AllFusion® Process Modeler Business Process Modeling

Design Guide

r7.2

This documentation and any related computer software help programs (hereinafter referred to as the "Documentation") is for the end user's informational purposes only and is subject to change or withdrawal by CA at any time.

This Documentation may not be copied, transferred, reproduced, disclosed, modified or duplicated, in whole or in part, without the prior written consent of CA. This Documentation is confidential and proprietary information of CA and protected by the copyright laws of the United States and international treaties.

Notwithstanding the foregoing, licensed users may print a reasonable number of copies of the Documentation for their own internal use, and may make one copy of the related software as reasonably required for back-up and disaster recovery purposes, provided that all CA copyright notices and legends are affixed to each reproduced copy. Only authorized employees, consultants, or agents of the user who are bound by the provisions of the license for the product are permitted to have access to such copies.

The right to print copies of the Documentation and to make a copy of the related software is limited to the period during which the applicable license for the product remains in full force and effect. Should the license terminate for any reason, it shall be the user's responsibility to certify in writing to CA that all copies and partial copies of the Documentation have been returned to CA or destroyed.

EXCEPT AS OTHERWISE STATED IN THE APPLICABLE LICENSE AGREEMENT, TO THE EXTENT PERMITTED BY APPLICABLE LAW, CA PROVIDES THIS DOCUMENTATION "AS IS" WITHOUT WARRANTY OF ANY KIND, INCLUDING WITHOUT LIMITATION, ANY IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NONINFRINGEMENT. IN NO EVENT WILL CA BE LIABLE TO THE END USER OR ANY THIRD PARTY FOR ANY LOSS OR DAMAGE, DIRECT OR INDIRECT, FROM THE USE OF THIS DOCUMENTATION, INCLUDING WITHOUT LIMITATION, LOST PROFITS, BUSINESS INTERRUPTION, GOODWILL, OR LOST DATA, EVEN IF CA IS EXPRESSLY ADVISED OF SUCH LOSS OR DAMAGE.

The use of any product referenced in the Documentation is governed by the end user's applicable license agreement.

The manufacturer of this Documentation is CA.

Provided with "Restricted Rights." Use, duplication or disclosure by the United States Government is subject to the restrictions set forth in FAR Sections 12.212, 52.227-14, and 52.227-19(c)(1) - (2) and DFARS Section 252.227-7014(b)(3), as applicable, or their successors.

CA Product References

This document references the following CA products:

- AllFusion® Process Modeler (AllFusion PM)
- AllFusion® ERwin Data Modeler (AllFusion ERwin DM)
- AllFusion® Model Manager (AllFusion MM)

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 4: Activity Decomposition Diagrams 47

Chapter 5: Additional IDEF0 Features 55

Index 81

Chapter 1: Activity Modeling

This section contains the following topics:

Activity Modeling and Business Systems Engineering

Activity modeling is a critical component of business systems engineering because a company's strategy is implemented by activities. It is the success or failure of a company to perform these activities that ultimately determines success or failure of a particular strategy.

Corporate strategy can be thought of as a compass, steering the company through turbulent markets with a long-term broad view of the company's intentions for where it wants to be. Strategy is implemented by activities, so activity models serve as a company's road map, helping the company navigate the local terrain to get where it wants to be.

Briefly, corporate strategy covers four general areas:

- Market segments the company chooses to serve.
- Products and services the company will offer to these market segments.
- Distribution and marketing channels the company will use to reach these market segments.
- Activities and processes the company will use to execute the strategy.

Deciding which market segments the company intends to serve is the most significant decision, because it focuses and constrains the other issues. The final list of exact markets will certainly evolve during strategy development, but significant changes in the target market can invalidate other important decisions.

The general strategy for serving these markets has three dimensions:

- Cost leadership
- Differentiated value
- Market scope

A company can strive to be the cost leader in the target segment, offering the lowest price and making a profit through manufacturing efficiencies. A second approach is to completely ignore costs and offer the highest value to the target market. It is also possible to balance the two approaches. A company can also exclusively focus on serving a small niche market, target several niches with focused strategies, or cover a broad market with a common strategy.

If a company cannot reach a target market directly, then it develops channels, or intermediaries, that can better reach the target segment, for example, by being geographically closer. Different kinds of resellers are usually required to reach different market segments. Companies also use various channels to communicate their marketing messages.

The company must identify and design the activities that will be used to fulfill the strategy. Activity models play an integral role in this crucial step of strategy development. Activities must be designed to meet the company's objectives, and costs must be weighed against perceived value. Often, discoveries made while designing activities and processes, such as the activity's costs, may result in changes to other strategic decisions.

For example, even though customers in a particular target segment may need extensive training to use a particular product, the customers may not be willing to pay enough for the company to justify the expense of developing an appropriate training program. This can have a tremendous impact on the company's strategy for that market segment.

Strategy development and implementation are themselves processes that can be analyzed and designed. The more efficiently a company can revitalize its strategy to capitalize on new opportunities and counter new competitive threats, the greater its chance of sustaining a long-term competitive advantage.

If a company neglects to continuously reexamine and revitalize its competitive strategy, it will one day find itself at a competitive disadvantage. Such was the case for many American companies in the late 1980s, when management gurus Michael Hammer and James Champy, authors of the book *Reengineering the Corporation*, began to expound on the need for companies to reinvent themselves, to discard current business practices and develop new ones. Business process reengineering is a strategy development and implementation process in which the existing strategy is discarded, and activity modeling can be utilized.

The increasing reliance on computing technology to support business activities has also increased the need to communicate information about activities among an increasingly diverse audience of business and computer systems professionals. The failure of the computer systems profession to consistently deliver correctly working software on time and within budget (commonly referred to as the *software crisis*) has been largely due to a combination of four root causes:

- Failure to meet stated system requirements
- Inadequate or incorrect system design
- Inadequate system performance
- Failure to properly address the human/system interface

A thorough understanding of the activities and processes the computer system is expected to support is crucial to successful software development. As computer systems continue to permeate society, the cost of such failures will increase. This is why a top priority on the agenda of company executives is to better align information technology with business needs.

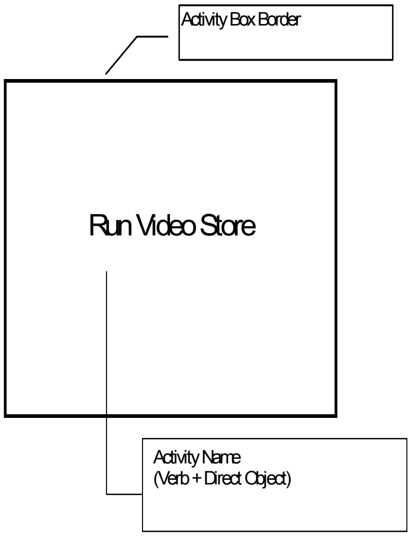
Activity and Process Models

Companies transform raw materials into finished goods through a complicated mass of interrelated business functions or *activities* and business *processes*. To a large extent, a company's success or failure depends on its ability to identify, design, and execute appropriate activities better than your competition can. Therefore, activities are at the heart of an organization.

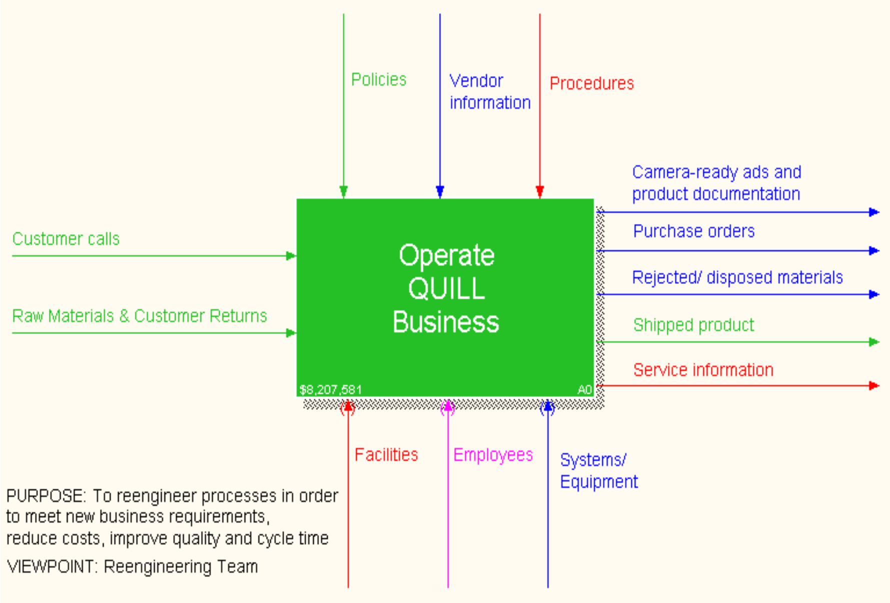
Textual descriptions of activities and processes are usually long and complicated, making them difficult to understand. But without a clear understanding of a company's activities, it is impossible to effectively design new activities, or understand existing ones. What is needed is a technique to document activities in a precise format, clarifying and organizing important information while eliminating the superfluous. In this way, activities can be effectively analyzed, designed, and implemented.

Activity and process models filter and organize information through their syntax and by a rigorous model development process. Activity models can be viewed as an extended form of punctuation that combines with a well-defined publishing format to filter and organize conversational language.

An activity model looks at a *system* as a collection of activities in which each activity transforms some object or collection of objects. Activity models emphasize activities by representing them with special graphic shapes, such as a box. Activities are usually labeled with a verb or a verb and a direct object that represents what the activity accomplishes. As shown in the following illustration, a labeled box represents an activity:



The interface between an activity and its environment, including other activities, is graphically depicted using arrows. The concept of activity remains relatively the same among the various modeling techniques, but the meaning of the arrows, and the meaning of their connections with the boxes, changes from one modeling technique to another.

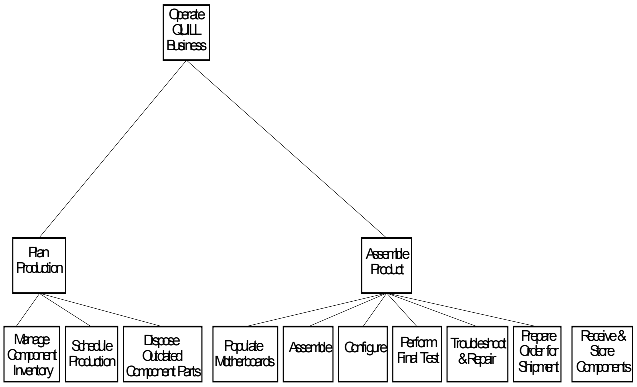


PURPOSE: To reengineer processes in order to meet new business requirements, reduce costs, improve quality and cycle time

VIEWPOINT: Reengineering Team

In the previous illustration, arrows represent the activity of the interface with its environment. Like slides used for presentations, activity model diagrams control the level of detail that is exposed at once. The goal of subdividing a model into diagrams is to gradually expose the details of a subject, striking a balance between *clarity*--ensuring that a given diagram is understood by its intended audience; and *scope*--ensuring that a given diagram contains all relevant information.

The activity box shown in the previous illustration represents the system's border, or boundary. If we take a closer look at this box, actually look inside it, we would see other boxes. These boxes, in turn, may have their own *child* boxes. The following illustration represents the *hierarchical* relationship of an activity in a graphical display format:



A more abbreviated hierarchical display format, showing a partial activity hierarchy in outline form, is illustrated below:

Activity Hierarchy

- Operate Quill Business
  - Plan Production
    - Manage Component Inventory
    - Schedule Production
    - Dispose of Outdated Component Parts
  - Assemble Product
    - Populate Motherboards
    - Assemble
    - Configure
    - Perform Final Test
    - Troubleshoot and Repair
    - Prepare Order for Shipment

Chapter 2: Business Process Modeling Method (IDEF0)

This section contains the following topics:

Business Process Modeling

IDEF0 activity modeling is a technique for analyzing whole systems as a set of interrelated activities or functions. This purely functional orientation is important--the functions (verbs) of a system are analyzed independently of the objects that perform them. The idea is that the packaging of functions within systems can be performed as part of a new design process but is irrelevant and possibly even misleading as part of analysis. A purely functional perspective allows for a clear separation of the issues of meaning from the issues of implementation.

