READING & CRITIQUING AN IDEF$_0$ MODEL

by

William D. Waltman

Adrien Presley

Enterprise Integration Frameworks Group
Automation & Robotics Research Institute
7300 Jack Newell Blvd.
Fort Worth, Texas 76118
(817) 794-5900

July 1993

1. INTRODUCTION

IDEF$_0$ is a modeling tool used to produce a model or structured representation of the functions of a system and of the information and objects which tie those functions together. A system can be any combination of hardware, software, and people. An IDEF$_0$ model consists of diagrams and text pages describing the diagrams. Diagrams are the major components of a model and this document will concentrate on how to read them.

IDEF$_0$ methodology recognizes that successful systems development requires input and validation from the people who will ultimately use the system. The Author/Reader Cycle serves as the mechanism to facilitate communication between systems analysts and users. This is accomplished by distributing *Kits* containing IDEF$_0$ models and supporting documentation to the Reader community for comment. When the Kits have been read and critiqued, they are returned to the Author. This cycle may go on for many iterations.

2. PAPER ORGANIZATION

The overall purpose of this paper is to provide new Readers with a brief overview of how to read IDEF$_0$ models and to constructively comment. We first discuss some of the basic syntax necessary to read an IDEF$_0$ model. Next, the makeup of Kits, which act as the vehicle for information exchange between Author and Reader, are discussed. Finally, The Author/Reader cycle is discussed, with emphasis on the responsibilities of the Reader. **APPENDIX A** provides a definition of words printed in **Bold** throughout the paper.

3. BASIC SYNTAX

Functions are represented by boxes and interfaces are represented by arrows, depicted in Figure 1a.

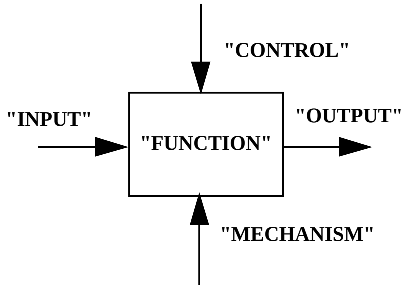

Figure 1a

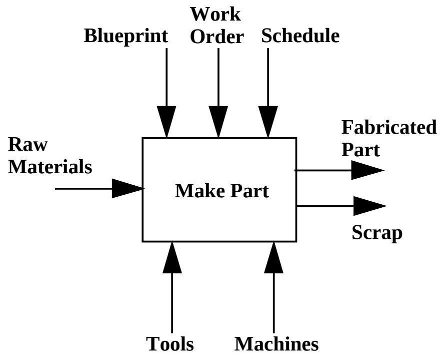

Figure 1b

The boxes represent functions such as activities, actions, processes or operations. Boxes are denoted by an active verb phrase inside the box, such as the "Make Part" box in Figure 1b.

Arrows indicate data. In IDEF, data can be information (like "current status") or physical objects (like "raw materials"). They are named by noun phrases such as "Raw Materials" or "Tools". The position of the arrow indicates the type of information being conveyed.

The arrows entering and leaving the boxes on the left and right represent "**Inputs**" and "**Outputs**", respectively. Inputs represent data needed to perform the function. Outputs show the data that is produced as a result of the function. The function transforms the inputs into the outputs. Arrows which enter from the top indicate "**Controls**", or things which constrain or govern the function. Arrows entering the bottom of the boxes are "**Mechanisms**". Mechanisms can be thought of as the person or device which performs the function.

