival stream type	UDP	Arena_C_ ArrivalType	SS Text List	▪ Constant ▪ Expression ▪ Random ▪ Schedule
Value	Mean of exponent when Random or Constant	UDP	Arena_C_Value	Real	Not applicable
Schedule Name	Name of schedule if type schedule	UDP	Arena_C_ ScheduleName	Text	Not applicable
Expression	Distribution or value for time between arrivals	UDP	Arena_C_ Expression	Text	Not applicable
Units	Type of time units	UDP	Arena_C_Units	SS Text List	▪ Days ▪ Hours ▪ Minutes ▪ Seconds

Entities per arrival	Number of entities entering at a given time	UDP	Arena_C_ EntitiesPerArrival	Real	Not applicable
Max Arrivals	Maximum number of entries that this module will initiate	UDP	Arena_C_ MaxArrivals	Text	Not applicable
First Creation	Starting time for first arrival	UDP	Arena_C_ FirstCreation	Text	Not applicable

Arena Process Module and AllFusion PM UOB (IDEF3 Activity)

The following table describes the requirements for the Arena Process Module and the AllFusion PM UOB (IDEF3 Activity):

Prompt Name	Description	AllFusion PM Meta Property	UDP Name	UDP Datatype	UDP Value List
Name	Unique name	Activity Name	Not applicable	Not applicable	Not applicable
Type	Method of logic	UDP	Arena_P_Type	SS Text List	▪ Standard ▪ Submodel
Action	Processing type	UDP	Arena_P_ Action	SS Text List	▪ Delay ▪ Delay Release ▪ Seize Delay ▪ Seize Delay Release
Priority	Priority value	UDP	Arena_P_ Priority	Text	Not applicable
Resources	Repeat group of resources	UDP Version 2.5	Arena_P_ Resources	MS Text List	Set by user
Delay Type	Distribution type for delay parameters	UDP	Arena_P_ DelayType	SS Text List	▪ Constant ▪ Expression ▪ Normal ▪ Triangular ▪ Uniform

Units	Time units for delay parameters	UDP	Arena_P_ Units	SS Text List	▪ Days ▪ Hours ▪ Minutes ▪ Seconds
Allocation	How processing time is allocated	UDP	Arena_P_ Allocation	SS Text List	▪ Non-value added ▪ Value added
Minimum	Minimum value for Uniform or Triangular distribution	UDP	Arena_P_ Minimum	Text	Not applicable
Value	Mean for Normal, Value for Constant, Mode for Triangular	UDP	Arena_P_ Value	Text	Not applicable
Maximum	Maximum value for Uniform or Triangular	UDP	Arena_P_ Maximum	Text	Not applicable
Std Dev	Std Dev for Normal distribution	UDP	Arena_P_ Std Dev	Text	Not applicable
Expression	Value of expression type	UDP	Arena_P_ Expression	Text	Not applicable

Arena Decide Module and AllFusion PM Junction (XOR or Sync OR)

The following table describes the requirements for the Arena Decide Module and the AllFusion PM Junction (XOR or Sync OR):

Prompt Name	**Description**	**AllFusion PM Meta Property**	**UDP Name**	**UDP Datatype**	**UDP Value List**
Name	Unique name	Junction Name	Not applicable	Not applicable	Not applicable
Type	Type of decision	UDP	Arena_?_Type	SS Text List	▪ By chance ▪ By condition

Percent True	Value percent to be checked and sent down to the True/False exit points	UDP	Arena_?_PercentTrue	Text	Not applicable
If	Types of conditions available for evaluation	UDP	Arena_?_ ConditionType	SS Text List	▪ Attribute ▪ Entity Type ▪ Expression ▪ Variable
VNamed	Name of variable used in conditional statement	UDP Version 2.5	Arena_?_ VariableName	Text	Not applicable
ANamed	Name of attribute used in conditional statement	UDP	Arena_?_ AttributeName	Text	Not applicable
TypeNamed	Name of Entity Type used in conditional statement	UDP	Arena_?_ EntityName	Text	Not applicable
Is	Evaluator	UDP	Arena_?_ ConditionEvaluator	SS Text List	▪ < ▪ <= ▪ <> ▪ == ▪ > ▪ >=
Value	Expression used for condition	UDP	Arena_?_ ConditionValue	Text	Not applicable

Note: For more information on this table, see the conditions in the Base Mapping Table.