IDEF0 is best suited as an analysis and logical design technique. As such, it is generally performed in the early phases of a project, perhaps being preceded by IDEF3 modeling for data collection and AS-IS process modeling. An analysis using IDEF0 modeling can feed a design process using IDEF3 models and DFD diagrams.

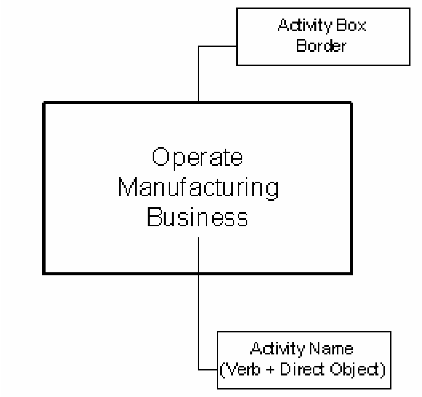
Note: For more information regarding the IDEF3 and DFD modeling methods, see the AllFusion Process Modeler *Process Flow Modeling Design Guide* and the AllFusion Process Modeler *Data Flow Diagramming Design Guide*.

Activities

An activity, sometimes called a function, processes or transforms inputs into outputs. Because IDEF0 models a system as a set of hierarchical (nested) activities, the first activity to be defined is the activity that describes the system itself--the *context* activity. This is drawn as a box, and is given a name.

Activity names in IDEF0 generally consist of a single, active verb plus a common noun that clarifies the objective of the activity from the viewpoint of the model. An adjective may be used to further qualify the noun. It is important that the activity name accurately reflect the system as it is observed from the specified viewpoint for the model.

An activity box is shown in the figure below. The activity is properly labeled with a verb + direct object. The word *manufacturing* functions as an adjective to qualify the noun *business*.


Activity Box Border
Operate Manufacturing Business
Activity Name (Verb + Direct Object)

It was noted earlier that IDEF0 models a system as a set of *hierarchical* (nested) *activities*. An activity can be *decomposed* (broken down) into its constituent activities. Decomposition is commonly referred to as top-down modeling, but this is a misnomer. Functional decomposition can be more correctly viewed as outside-in modeling, in which the layers of a system are peeled away in order to view the inner details.

The following is part of an activity hierarchy in node index format:

Operate QUILL Business

Sell and Market Products
Manage Advertising
Take Orders
Answer Phone Calls
Provide General Information
Answer Fax Pricing Requests
Record Order Information
Check Credit
Provide Order Follow-up
Provide Pricing Information
Develop Documentation
Prepare Work Ticket

Design Configuration
Identify Vendors and Components
Develop Specifications
Specify Components
Test Configuration
Approve Vendors

Plan Production
Order Assembly Components
Issue Work Ticket
Manage Component Inventory
Schedule Production
Dispose Outdated Component Parts

Boundary and Interface (Arrows)

To be useful, the description of an activity must at a minimum also include a description of the objects the activity creates as output, as well as those objects the activity consumes or transforms.

In IDEF0, *controls* and *mechanisms* are also modeled. Controls are the objects that govern the manner in which inputs are transformed yet are not themselves transformed by the activity. Mechanisms are those objects that actually perform the transformation of inputs to outputs yet are not themselves transformed by the activity.

ICOM is the acronym for the categories of information that are captured on IDEF0 diagrams. It represents the four types of arrows, which are *Input*, *Control*, *Output*, and *Mechanism*.

Whereas activities are always verbs or verb phrases, arrows are always nouns. The arrows can represent people, places, things, concepts, or events. An arrow is represented on IDEF0 diagrams by lines with an arrowhead at one end. The line is labeled with the name of the arrows. As is the case with activities, an arrow name is not enough to guarantee that the model will be understood by its readers. A textual definition of each arrow is critical to developing an accurate and useful model.

Input Arrows

Inputs represent material or information that is consumed or transformed by the activity in order to produce the outputs. Input arrows always enter the left side of an IDEF0 activity box. *Input arrows are optional*, since some activities do not transform or change anything. An example of an activity without an input might be Make Executive Decision, where several factors are analyzed to produce a decision, but none of the factors is transformed or consumed by the decision.

Control Arrows

Controls govern or regulate how, when, and if an activity is performed and which outputs are produced. Since controls govern the conduct of an activity to ensure the creation of the desired output, e*ach activity must have at least one control arrow.* Controls *always* enter the top of an IDEF0 activity box.

Controls are often in the form of rules, regulations, policies, procedures, or standards. They influence the activity without actually being transformed or consumed. There will be times when an activity's purpose is to change a rule, regulation, policy, procedure, or standard. In this case, it would be expected that the arrow containing that information would be an input.

A control is a special type of an input to an activity. If it is unclear whether an arrow should be modeled as an input or a control; choose control until the ambiguity can be resolved. Integrating data and process models is a valuable technique for resolving ambiguities.

Output Arrows

Outputs are the material or information produced by the activity. *Each activity must have at least one output arrow*. An activity that does not produce a definable output should not be modeled (or, at a minimum, should be a candidate for elimination).

In a non-manufacturing environment, outputs are often data that was processed in some form by the activity. It is important to use modifiers in front of the arrow labels to indicate how the output data is different from the input data. For example, the activity Admit Patients may have *patient data set* as both an input and an output. Properly labeled, the input arrow could be labeled *raw patient data*, and the output arrow could be labeled *verified patient data*.

Mechanism Arrows

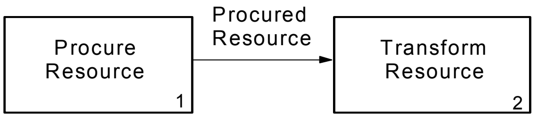
Mechanisms are those resources that perform the activity. Mechanisms could be the important people, machinery, and equipment that provide and channel the energy needed to perform the activity. A mechanism arrow may be omitted from an activity if it is determined that it is not needed to fulfill the model's purpose.

Arrow Interface Combinations

There are five basic arrow interface combinations:

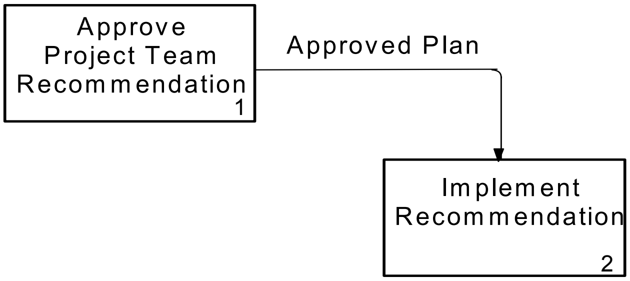
Output-Input

Represents an activity with precedence over another activity. In the following illustration, *Procure Resource* must precede *Transform Resource*:



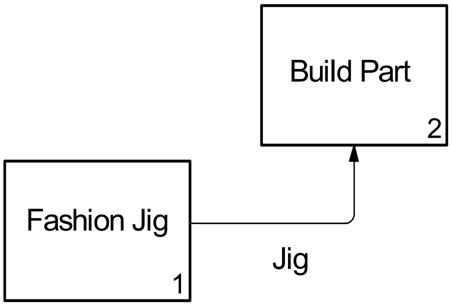
Output-Control

Represents activity dominance, in which one activity is dominant over another because it controls how that activity transforms inputs into outputs. In the following illustration, the *Approved Plan* guides the implementation of the recommendations. The recommendations themselves are not changed by the implementation, so the *Approved Plan* is depicted as a control arrow of *Implement Recommendations*.



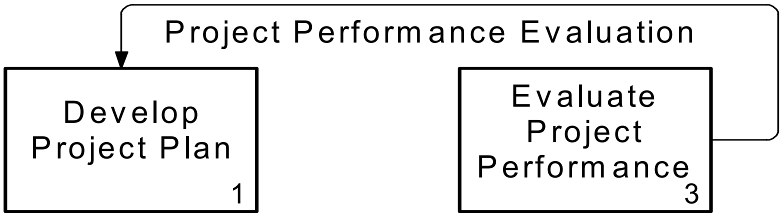
Output-Mechanism

Represents a situation in which the output of one activity becomes the means to perform another. These types of arrows are less common. In the following illustration, a jig (a device used to temporarily hold something in place while it is being manipulated) must be fashioned in order to *Build Part*.



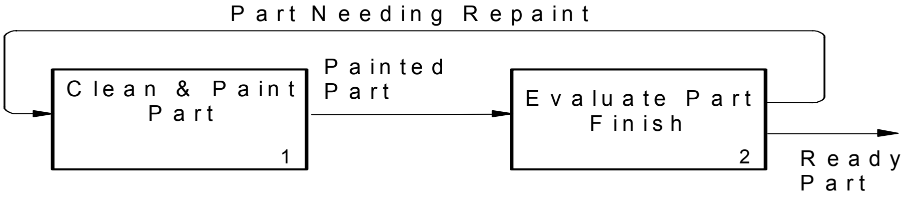
Output-Control Feedback

Represents scenarios where boxes of normally lesser dominance provide feedback to boxes of greater dominance. The following illustration is an example of output-control feedback. The output of *Evaluate Project Performance*, *Project Performance Evaluation*, feeds back to the activity, *Develop Project Plan*. The intent is to apply the lessons learned on this project so that future projects can be managed better. *Project Performance Evaluation* is a control arrow to *Develop Project Plan* because the evaluation is not transformed by the activity *Develop Project Plan*.



Output-Input Feedback

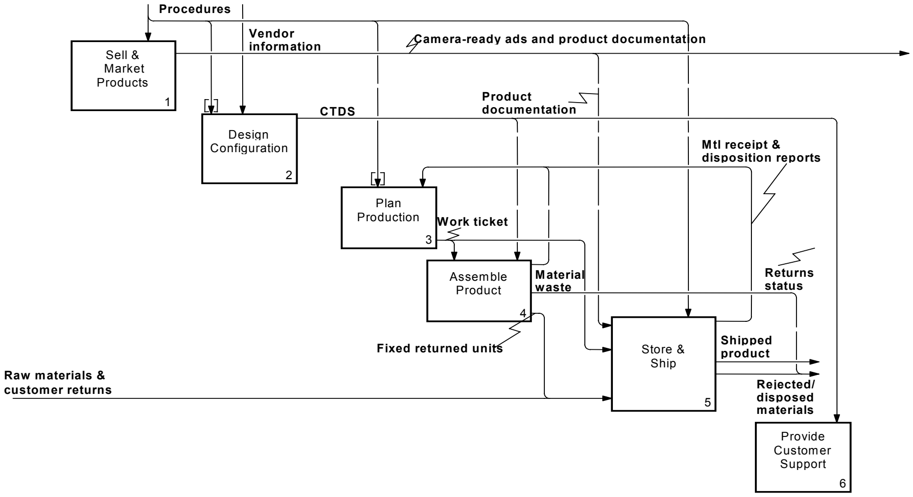
Represents scenarios where boxes of normally lesser dominance provide feedback to boxes of greater dominance. Output-input feedback is commonly used to describe rework cycles. The following illustration is an example of an output-input feedback:



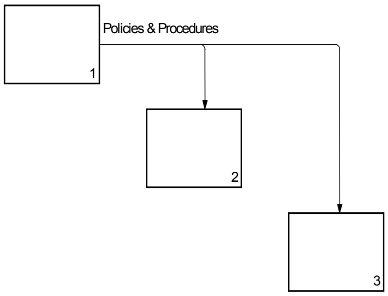
Output-input feedback is also used to describe cases where scrap materials can be reused along with raw materials. For example, in the bottle-making process, bottles that break during manufacture are simply remelted and fashioned into other bottles.