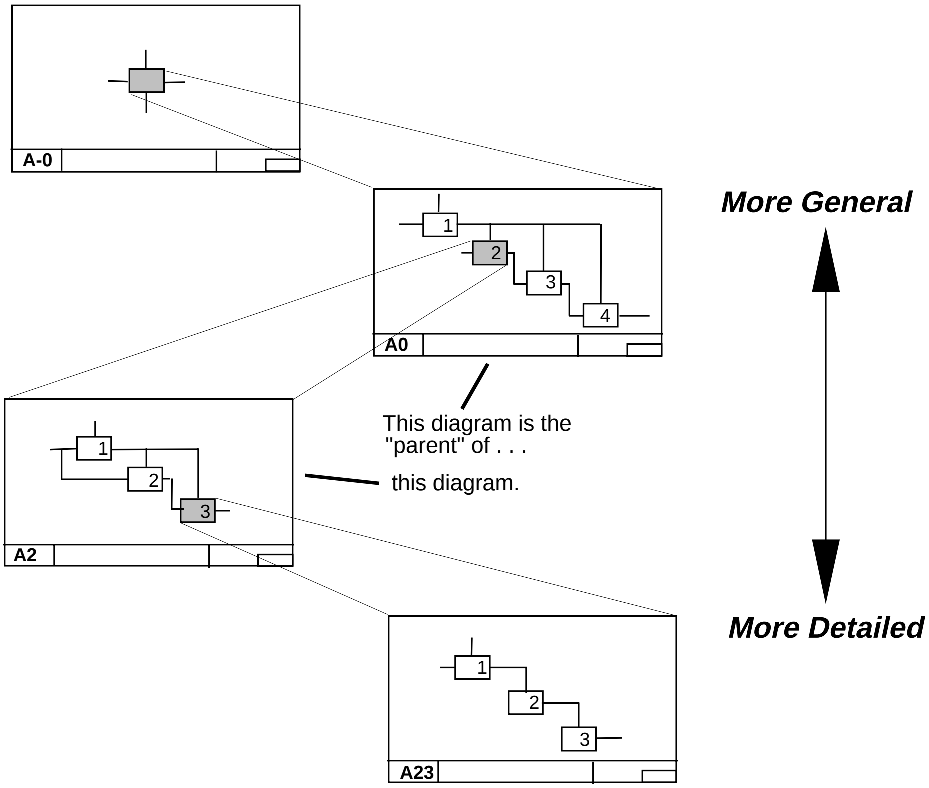

Figure 2: Decomposition Overview

An IDEF model is made up of several diagrams. Each diagram describes in more detail a box from a more general diagram. The process of describing a box in more detail is known as decomposing a function. The more general diagram is called the parent of the detailed diagram. IDEF models are read in a "Top-Down" fashion. The top level diagram, also called the Context or A-0 Diagram, summarizes the overall function of the system which is represented by a single box. The A0 diagram represents the first decomposition of the system. The A0 and all subsequent diagrams must contain 3 to 6 numbered boxes. The numbers help tie the diagrams together. For example, in Figure 2, Box 2 of the A0 is decomposed on diagram A2. Box 3 of A2 is decomposed on diagram A23 and so forth. All diagrams are named beginning with the letter A, representing Activity. Each arrow entering or leaving an upper box must also be shown entering and leaving the lower diagram.

The location of the boxes on a diagram do not necessarily imply sequence or time. Feedback, iteration, continuous processes, and overlapping can be shown by arrows. An output of one box could feed back to a previous box to reactivate that activity. For example, Figure 3 shows a case where the output of the "Inspect Part" box can be a rejected part which is sent back to the previous box as an input for more work.

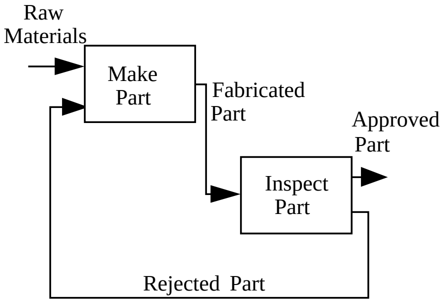

Figure 3

BRANCHES AND JOINS

To gain a complete understanding of a diagram, the boxes and arrows tying them together must be studied. The arrows represent the connections between boxes. Any output arrow may become an input, control, or mechanism to any other box. Arrows may "branch" (indicating they provide data to more than one box), as shown in Figures 4a and 4b.