Arena Batch Module and AllFusion PM Junction (AND or Async OR)

The junctions AND or Async OR are where the links (arrows) fan-in. The following table describes the requirements for the Arena Batch Module and the AllFusion PM Junction (AND or Async OR):

Prompt Name	Description	AllFusion PM Meta Property	UDP Name	UDP Datatype	UDP Value List
Name	Unique name	Junction Name	Not applicable	Not applicable	Not applicable
Type	Batch type	UDP	Arena_B_Type	SS Text List	▪ Permanent ▪ Temporary
Batch size	Number of entities	UDP	Arena_B_ BatchSize	Text	Not applicable
Rule	How batched	UDP	Arena_B_Rule	SS Text List	▪ Any entity ▪ By attribute
Attribute name	Name if Rule set to By attribute	UDP	Arena_B_ AttributeName	Text	Not applicable

Arena Separate Module and AllFusion PM Junction (AND or Async OR)

The junctions AND or Async OR are where the links (arrows) fan-out. The following table describes the requirements for the Arena Separate Module and the AllFusion PM Junction (AND or Async OR):

Prompt Name	Description	AllFusion PM Meta Property	UDP Name	UDP Datatype	UDP Value List
Name	Unique name	Junction Name	Not applicable	Not applicable	Not applicable
Type	Separation method type	UDP	Arena_S_Type	SS Text List	▪ Duplicate original ▪ Split existing batch
Percent cost to duplicates	Percent of entities' original cost	UDP	Arena_S_ PercentCostToDuplicates	Text	Not applicable

Number of duplicates	Number of outgoing that will leave in addition	UDP	Arena_S_#ofDuplicates	Text	Not applicable
Allocation Rule	Method of cost and allocation time	UDP	Arena_S_AllocationRule	SS Text List	▪ Do not split costs and times ▪ Split all costs and times ▪ Split only new costs and times

Note: For more information on this table, see the conditions in the Base Mapping Table.

Arena Assign Module and AllFusion PM UOB

The following table describes the requirements for the Arena Assign Module and the AllFusion PM UOB (Arena BE object type = Assign) ? Future:

Prompt Name	Description	AllFusion PM Meta Property	UDP Name	UDP Datatype	UDP Value List
Name	Unique name	Activity name	Not applicable	Not applicable	Not applicable
Type	Assignment type	UDP	Arena_A_Type	SS Text List	▪ Attribute ▪ Entity picture ▪ Entity type ▪ Other ▪ Variable
Variable name	Name if a variable	UDP	Arena_A_ VariableName	Text	Not applicable
Attribute name	Name if an attribute	UDP	Arena_A_ AttributeName	Text	Not applicable
Entity Type name	Name if an Entity Type	UDP	Arena_A_ EntityType	Text	Not applicable

Entity Picture name	Name if an Entity Picture		Arena_A_ EntityPictureName	Text	Not applicable
Other name	Name if Other	UDP	Arena_A_ OtherName	Text	Not applicable
New Value	Assignment value if not Entity Type or Entity Picture	UDP	Arena_A_NewValue	Text	Not applicable

Arena Resource Module and AllFusion PM UOB

The following table describes the requirements for the Arena Resource Module and the AllFusion PM UOB (Arena BE object type = Resource):

Prompt Name	**Description**	**AllFusion PM Meta Property**	**UDP Name**	**UDP Datatype**	**UDP Value List**
Name	Unique name	UDP	Arena_Res_Name	Text	Not applicable
Type	Resource type	UDP	Arena_Res_Type	SS Text List	▪ Fixed capacity ▪ Based on schedule
Capacity	Capacity	UDP	Arena_Res_ Capacity	Text	Not applicable
Schedule name	Schedule name	UDP	Arena_Res_ ScheduleName	Text	Not applicable
Busy/Hour	BusyPerHour	UDP	Arena_Res_ BusyPerHour	Real	Not applicable
Idle/Hour	IdlePerHour	UDP	Arena_Res_ IdlePerHour	Real	Not applicable
Per Use	PerUse	UDP	Arena_Res_ PerUse	Real	Not applicable