Arrow Branching and Joining

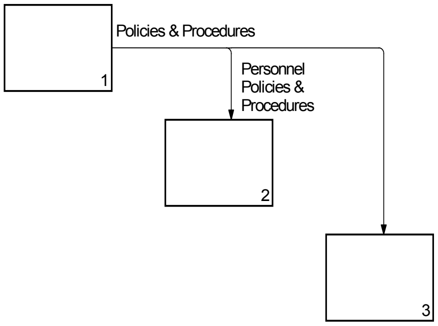
The outputs of an activity may be used by more than one other activity. In fact, IDEF0 is most valuable as a tool to help visualize activity interdependencies within a system. IDEF0 arrows may branch (split) and join (merge). The following illustration is an example of an IDEF0 diagram that only shows branching and joining arrows:



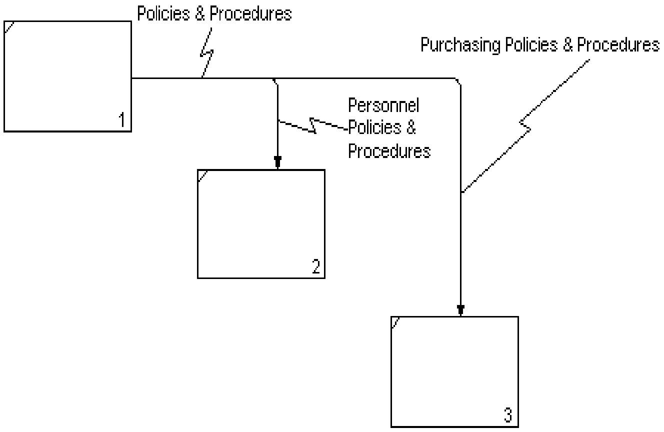
The definition of a split or joined arrow is made clear by the judicious labeling of the arrow segments that join and split. In the following illustration, the control arrow *Policies & Procedures* is an example of an arrow that splits but is not renamed. The *Policies & Procedures* arrow splits and is a control to both activity 2 and activity 3, which means that both activities use *Policies & Procedures* to guide their transformation of inputs to outputs.



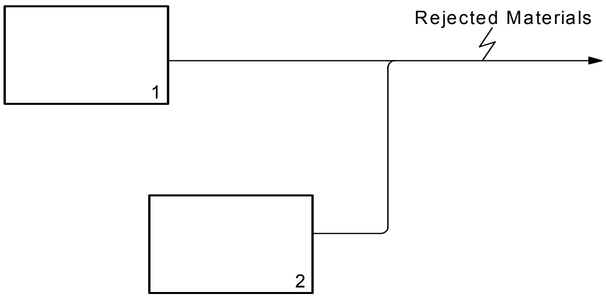
In the following illustration, one of the arrow branches is renamed to show that only a subset of *Policies & Procedures* is used by activity 2. If an arrow is split and a branch is renamed, the new name must reflect a subset of the original arrow. Arrow splitting represents data decomposition (just as activity hierarchy represents function decomposition).



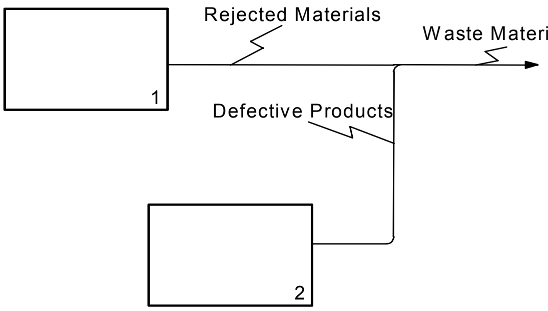
The following illustration is another example of data decomposition. The squiggles seen in this example are often used in IDEF0 diagrams to clearly link an arrow label and the line.


Policies & Procedures
Purchasing Policies & Procedures
Personnel Policies & Procedures
1
2
3

In the following illustration, it is implied that the output of activities 1 and 2 is also labeled *Rejected Materials*:



In the following illustration, each arrow segment is labeled to clearly identify the individual components and the bundled result:



Arrow Hierarchy and Data Associations

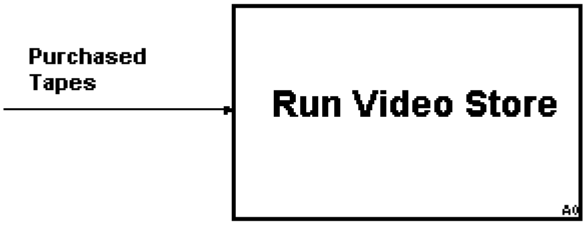
You can branch a single arrow in a context-level diagram into multiple arrows in decomposition diagrams. You can also join output arrows in decompositions into a single output arrow in a context diagram. This branching and joining of arrows creates an arrow hierarchy. You create a new level in the hierarchy each time you branch or join an arrow.

Just as in the activity hierarchy, the arrow hierarchy includes leaf-level arrows. Leaf-level arrows are arrows that are not branched further in the model and are not the result of joining two or more arrows.

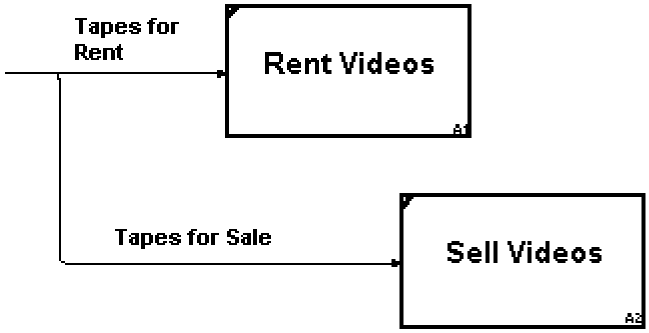
In a video store example, the *Tapes for Rent* and *Tapes for Sale* arrows are leaf-level arrows. Because both arrows are branched from the *Purchased Tapes* arrow, the *Purchased Tapes* arrow is considered the parent arrow of the *Tapes for Rent* and *Tapes for Sale* arrows in the arrow hierarchy.

Before you add arrow data to your model, first decide what data is associated with each arrow in your diagram. You can create a table to show the arrow data associations.

Example: Video Store Context Level Diagram

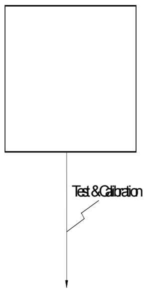


Example: First Decomposition of the Video Store Context Level Diagram with Two Leaf-Level Arrows



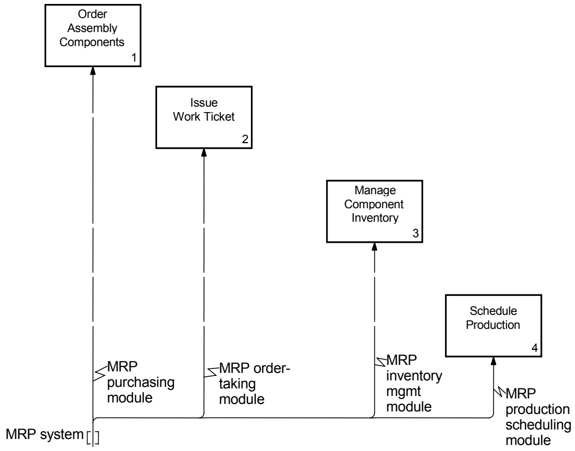
Call Arrows

Call arrows are used to reference other models or diagrams within a model that may help the reader better understand the current model, or perform the same function. For example, in the following illustration, *Test & Calibration* is the name of another model that documents testing and calibration. Rather than being duplicated, the model is simply referenced, that is, *called*. A call arrow can refer to another diagram (more correctly, the diagram's parent activity) within the same model, and can also refer to a specific child activity in another model. This is a way in which functional duplication (similar functions operating on different data) can be properly documented in the model.

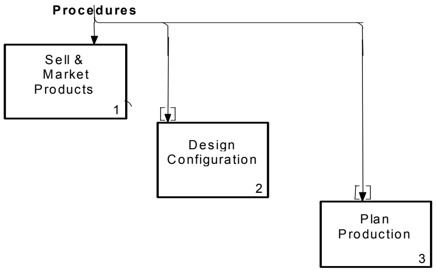


Tunnels

Arrow bundling is used to control the level of detail in a diagram. In the case of an arrow that does not warrant appearing on the parent diagram (it is not significant enough), but does not belong bundled with other arrows, the *tunnel* is used to indicate that the arrow exits or enters the system. In the following illustration, for example, *MRP System* is an important mechanism on this diagram, but is probably not used anywhere else in the model. Tunneling is used as an alternative to cluttering up parent diagrams by including this mechanism arrow.


Order Assembly Components
1
Issue Work Ticket
2
Manage Component Inventory
3
Schedule Production
4
MRP purchasing module
MRP order-taking module
MRP inventory mgmt module
MRP production scheduling module
MRP system

In other cases, a tunnel can be used on an arrow leading to or from a parent activity. This indicates that the arrow's relationship with the activity's children is undefined. In the following illustration, the tunnels indicate that the *Procedures* arrow is not detailed in the decomposition of *Design Configuration* or *Plan Production*.



Activities and Activations

In the machining industry, there are various activities that are performed such as drill, lathe, route, and punch. However, it is not typical to perform all these activities every time a machining operation is performed. It is more typical to perform a subset of these activities.

For example, wood is drilled differently than metal, because wood is generally fashioned with screws and metal is fitted with bolts or rivets. A screw hole is usually countersunk, so that the screw head is embedded in the wood. Bolts are seldom countersunk. The model can be decomposed to create separate activities for drilling, bolting, and so on. Another option is to create separate IDEF3 models for each of these activities. This is particularly useful for evaluating drill, bolt, and other activities using simulation.

As a simpler alternative to the above two approaches, an *activation table* can describe the various input, control, output, and mechanism combinations for each activation of an activity. An activation is a unique configuration of input and control values, and resource requirements.

The following table is such an activation table. Each activation is given a name unique within the activity, and the values of the various arrows are listed. The combination of arrow values should be unique for each activation, that is, two activations should not have the same values for all of their arrows. It also describes policy, for example, a bank may have a policy for counting cash that stipulates a counting machine be used for cash amounts greater than $500.

Activation Name	Arrow Name	Arrow Value
Large amount of cash	Cash	>$500
	Cash-counting machine	1 Required
Small amount of cash	Cash	<$500
	Cash-counting machine	0 Required

Using IDEF3 Models to Illustrate IDEF0 Activations

IDEF3 models can be built from an IDEF0 model to illustrate the activations of leaf-level activities (that is, non-decomposed activities or activities with no children). If the authors intend to reuse IDEF0 models in this way, then the activations of each activity's interface should be carefully documented. The following table describes different activations of the activity *Process Account Deposit*. As an alternative, IDEF3 models can be built and the activation models derived from these.

Activation Name	Arrow Name	Arrow Value
Check-only	Cash in	=$0.00
	Check	>$0.00
	Deposited	= Check Total - Return
	Return	N/A
	Cash out	= Return
	Cash-counting machine	0 Required
	Teller	1 Required
Cash-only	Cash in	>$0.00
	Check	=$0.00
	Deposited	= Cash Total - Return
	Return	N/A
	Cash out	= Return
	Cash-counting machine	If cash in > $500, then 1 required Else, 0 Required
	Teller	1 Required
Cash & check	Cash in	>$0.00
	Check	>$0.00
	Deposited	= Check Total + Cash Total - Return
	Return	N/A
Cash & check	Cash out	= Return
	Cash-counting machine	If cash in > $500, then 1 required Else, 0 Required
	Teller	1 Required

Chapter 3: Business Process Models

This section contains the following topics:

IDEF0 Model Structure

IDEF0 combines a limited graphical notation with a rigorous and well-defined process by which the models are authored to improve the quality of the completed model. The graphical notations used in an IDEF0 model are boxes and arrows.

The IDEF0 methodology is similar in some respects to that used for publishing books, and often a set of printed IDEF0 models are organized into a binder with a table of contents, a glossary, and other “book-like” matter. In IDEF0, this is known as a *kit*.

The first step in building an IDEF0 model is to identify the *purpose* of the model, or the set of questions the model is intended to answer. These questions should be enumerated as part of the model documentation, much like the preface of a book describes its purpose. A statement of purpose, which summarizes the questions, is often written.

The model's *scope* consists of the breadth and depth of the detail and is similar to information usually included in a book's preface. It is not sufficient to merely list the set of questions you want to answer. The reader of the model, and so too, those who are to actually author the model, need to understand the amount of detail they should expect for each answer. The scope statement can be summarized as the activity name that appears in the model's context diagram.