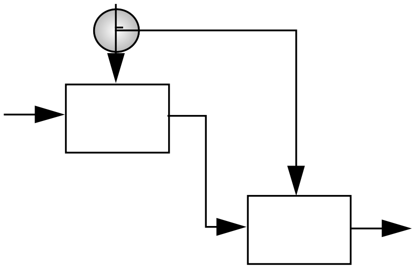

Figure 4a: Arrow Branches

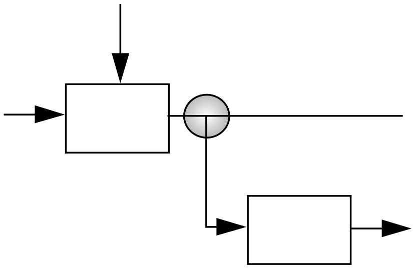

Figure 4b: Arrow Branches

Arrows may also "join" (indicating that output from more than one function can produce output of the same class), as indicated in Figure 5.

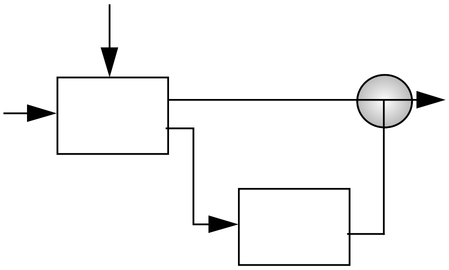

Figure 5: Arrow Join

MULTIPLE INPUTS, CONTROLS, OUTPUTS, AND MECHANISMS

The interpretation of Figure 6a is: In order to produce any subset of the outputs [O1, O2, O3], any subset of the entries [I1, I2, I3, C1, C2, C3, C4, M1, M2, M3] may be required. In the absence of further detailing it cannot be assumed that:

a. any output can be produced without all entries present, or

b. any output requires all entries for its production.

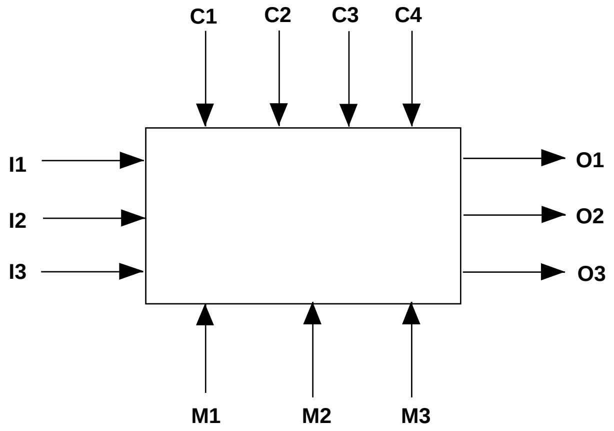

Figure 6a: Illustration of Multiple ICOMs

The partial detailing of Figure 6a as shown in Figure 6b (as it might appear in an A-0 IDEF$_0$ diagram) indicates that I3, C2, C3, C4, M1, M3 are *not* required to produce O1. This indicates that:

a. some form of further detailing will specify the of inputs, controls, and mechanisms to outputs.

b. until that detailing is provided, assumptions should be avoided.

c. reading of the diagrams should concentrate on the arrows, which are explicit, rather than on box contents, which are only implicit.

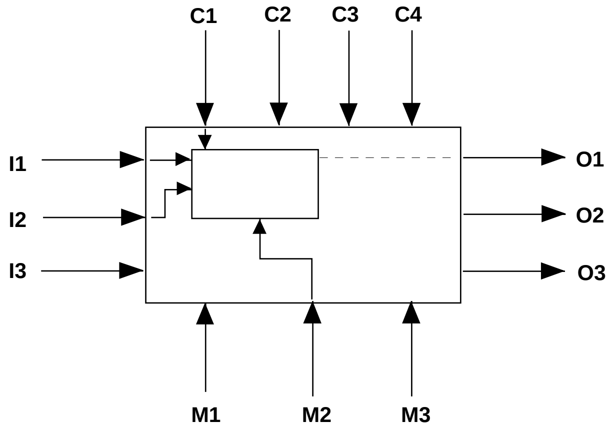

Figure 6b: Decomposing and Detailing Multiple ICOMS

4. IDEF KITS

The IDEF **Kit** is a small collection of related IDEF diagrams and associated material, made by an author for review by one or more readers. Kits are made whenever an author feels there is enough information for review by a Reader Community. Kit size is kept small to allow for quick review and are generally composed of a **Kit Cover Page**, associated diagrams, and supporting material.