Arena Resource Repeat Group and AllFusion PM Arrow

The following table describes the requirements for the Arena Resource Repeat Group and the AllFusion PM Arrow (Arena BE object type = Resource Repeat Group):

Prompt Name	Description	AllFusion PM Meta Property	UDP Name	UDP Datatype	UDP Value List
Type	ResourceType	UDP	Arena_ResGroup_Type	SS Text List	▪ Resource ▪ Set
Resource Name	ResourceName	UDP	Arena_ResGroup_ResName	Text	Not applicable
Set Name	SetName	UDP	Arena_ResGroup_SetName	Text	Not applicable
Quantity	Quantity	UDP	Arena_ResGroup_Quantity	Text	Not applicable
Selection Rule	SelectionRule	UDP	Arena_ResGroup_SelectionRule	SS Text List	▪ Cyclical ▪ Random ▪ Preferred Order ▪ Specific Member ▪ Largest Remaining Capacity ▪ Smallest Number Busy
Save Attribute	SaveAttribute	UDP	Arena_ResGroup_SaveAttr	Text	Not applicable
Input Attribute	InputAttribute	UDP	Arena_ResGroup_InputAttr	Text	Not applicable

AllFusion PM to Arena Sample Models

The UDP library for Arena export, ArenaBEUDPs-5.bp1, can be merged into any model that you want to use as a source for an Arena Version 5.0 model. The following table describes the sample models for use with Arena Version 5.0:

Sample Model	Description
NC Vehicle Inspection.bp1	The diagram *Vehicle Inspection Certification Process* uses non-standard shapes. This diagram can be exported to Arena 5.0. When run in Arena, it will reveal problems with the process. This is the legitimate result. The diagram *Award New Inspection Certification Sticker* can be exported to Arena 5.0.
ACME Medical Equipment 41-5.bp1	This model contains two versions of a single decomposition diagram. These diagrams illustrate the definition and use (instances on named grey dashed arrows) of resources. The diagram *ACME Medical Equipment* is in IDEF3 notation. The IDEF3 alternate diagram, *ACME Medical Equipt.*, uses non-standard shapes. Either diagram can be exported to Arena by clicking Export, Arena from the File menu.
ArenaOrderEntry41-5.bp1	The diagram *Credit Verification* uses non-standard shapes that emulate those in Arena. The IDEF3 alternate diagram, *WORST CASE CREDIT*, is the same diagram in IDEF3 notation. Either diagram can be exported to Arena by clicking Export, Arena from the File menu.

The UDP library for Arena export, ArenaBEUDPs.bp1, can be merged into any model that you want to use as a source for an Arena Version 4.0 model. The following table describes the sample models for use with Arena Version 4.0:

Sample Model	Description
ArenaOrderEntry40.bp1	The diagram *Credit Verification* uses non-standard shapes that emulate those in Arena. The IDEF3 alternate diagram, *WORST CASE CREDIT*, is the same diagram in IDEF3 notation. Either diagram can be exported to Arena by clicking Export, Arena from the File menu.
ACME Medical Equipment 40.bp1	This model contains two versions of a single decomposition diagram. These diagrams illustrate the definition and use (instances on named grey dashed arrows) of resources. The diagram *ACME Medical Equipment* is in IDEF3 notation. The IDEF3 alternate diagram, *ACME Medical Equipt.*, uses non-standard shapes. Either diagram can be exported to Arena by clicking Export, Arena from the File menu.

Index

A

activity decomposition • 26
activity-based costing (ABC) • 29

E

exclusive OR junction • 18

I

IDEF3 model
    IDEF3 model, creating • 31
    IDEF3 model, defined • 7
    IDEF3 model, example • 31

J

junctions • 17, 18, 19, 22, 23

L

links • 9, 10, 11

M

modeling methods
    modeling methods, IDEF3 • 7

O

object flow link • 11
organization charts • 45, 52

P

process flow model
    process flow model, creating • 31
    process flow model, defined • 7
    process flow model, example • 31

R

referents • 9, 41
relational link • 9, 11

S

swim lane diagrams • 44, 51

T

temporal precedence link • 9, 10