In addition, you need to identify the intended *audience*. Often, the intended audience will have a great impact on the level of detail that can be included, or should be included, in a model. Consider what information the audience already knows about the subject, the background or technical information that they might need to understand the subject, and the language and style that is most appropriate.

Viewpoint is the perspective from which the model will view the system. The viewpoint is chosen to encompass the chosen scope and fulfill the purpose. Once selected, the viewpoint must remain consistent throughout a single model. If necessary, other models should be created to model the system from different viewpoints. Some viewpoints are customer, supplier, store owner, and editor.

IDEF0 modeling always starts with a context diagram. When you create a business process model, a context diagram is created with one activity that defines your model. You can then add decomposition diagrams that can contain activities, arrows, and related properties. The context diagram depicts the highest-level activity in a model, and represents the boundary of the process under study with respect to purpose, scope, and viewpoint.

IDEF0 Diagrams

IDEF0 diagrams are used to model a system as a set of activities (functions). They use only two graphic symbols to accomplish this, boxes and arrows.

- Activities are represented by boxes containing a single, active verb plus a common noun that clarifies the objective of the activity from the viewpoint of the model, for example, Obtain Driver's License. You can use an adjective to further qualify the noun.
- Arrows are typically labeled with nouns and are used to represent four types of information that are connected to an activity:

  Input

  Represents what is consumed or transformed by an activity.

  Control

  Represents the objects that govern the manner in which inputs are transformed yet are not themselves transformed by the activity; a constraint on the operation of the activity.

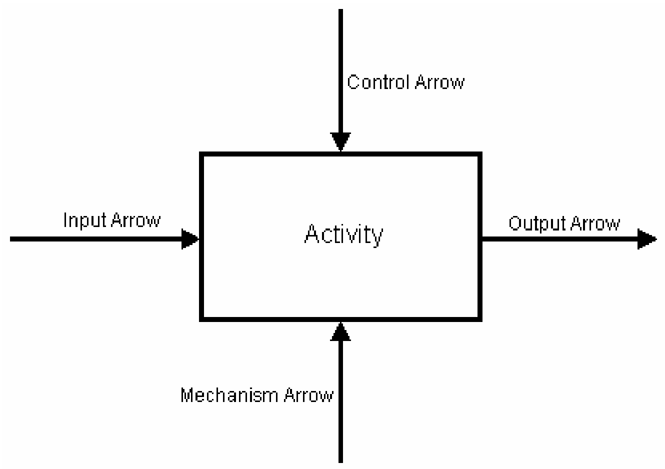
Output

Represents what an activity produces or creates; the result of the process.

Mechanism

Represents those objects that actually perform the transformation of inputs to outputs yet are not themselves transformed by the activity; objects that are used to perform the process, but are not themselves consumed.

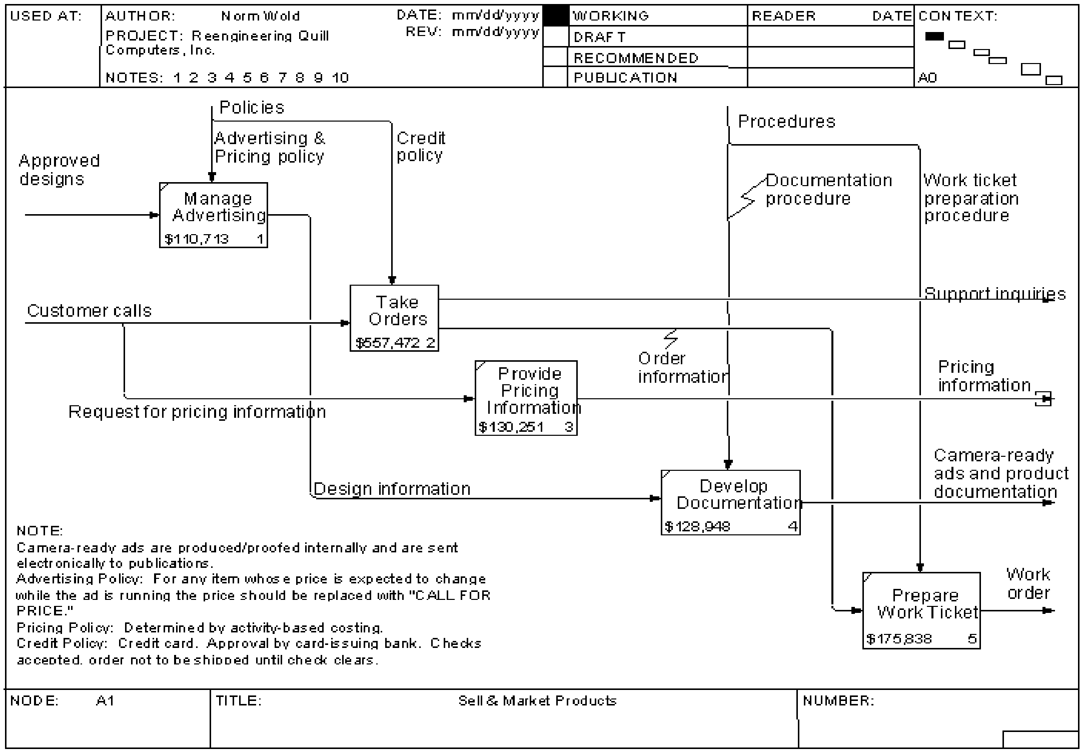
The acronym ICOM is used to represent these four types of arrows. The following illustrates these four arrow types, showing the specific box side of an IDEF0 activity to which each arrow type must connect:



IDEF0 Diagram Kit and Title

In IDEF0 diagrams, it is important to have a well-defined kit and title. The kit is the diagram header and is used to track the progress of the diagram. The title is the diagram footer, and displays the diagram's identity and parentage. The following illustration is an example of an IDEF0 diagram showing its kit and title areas.

Example: IDEF0 Diagram with Kit and Title Areas Defined



USED AT:
AUTHOR: Norm Wold
PROJECT: Reengineering Quill Computers, Inc.
NOTES: 1 2 3 4 5 6 7 8 9 10
DATE: mm/dd/yyyy
REV: mm/dd/yyyy
WORKING
DRAFT
RECOMMENDED
PUBLICATION
READER
DATE
CONTEXT:
A0

NOTE:
Camera-ready ads are produced/proofed internally and are sent electronically to publications.
Advertising Policy: For any item whose price is expected to change while the ad is running the price should be replaced with "CALL FOR PRICE."
Pricing Policy: Determined by activity-based costing.
Credit Policy: Credit card. Approval by card-issuing bank. Checks accepted. order not to be shipped until check clears.

NODE: A1
TITLE: Sell & Market Products
NUMBER:

The following information describes the border elements of the kit:

Used At

Documents all of the places where this diagram's parent box is called by a call arrow.

Author

Describes who originally created the diagram.

Date

Specifies the creation date of the diagram.

Project

Specifies the project title under which the diagram was created. It is used to track diagrams within projects.

Rev

Specifies the date the diagram was last changed.

Notes 1 2 3 4 5 6 7 8 9 10

Provides an area where readers can cross off a number each time they create a new note on the diagram, when diagrams are edited by hand.

Status

Reflects the stage of approval for the diagram. It is used to implement a formal publishing process with review and approval steps. It includes the following options:

Working

Specifies a new diagram, a major change, or a new author for an existing diagram.

Draft

Specifies that the diagram has reached some level of acceptance by the readers. It is ready for evaluation by a review committee.

Recommended

Specifies that the diagram and all its supporting text was reviewed and approved. It is not expected to change.

Publication

Specifies that the diagram is ready for final printing and publication.

Reader

Lists the names of the readers.

Date

Specifies the date of the reading.

Context

Outlines of the activity boxes in the parent diagram, with the parent box of the diagram highlighted. Displays the node number of the parent box as well. The context field of the context diagram (the topmost diagram) should read *TOP* to indicate that there is no parent diagram in the model.

The following information describes the border elements of the title:

Node

Specifies the node number of the diagram, which is the same as the activity number of the diagram's parent activity box.

Title

Specifies the name of the parent activity box.

Number

Specifies the unique identifier, or C-Number, for *this* version of *this* diagram. Any new version of a diagram will have a new C-Number. Typically the C-Number consists of the author's initials (which must be unique among all modelers on the project) and a sequential, unique identifier, such as *JDM001*. Upon publication, the C-Number can be replaced with a standard page number. If a diagram replaces another, then the C-Number of the diagram that is replaced is often included in parenthesis, such as *JDM002 (JDM001)*. This provides a complete history of revisions to all diagrams in the model.

Note: C-Numbers are never reused on a project.

Author-Reader Cycle

Like the author/editor cycle used in book publishing, IDEF0 diagrams are typically reviewed and edited to validate their correctness and improve their quality.

When an author is ready for review, the author typically prepares a *kit,* or folder, for each reviewer who will review the model and make comments (notes) on the diagrams or related text. These commented diagrams are then returned to the author who makes the corrections. If there is disagreement, a commented diagram can be distributed to other readers for group consensus.

A formal review and approval mechanism is supported by the Status field and diagram versioning, and history is supported through the Number field. Usually, a librarian is assigned to large projects, and various levels of controls are implemented depending on the size of the project.

Purpose Statement

No model should be constructed without a clear objective or purpose statement. Components of a purpose statement should answer the following questions:

- Why is the process being modeled?
- What will this model show?
- What can readers of the model do with it?

A purpose statement allows the modeling team to remain focused throughout the modeling effort. Without a purpose statement, modeling sessions may drift aimlessly. An example of a purpose statement can be:

Identify the tasks of each shop worker and understand how the tasks relate in sufficient detail to develop a training manual.

Models are built to answer a set of questions. These questions should be developed early on and serve as the basis for the model's purpose. Using the purpose statement example, some of these questions can be:

- What are the foreman's tasks?
- What are the machinist's tasks?
- Who inspects completed products?
- Who inspects subassemblies?
- How do the assemblies move through the shop?
- Which tools are needed for which steps?

Viewpoint

Although it is important to include individuals with differing viewpoints in a modeling session, a given model must still be constructed from a single specific viewpoint. Often, other viewpoints are documented briefly in attached *For Exposition Only* (*FEO)* diagrams. These diagrams are intended for presentation only.

The viewpoint must be chosen with care, based on the purpose. In the Machine Shop example, only the foreman sees the interrelationships of the different tasks, so the model should be developed from the foreman's perspective.

It is important to maintain a single viewpoint throughout the model. The viewpoint should be a job title, department, or role (such as foreman or welder). As is the case for a purpose statement, a specific viewpoint is necessary to prevent drifting and continuous restructuring.

You may need to build models from different perspectives in order to document all of the activities in detail.

More information:

For Exposition Only Diagrams (see page 59)

Scope

One of the primary benefits of constructing an activity model is the clarification of the *scope* of an entire system and its specific component activities. Although it is expected that the scope will be modified slightly during the modeling effort, it must be maintained to direct the modeling effort. As was the case with the purpose statement, without a defined scope it is difficult to know when the model is complete, as the scope of a model tends to grow as the model grows.

The scope has two components:

Breadth

Defines the lateral borders of the modeling effort.

Depth

Defines the level of detail of activity decomposition.

To facilitate the definition of an accurate scope, many IDEF0 modeling efforts spend considerable time developing the model's context diagram. A diagram may even be developed that represents one level above the context diagram to validate the larger system within which this system resides. It is well worth the extra effort since the context diagram becomes the anchor and central point of reference for the entire modeling effort because all changes to the context diagram cascade down to the decomposition diagrams.

When the scope statement is defined, material that will not be included in the model is clarified as well.

More information:

Breadth-first Compared to Depth-first Modeling (see page 40)

Context Activity Naming

After you have created the purpose, viewpoint, and scope for your IDEF0 diagram, you need to create the activity name that appears in the context diagram to summarize the scope statement. The activity name is the highest level activity in the model.

The context activity must be assigned an active verb phrase that is consistent with the model's scope statement. For this reason, broad verb phrases such as Manage Customer Service or Process Applications are often used as labels. It is expected that the model's scope will fluctuate slightly during the first few days of modeling. Purpose and viewpoint should be rigid from the start. To illustrate issues from different viewpoints, FEO diagrams can always be developed.