KIT COVER PAGE

The Kit Cover Page is a special form that packages together material, describes what the kit contains, and records how it was processed by project members. To accomplish these ends, the Kit Cover Page has several regions and areas. Figure 7 shows an IDEF Kit Cover Page with bold lines defining three separate regions. The top and bottom regions of this form are filled in by the author. The middle region is reserved for the project Librarian.

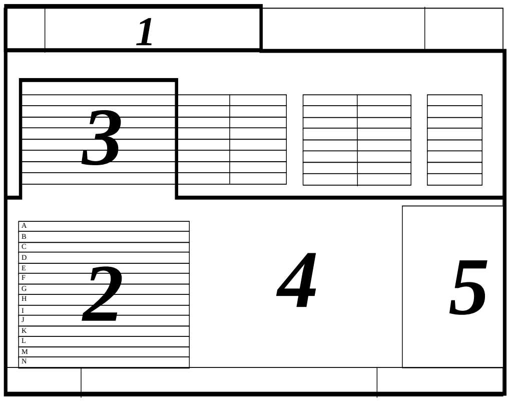

Figure 7: IDEF Cover Page Layout

Figure 7 also has large numbers drawn on it, dividing it into five areas.

The kit's Author includes certain key information in these areas:

Kit Cover Page Area	**Description of Included Information**
1	Identification area: includes **name**, **data**, **project**, **status** and **C-Number**.
2	Information about what the kit contains: **node** and **title** for the kit, and node, title, and C-Number for each page in the kit.
3	Names of kit readers.
4	Used for notes about the entire kit.
5	Reserved for Author to write special instructions to the Librarian.

5. THE AUTHOR/READER CYCLE

The goal of an IDEF analyst is to develop a correct description of the system. IDEF methodology contends that correctness can only come from peer review via the Author/Reader Cycle. Authors create small work packets, called kits, which are distributed to the readers. Readers then write their comments and concerns on these kits and return them to the Author. Authors respond to each of these concerns by also writing their response(s) on the kit and return the kits to the readers. Issues not resolved by the written commenting and reacting process are later addressed in talks.

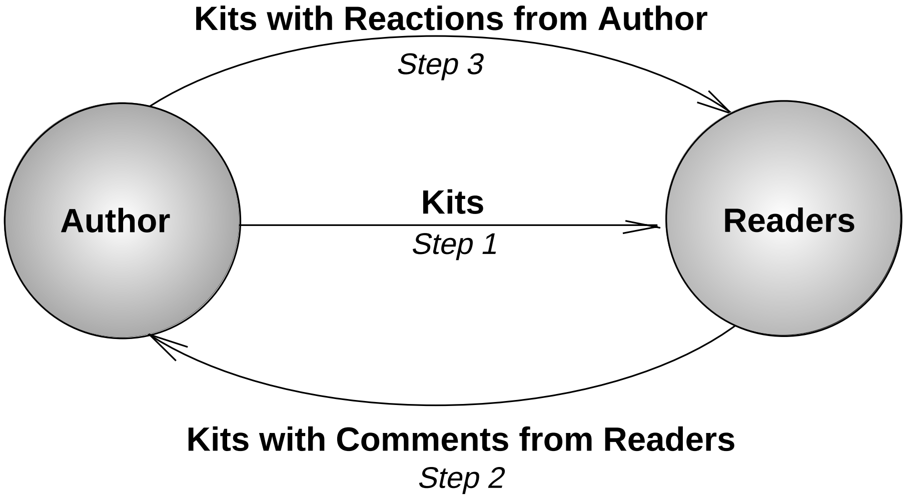
Kits with Reactions from Author
Step 3
Author
Kits
Step 1
Readers
Kits with Comments from Readers
Step 2

Figure 8: The Author/Reader Cycle

THE PROCESS OF READING

Once readers attain skill in reading IDEF diagrams they can systematically understand the messages and the more subtle information contained in the diagrams of the kit step through each diagram to understand its details, and evaluate and validate each diagram.