It is important to always begin with the definition of the purpose, viewpoint, and scope of the model. The rigid definition of these terms is critical to the entire modeling effort and is worth a significant investment of time. Throughout the modeling project, the model's purpose, viewpoint, and scope is continually referenced to guide the project.

Order of the Definition of the Major ICOMs

With respect to order of construction of the arrows in IDEF0 diagrams, it is often easier to start from the outputs, then move to the inputs, followed by the mechanisms and controls. Each activity exists for a specific function, and that function often has readily identifiable outputs. Should the outputs of a given activity be difficult to ascertain, it may be a sign that an opportunity exists to improve the business process.

Definition of Outputs

As the outputs are identified, it is important to note that the model captures *if ever* scenarios. That is, if it is ever possible for a given situation to exist in the business, it must be modeled to show that possibility. Many new modelers forget to model as outputs the negative results of an activity. For example, the activity *Administer Driver's Test* will certainly produce an output of licensed drivers; however, there should be another output arrow for failed applicants. Negative results are often used as feedback arrows and must be considered for every activity. Likewise, it is important to include questionable arrows in a diagram and let the business experts decide whether they should be included in the model.

Definition of Inputs

After the outputs are created, inputs are considered. Inputs are specifically transformed or consumed by the activity to produce the outputs. In the manufacturing industry it is easy to note how raw material inputs are transformed and consumed into drastically different outputs. However, in the information industry, an input of data may initially appear not to be transformed or consumed at all. It is very rare that an input and an output arrow are labeled exactly the same. Generally, this indicates that this activity is adding little value to the business, or the output was improperly labeled. The solution is to use adjectives to modify the nouns in the arrow labels to indicate the transformation that took place during the activity. For example, an input can be labeled "raw patient data," and the corresponding output can be labeled "verified patient data." The adjectives *raw* and *verified* modify *patient data* to clarify the transformation.

Definition of Mechanisms

After the outputs and inputs are created, it is time to consider the mechanisms, or resources, applied to the activity. Mechanisms include people, machinery, computer systems, and so on. For example, an activity, Make Part, often requires some kind of machinery, such as a drill. In the Administer Driver's Test, a proctor administers the test. In the hospital example, some member of the clerical or nursing staff verifies the data, often with the help of another mechanism--computer equipment.

Definition of Controls

Finally, after the outputs and inputs are created and mechanisms defined, the controls that regulate the activity are added. Controls often are in the form of rules, regulations, policies, procedures, or standards. All activities in IDEF0 are required to have at least one control. Uncertain cases of input compared to control status should default to control. Remember, controls are still a form of input to an activity.

When the context diagram appears complete and stable, ask the following questions:

- Does the diagram summarize the business activity that will be modeled?
- Is the context diagram consistent with the scope, viewpoint, and purpose statements?
- Are the arrows at an appropriate level of detail for the activity? **Note:** As a guideline, you should limit the number of arrows to six per type.
- Does the model have work group consensus?

Enumeration of Activities and Diagrams

All IDEF0 activities are numbered. A prefix of any length can be used, but in almost all models A is used. Following the prefix is a number. The root activity is almost always numbered A0.

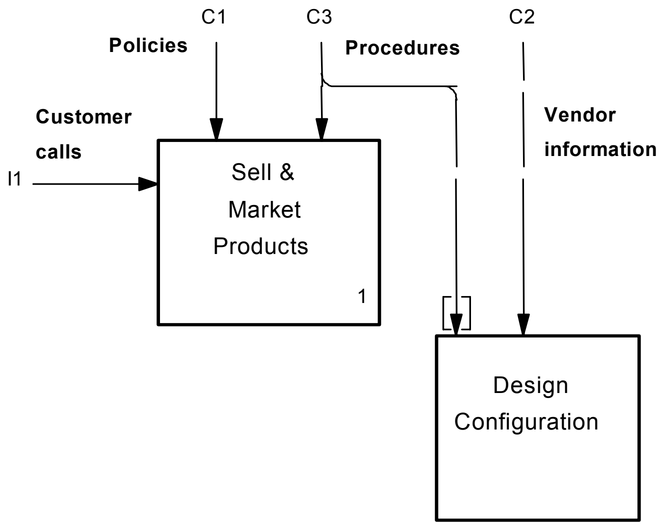
The prefix is repeated for each activity. The numbers are used to represent how detailed the activity is. The A0 activity is decomposed into A1, A2, A3, and so on. A1 is decomposed into A11, A12, A13, and so on. A11 is decomposed into A111, A112, A113, and so on. At each level of decomposition, another sequential digit is added. The only exception to this format is the first level, where A0 is not decomposed into A01, A02, and so on, but into A1, A2, and so on.

Relationship of Diagram to Parent Activity (Boundary and Hierarchy)

An activity is decomposed if it is necessary to document it in more detail. When decomposing an activity, think about its life cycle. This provides a list of candidate child activities. For example, the node *make cookies* can have the life cycle of *gather materials*, *prepare batter*, *bake dough*, and so on.

In IDEF0 function modeling, it is important to realize that the border of a child diagram is the border of the parent activity. There is no difference. This has an important consequence. All work is performed in the leaf (lowest-level) activities. Unlike hierarchy as defined in structured programming, the higher-level activities are *not* controllers of the child activities. The children *are* the parent, only shown in more detail. Activities performed by the CEO can show up next to activities performed by line workers.

ICOM codes are placed at the end of *border arrows* in child diagrams to indicate where the corresponding arrow is on the parent diagram. This is illustrated in the following figure. They serve as a consistency check and can be helpful when the order of arrows in the child differs from that in the parent. An ICOM code consists of a letter *I*, *C*, *O*, or *M*, and a number indicating the top-down or left-right placement of this arrow on the parent activity box, such as I1, C1, O1, and M1.



Breadth-first Compared to Depth-first Modeling

You can build models either *breadth-first*, in which each diagram is detailed as much as possible before decomposing, or *depth-first*, in which the activity hierarchy is first identified and then the arrows are created to connect them. You can use both techniques during the construction of a single model. An activity hierarchy can change slightly as the arrows are drawn because drawing the arrows identifies some new insights into the structure.

Create an IDEFO Model

Create an IDEF0 model to describe and document business processes.

To create an IDEF0 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 Business Process (IDEF0) 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

   Displays the name of the model author

   Author initials

   Displays the initials of the model author

   Apply CRUD/IRUN restrictions

   Enforces CRUD and IRUN restrictions on Call arrows and Mechanism arrows. Select the check box 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 the context activity.

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

Add More Activities to an IDEF0

When you create an IDEF0, 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 IDEF0 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 48)

Insert a Text Block into an IDEF0 Diagram

You can insert a text block into a diagram to display important information.

To insert a text block

1. Click the Text Block tool T and then click the diagram drawing area where you want the text to appear.

   The Text Block Properties dialog opens.

2. Select the Normal option, enter the text as needed in the Text tab, define the text font and color in the Font and Color tabs, and then click OK.

   The Text Block Properties dialog closes and the text you entered is displayed in the diagram drawing area.

Edit Model Properties

You can view and edit the various properties of your model as needed during the modeling cycle.

To edit model properties

1. Click Model Properties from the Model menu.

   The Model Properties dialog opens.

2. Click a tab for the property that you want to edit, edit the following properties as needed, and then click OK:

   General

   Displays the general model properties such as model name, project name, and author name.

   Purpose

   Displays the model purpose and viewpoint.

   Definition

   Displays the model definition and scope.

   Source

   Displays the model source information.

   Status

   Displays the current status of the model.

   Numbering

   Displays numbering options for activities, next number, and diagram.

   Display

   Displays options for various model objects, ABC units, and off-page references.

   Layout

   Displays layout options such as the ability to move or resize diagram objects, arrow spacing, and text wrapping in boxes.

   ABC Units

   Displays ABC options such as currency, time units, and symbol placement.

   Page Setup

   Displays custom header and footer text, sheet size, and units of measurement.

Header/Footer

Displays header and footer text.

Shapes

Displays the default shapes for diagram objects.

Draw Style

Displays the diagram style and organization chart box style.

The properties are edited and the Model Properties dialog closes.

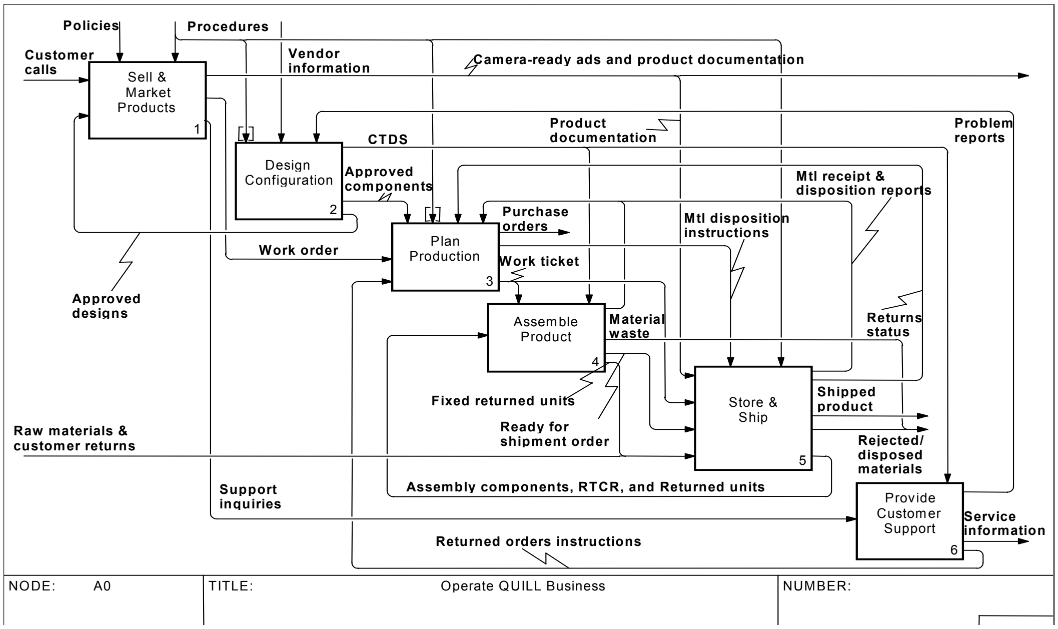
Completion of the Business Process Modeling Project

The IDEF0 diagram's purpose should indicate the kinds of questions that the IDEF0 model will answer. When these questions can be answered by the model, then the model has fulfilled its purpose and is considered complete. When developing the first-level decomposition, ensure that the activities appearing on the diagram are within the model's scope. Before decomposing an activity, determine whether the activity is covered in sufficient depth to fulfill the model's purpose. In addition, IDEF0 should only proceed to the point where precedence (output-input) arrows dominate the diagram. If necessary, you can then use IDEF3 models to model the detailed processes.

Note: For more information on IDEF3 modeling, see the AllFusion Process Modeler *Process Flow Modeling Design Guide*.

Example: IDEF0 Diagram

The following illustration is an example of a typical IDEF0 diagram:



Chapter 4: Activity Decomposition Diagrams

This section contains the following topics:

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.

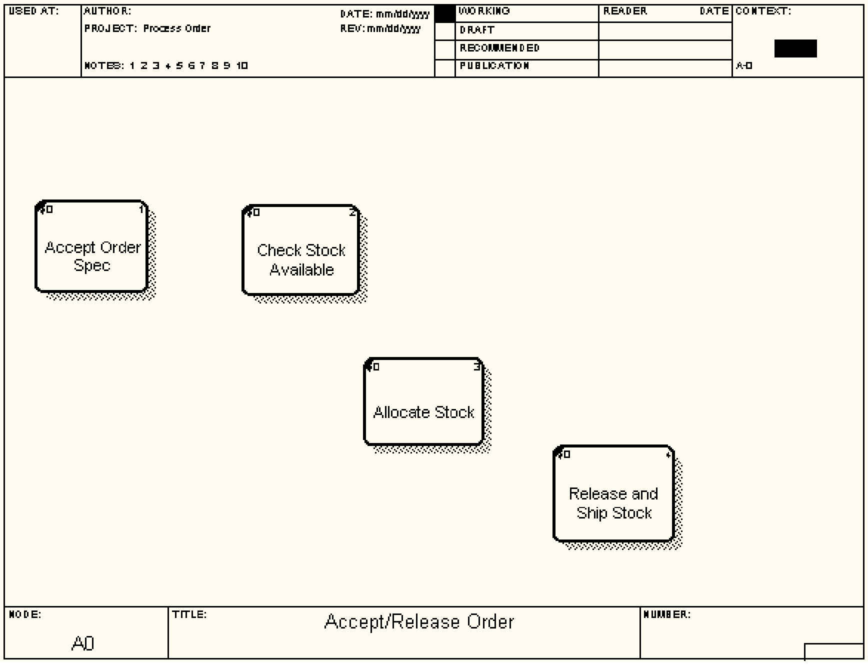
4. Enter a name for the activity or select an existing name from the Unused Activity Names list, and click OK.

   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:



USED AT:

AUTHOR:

PROJECT: Process Order

NOTES: 1 2 3 4 5 6 7 8 9 10

DATE: mm/dd/yyyy

REV: mm/dd/yyyy

NODE: A0

TITLE: Accept/Release Order

NUMBER:

Add an Arrow

Add an arrow to a diagram to show a connection to an activity.

To add an arrow, click the Arrow Tool button ➔ , click the source border or activity, release the mouse button, and click the destination border highlight or activity highlight. The arrow is added to the diagram.

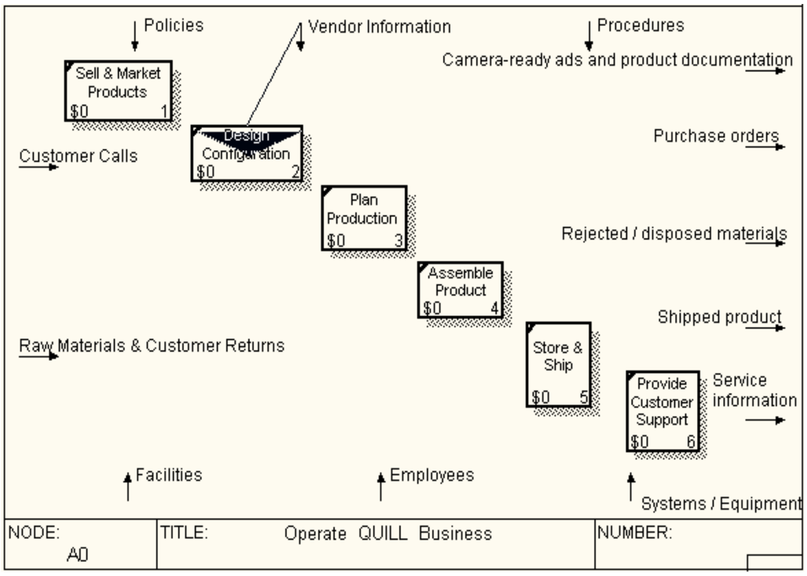
Connect an Existing Arrow to an Activity Box

When you are creating a diagram you can establish a relationship between an existing arrow and an activity when you connect them.

To connect an existing arrow object to an activity

1. Click the arrow and move the cursor over to the activity box to which you want to connect.

   A large highlight triangle appears in the activity box, similar to what is shown in the following illustration.



NODE: A0

TITLE: Operate QUILL Business

NUMBER:

2. Click the mouse again.

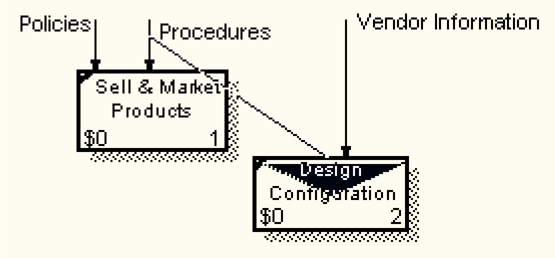
   The arrow is connected to the Activity Box.

Note: The side of the activity box in which a highlight triangle appears depends on the arrow type. For example, if the arrow represents a control, then it must enter an activity box on the top side of the box. If the arrow represents an input, then it must enter the activity box on the left side of the box.

Connect an Arrow Object to Multiple Activities

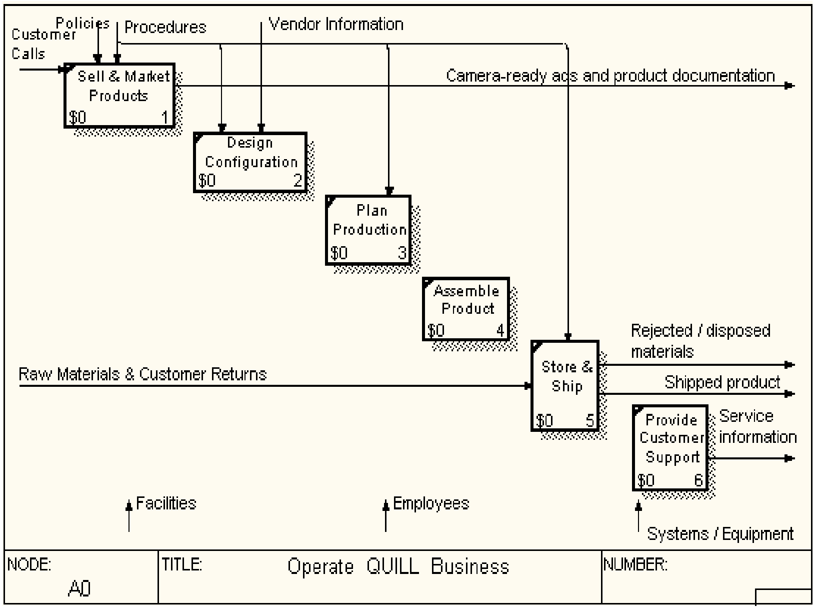
If an arrow can be related to multiple activities, you can connect that arrow to those activities by branching.

To connect an arrow object to multiple activities, click the Arrow Tool button ➔ , click the arrow that you want to branch, click the arrow again, drag it to the second activity box, and release the mouse button. The arrow is branched (connected) to the second activity box, as shown in the following illustration.



Note: You can connect an arrow object to as many activity boxes as you need by repeating this procedure for each activity box to which you need to connect.

Example: Arrow Connected to Multiple Activities



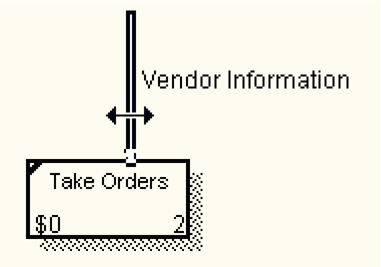
Move Arrow Objects to any Location

You can move the arrow object to any location within the diagram, constrained by its connections.

To move an arrow object

1. Place the cursor arrow over the arrow segment that you want to move.

   A horizontal double-headed arrow appears similar to what is shown in the following illustration.



2. Click and drag the arrow to the location that you want and release the mouse button.

   The object is moved.

Add a Squiggle

You can add a squiggle to a diagram to provide more room between an arrow and its label. Adding a squiggle makes it possible to move the arrow label further away from the arrow, which is helpful if you need to expand the arrow label description.

To add a squiggle, right-click the arrow label to which you want to add a squiggle and click Squiggle from the shortcut menu. A squiggle appears on the diagram connecting the label with its associated arrow object.

Remove Inherited Arrows

You can remove any inherited arrow objects that will not be represented in your decomposition diagram.

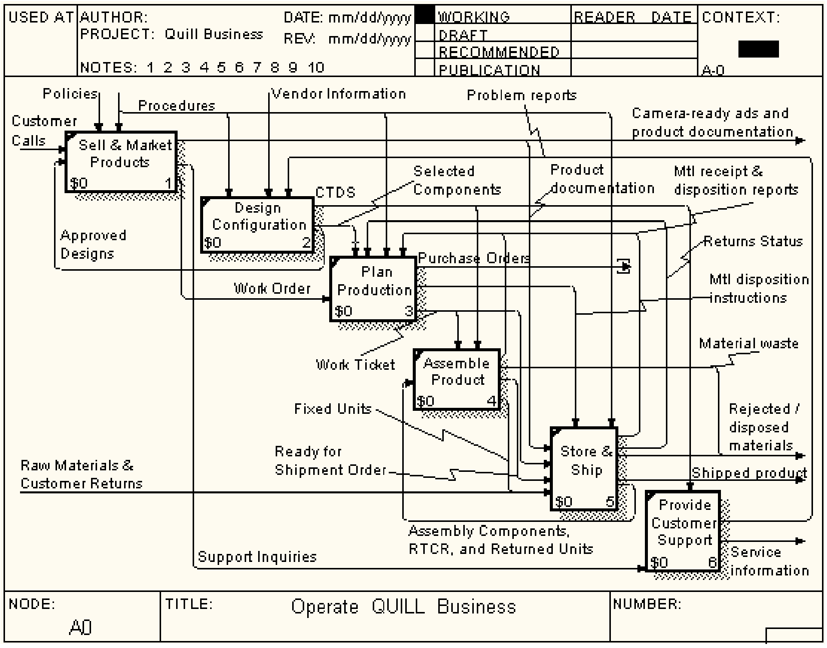
To remove inherited arrows

1. Press the Ctrl key and click each arrow object that you want to remove and then press the Delete key.

   The selected arrows appear with a tunnel in the parent diagram, which means that they are unresolved in the decomposition diagram.

Example: Activity Decomposition Diagram

The following illustrates a typical activity decomposition diagram:



Chapter 5: Additional IDEF0 Features

This section contains the following topics:

Additional IDEF0 Diagrams

In addition to the context and decomposition diagrams, other IDEF0 diagrams can assist in the development and presentation of the model.

IDEF0 also uses Node Tree diagrams and For Exposition Only (FEO) diagrams. After you have created a context diagram and have decomposed your activities to the level of detail you need, you can use Node Tree and FEO diagrams to add another dimension to your model.

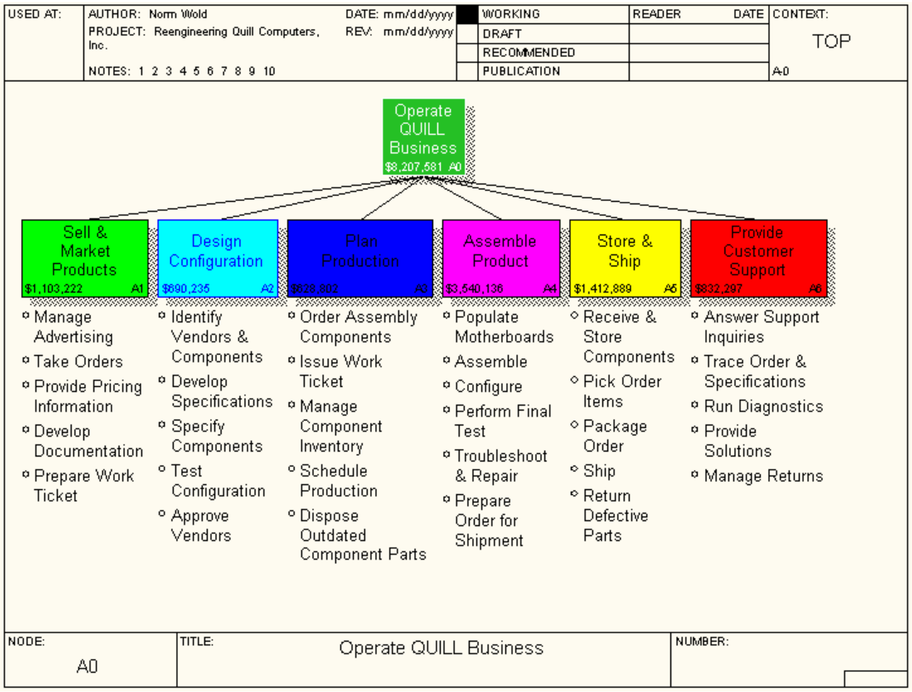
Node Tree Diagrams

Node Tree diagrams are used to show all parent-child activity relationships in a single diagram. You can also create a Node Tree diagram of a section of a model by using a child decomposition as the top node in the diagram. Node Tree diagrams use a traditional tree hierarchy where the top node (box) corresponds to the context diagram activity (process), and the lower level nodes correspond to child decompositions (component activities). However, you can designate any activity as the top node, with its children composing the rest of the tree. Each node contains the name of the process it represents and also a label that consists of a letter followed by one or more numbers. The nodes also retain the properties of the corresponding activities in the model.

Due to the highly iterative nature of activity modeling in general, you should expect to rework the Node Tree diagram numerous times before a stable version is created. Remember that the purpose, scope, and viewpoint statements of the context diagram are available as anchors. Without having those statements as a reference, the Node Tree diagram can never reach a stable state. Viewing your model as a Node Tree helps you concentrate on the functional decomposition of the model, without regard for the flows that connect them.

Example: Node Tree Diagram

The following illustration is an example of a completed Node Tree diagram:



Create a Node Tree Diagram

You can create a Node Tree diagram to show all parent-child activity relationships in a single diagram.

To create a Node Tree diagram

1. Click Add Node Tree from the Diagram menu.

   The Node Tree Wizard - Step 1 of 2 dialog opens.

2. Complete the following fields:

   Node Tree Name

   Specifies the name of the Node Tree diagram.

   You can either enter a new Node Tree diagram name, or keep the default Node Tree name. The default Node Tree diagram name derives from the name of the top-level node in the model. The Node Tree diagram name displays in the TITLE box in the diagram title area.

   Top level activity

   Specifies the activity to be the top-level activity in the Node Tree diagram.

   You can select any activity to be the top-level activity in a Node Tree diagram.

   Number of levels

   Specifies the number of levels in the Node Tree diagram.

   Each level in the Node Tree diagram represents a level of decomposition.

   Click Next.

   The Node Tree Wizard - Step 2 of 2 dialog opens.

3. Define the various Node Tree diagram style options that you require:

   Drawing

   Specifies the following drawing options:

   Bullet last level

   Specifies to add bullets instead of boxes to the lowest level of the diagram.

   Using bullets on the last level occupies less diagram area than using boxes, especially if the Node Tree diagram has many low-level activities.

   Show node numbers

   Specifies to display activity node numbers in the bottom right corner of each node.

   Show boxes

   Specifies to draw boxes for every node.

Box Size

Specifies the following box size options:

Fit each box to text

Specifies to fit a box around the node text.

Depending on the length of activity names, this option can cause the Node Tree diagram to draw node boxes of varying size.

One size per row

Specifies to draw all node boxes on the same row to be the same size as the largest box in that row.

All one size

Specifies to draw all node boxes to be the same size as the largest box on the diagram.

The largest box is usually the activity with the longest name.

Border

Specifies the following border options:

Include Kit

Specifies to include the border kit at the top of the Node Tree diagram.

Include title

Specifies to include the border diagram title at the bottom of the Node Tree diagram.

The border diagram title area also includes the node number, C-Number, and page number.

Connection Style

Specifies the following connection style options:

Diagonal lines

Specifies to draw diagonal lines to connect diagram nodes.

Orthogonal lines

Specifies to draw orthogonal lines to connect diagram nodes.

Click Finish.

The Node Tree Wizard closes and the Node Tree diagram displays in the diagram area using the diagram properties you defined in the wizard.

For Exposition Only Diagrams

A For Exposition Only (FEO) diagram is a graphic representation of specific facts about an IDEF0 diagram. They can be used to test a theory, illustrate different scenarios, show different viewpoints, or highlight other functional details that require special attention, without affecting the original model diagram. For example, you may want to show only two activities or omit certain mechanisms to provide a picture of what could be. Unlike IDEF0 diagrams, FEO diagrams do not need to comply with IDEF0 modeling rules. They are useful for testing how different scenarios or theories might impact a given process. You can add any number of FEO diagrams to the original model diagram, and you can add a FEO diagram at any time.

A FEO diagram looks the same as the original diagram except for the name that you assign to the FEO diagram, and the FEO node reference name. For example, if you add a FEO diagram to a model diagram with the node reference name of A1.3, the corresponding FEO diagram node reference name becomes A1.3F. You can view the FEO diagram name and the FEO node reference name in the border title when the diagram is open.

You can create a FEO diagram with a single activity and the arrows which connect to its border to isolate an activity from its siblings, much like a context diagram. This can be useful in situations in which the facilitator is trying to collect information about the interface (arrows) of the function and the decomposition diagram is too cluttered to easily accommodate changes.

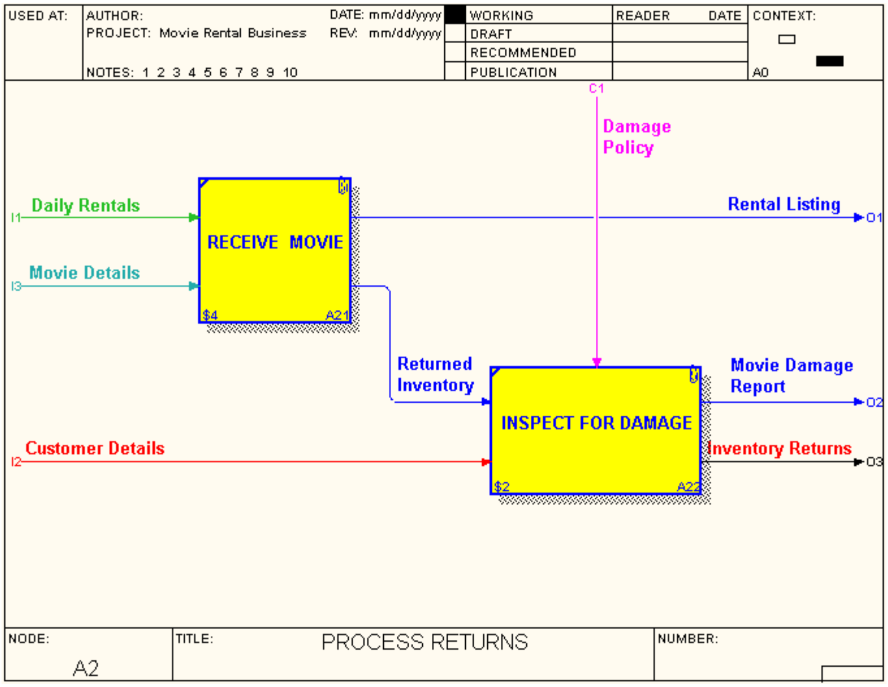
The following describes other common FEO diagrams:

- Alternate context diagrams for a child activity. This type of FEO diagram can be used to explore the potential impact of changes in the environment (context) on an activity.
- A copy of a diagram that includes all of the activities, but only the arrows that connect to a chosen activity. This type of FEO diagram highlights the chosen activity's interaction with the other activities in the diagram.
- A copy of a diagram that includes all of the activities, but only the arrows that directly represent the parent activity's main output. This type of FEO diagram highlights how the main inputs transform into the main outputs.
- Different viewpoints one level deep. This type of FEO diagram can be used to explore how important stakeholders view the system being modeled.
- A copy of a diagram that includes some or all of the activities, and only those arrows needed to highlight a particular point.

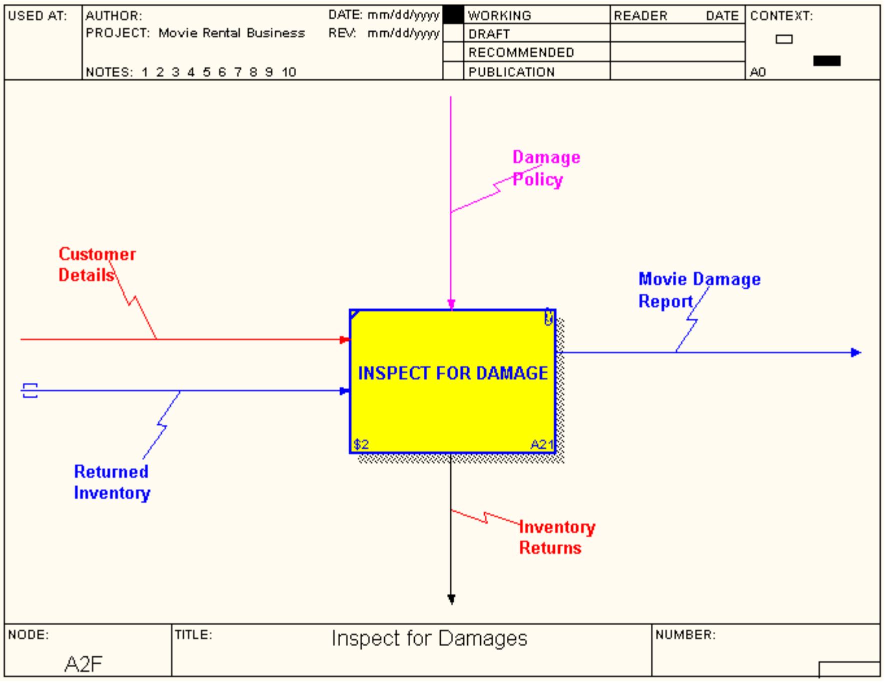
Example: FEO Diagram

The following illustration is a diagram named PROCESS RETURNS. The second illustration is an example of a FEO diagram, which is based upon the original diagram shown in the first illustration.

Example: Original *PROCESS RETURNS* Diagram



Example: FEO Diagram Based on the *PROCESS RETURNS* Diagram Called *Inspect for Damages*



USED AT: AUTHOR: PROJECT: Movie Rental Business DATE: mm/dd/yyyy REV: mm/dd/yyyy NOTES: 1 2 3 4 5 6 7 8 9 10

WORKING DRAFT RECOMMENDED PUBLICATION READER DATE CONTEXT: A0

NODE: A2F TITLE: Inspect for Damages NUMBER:

Create a FEO Diagram

You can create a FEO diagram to graphically represent specific facts about an IDEF0 diagram.

To create a FEO diagram

1. Click Add FEO diagram from the Diagram menu.

   The Add New FEO Diagram dialog opens.

2. Complete the following fields:

   Name of new diagram

   Specifies the name of the FEO diagram.

   FEO Of

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

   Select one of the following options:

   Context

   Specifies to use the model context diagram.

   Select a context diagram from the Source Diagram Name list.

   Decomposition

   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 FEO diagram.

   Copy contents of source diagram

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

   Click OK.

   The Add New FEO Diagram dialog closes and the FEO diagram displays in the diagram area.

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.

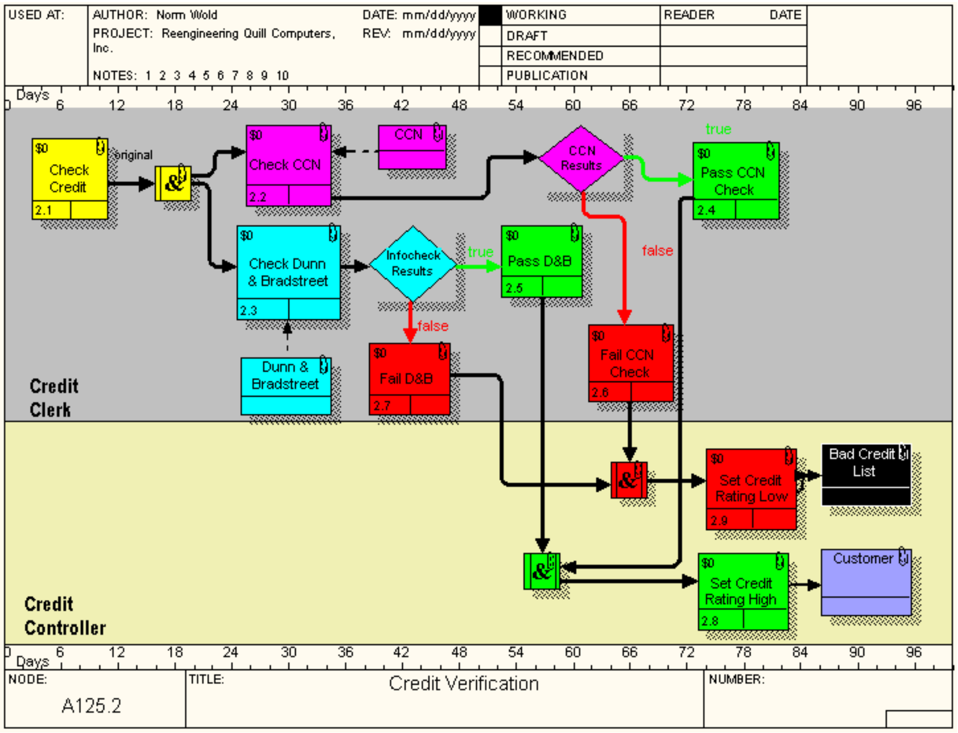
Swim Lane Diagrams

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:



USED AT:

AUTHOR: Norm Wold
PROJECT: Reengineering Quill Computers, Inc.
NOTES: 1 2 3 4 5 6 7 8 9 10

DATE: mm/dd/yyyy
REV: mm/dd/yyyy

WORKING
DRAFT
RECOMMENDED
PUBLICATION

READER DATE

NODE: A125.2
TITLE: Credit Verification
NUMBER:

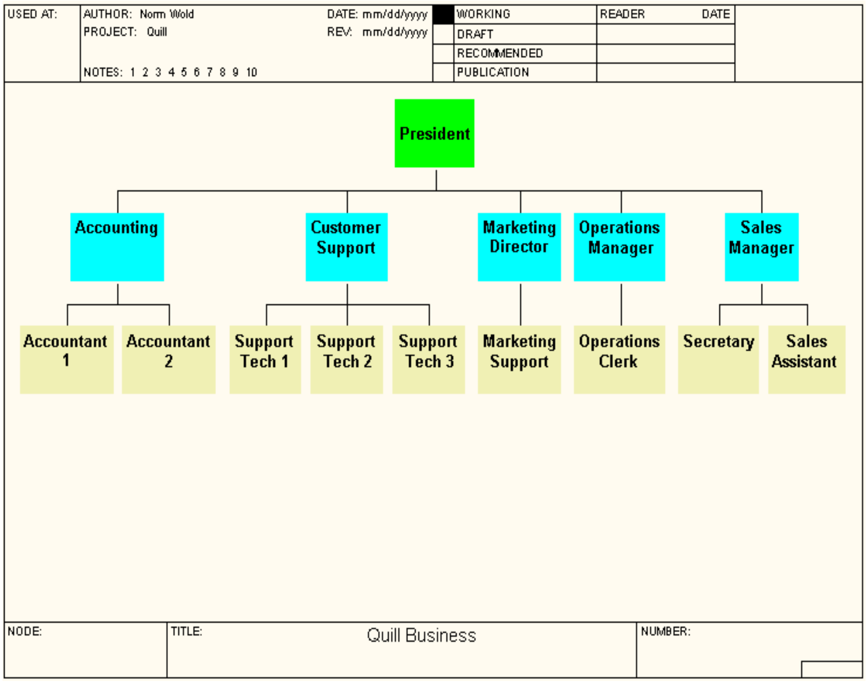
Organization Charts

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:



USED AT:
AUTHOR: Norm Wold
PROJECT: Quill
NOTES: 1 2 3 4 5 6 7 8 9 10
DATE: mm/dd/yyyy
REV: mm/dd/yyyy

		READER	DATE
■	WORKING		
□	DRAFT		
□	RECOMMENDED		
□	PUBLICATION		

NODE:
TITLE: Quill Business
NUMBER:

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.

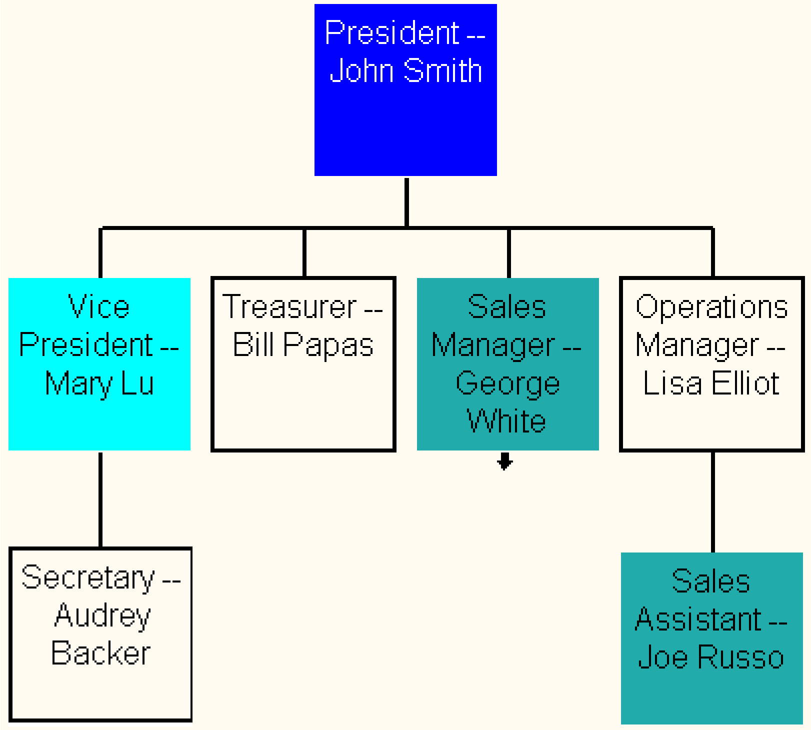
2. Open the organization chart that you need to edit, single-click each object, drag it to its new location in the organization chart, and then release the mouse button.

   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.

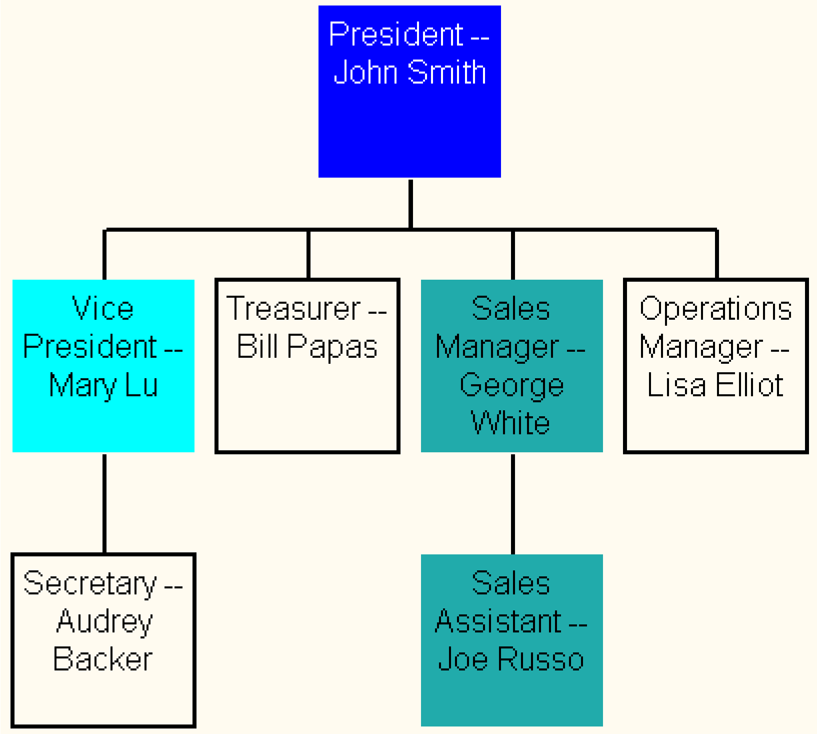
Example: Organization Chart Object Movement

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*:


President -- John Smith
Vice President -- Mary Lu
Treasurer -- Bill Papas
Sales Manager -- George White
Operations Manager -- Lisa Elliot
Secretary -- Audrey Backer
Sales Assistant -- Joe Russo

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:


President -- John Smith
Vice President -- Mary Lu
Treasurer -- Bill Papas
Sales Manager -- George White
Operations Manager -- Lisa Elliot
Secretary -- Audrey Backer
Sales Assistant -- Joe Russo

Activity-Based Costing

To more fully understand underlying production costs, organizations are turning to *Activity-Based Costing* (ABC). This is a technique that captures and analyzes activity costs. ABC captures the costs of resources (such as materials or labor), assigns these expenses to various activities, and then allocates activities to various system outputs called cost objects. Compared to traditional cost accounting, which systematically under-costs low-volume products and over-costs high-volume ones, ABC provides a more exact calculation of the cost to produce a specific product based on the cost to perform all of the activities involved in creating it.

Because it is activity-based, the design of an ABC system mirrors your company's operations.

Costs Allocation

Costs allocation is how the ABC technique is used to accrue costs into cost centers. You need to have the ability to allocate costs throughout a model where they actually occur. Make sure that your model is complete and stable before you begin to allocate costs.

How You Allocate Costs

You need to perform the following tasks to complete the costs allocation process:

- Set the units of measurement
- Define the cost centers
- Enter the cost information

Set Units of Measurement

In order to measure costs, you must set the unit of measurement that you will use to do this. First decide what currency you will use to measure cost (usually US dollars) and how it should be formatted for display and in reports (such as with cents). You also need to specify the unit of time you will use (such as minutes, hours, and so on). These values are global for the model.

To set units of measurement

1. Click Model Properties from the Model menu.

   The Model Properties dialog opens.

2. Click the ABC Units tab, and select from the following options as needed:

   Currency description

   Displays the unit of currency.

   Select the unit of currency appropriate for your environment from the dropdown list.

   Symbol placement

   Displays the placement of the currency symbol.

   Symbol

   Displays the currency symbol.

Number of decimals in diagrams

Displays the number of decimal places used in the diagram activity costs.

Select a number from 0-9.

Number of decimals in reports

Displays the number of decimal places used in activity cost reports.

Select a number from 0-9.

Time Unit

Displays the unit of time.

Decimals in frequency values

Displays the number of decimal places used for frequency values.

Select a number from 0-9.

Decimals in duration values

Displays the number of decimal places used for duration values.

Select a number from 0-9.

Click OK.

The units of measurements settings are globally set for the model and the Model Properties dialog closes.

Define Cost Centers

Cost centers are categories of costs that are shared across all activities. Some examples might include Marketing and Advertising, Purchasing, and Technical Support. After you have set the units of measurement to use, you will need to add the cost centers.

To define cost centers

1. Click Cost Center Editor from the Model menu.

   The Cost Center Editor opens.

2. Enter a new cost center name in the Cost center name (to be added after selected cost center) box and then click the Add button.

   The new cost center appears in alphabetical order in the Cost centers list.

3. Click the Close button.

   The Cost Center Editor closes.

Establish Cost Information for a Leaf-Level Activity

To apply costing estimates to the model, you must first assess the cost of performing each activity in the model. The values you calculate must then be allocated to one or more of the Cost Centers you previously defined.

To establish cost information for a leaf-level activity

1. Right-click the activity for which you want to establish cost information and then select Costs from the shortcut menu.

   The Activity Properties dialog opens.

2. Click the Costs tab and specify the following for each cost center as appropriate:

   Marks

   Defines a cost value.

   Frequency

   Defines a frequency value.

   Duration

   Defines a duration value.

   Click OK.

   Cost information is established and the Activity Properties dialog closes.

Establish Cost Information for a Non Leaf-Level Activity

To apply costing estimates to the model, you must first assess the cost of performing each activity in the model. The values you calculate must then be allocated to one or more of the Cost Centers you previously defined.

To establish cost information for a non leaf-level activity

1. Right-click the activity for which you want to establish cost information and then select Costs from the shortcut menu.

   The Activity Properties dialog opens.

2. Click the Costs tab and specify the following for each cost center as appropriate:

   Override decompositions

   Overrides the cost totals of child decomposition diagrams. If you select this option you can define a cost value in the Marks column.

   Compute from decompositions

   Computes the cost totals from child decomposition diagrams automatically. If you select this option you can define a value in the Frequency field.

   Note: When you click the Costs tab in a parent activity, the default selection is Compute from decompositions.

   Click OK.

   Cost information is established and the Activity Properties dialog closes.

Index