UNDERSTANDING DETAILS OF A SINGLE DIAGRAM

1. **Read the Title & Node Number** - Start reading a diagram by first scanning the form, and concentrate especially on the title and node number of the diagram.
2. **Read Each Box** - Now read each box separately. Concentrate on understanding a single box in its entirety - that is, what it does (function), what it transforms (I/O), and what constrains its functions (controls). By focusing on just a single box, you can understand how all arrows that touch it are related.
3. **Read the Internal Arrows** - After concentrating on the boxes, now focus on the internal arrows of the diagram. This will uncover details about simple and complex constraints, the predominant data flow path, and various feedback situations.
4. **Read Any Author Notes** - Once you have studied boxes and arrows, read any author's notes. These will be square notes, which usually elaborate an important point or document issues or confusion. Use the notes to gain a better understanding of what the diagram is trying to say, or use them to give specific feedback to the author.
5. **Read Associated Supporting Material** - Read any associated supporting material. Figures, text, and glossary are often attached to a diagram to provide a visual context, elaborate on a key point, or clarify terminology.

UNDERSTANDING THE IMMEDIATE CONTEXT OF THE DIAGRAM

Once you have read the internals of the diagram, focus on their context by determining all the connections between the diagram and its parent. By doing so you will get a clearer understanding of the diagram, because the subject boundary defines how the diagram fits into the overall model. An understanding of the diagram context is acquired by performing the following tasks in a sequential fashion:

1. Read the Parent Box and Its Arrows
2. Read How the Diagram Connects to its Parent
3. Read the Supporting Material of the Parent

UNDERSTANDING HOW A DIAGRAM FITS INTO A MODEL

Once you understand the immediate context of the diagram, focus on how the data fits into the rest of the model. This is done by tracing data flows and constraints from the top of the model down to the diagram under consideration. This will give you a complete understanding of the diagram, because a top-down reading reviews the successive functional details that led to developing the diagram and reveals the way the arrows were decomposed.

CRITICIZING AUTHOR'S MESSAGE

At this point in the reading process, the IDEF reader has acquired an understanding of the diagram, its immediate context, and how the diagram fits into the overall model. *Good IDEF readers know only what is written on paper; they assume nothing.* Thus their acquired understanding comes only from the model and its supporting material.

Now comes the time when the author's message is constructively criticized according to the readers newly acquired understanding. Criticizing means questioning the message told by a diagram.

IDEF readers must ask three basic kinds of questions:

1. Questioning Syntax
2. Questioning Your Understanding of the Message
3. Agreeing with the Author's Message

All of the questions in each of these three groups are explicitly listed in **APPENDIX B** at the end of this paper.

COMMENTING CONSTRUCTIVELY

The documenting of issues as they arise when reading a diagram is called **commenting**. Commenting is a skill, and constructive comments — those that criticize with the goal of improving a diagram — are the only kind readers should make. Comments are written directly on the pages of the kit in **RED**. Simple agreement and disagreement marking conventions make comments easy to write. Numbered notes help keep track of each comment and the sequence in which they were written.

Record Work Times - Record the the start and finish times for reading and commenting on a kit in the Special Instructions area of the Kit Cover Page. As you become familiar with the Author-Reader Cycle, your elapsed times to process a kit will drop considerably.

Checking Frame Contents:

- Check the author's name, project, date, revision, and C-Number to distinguish this diagram from all others.
- Check the context box, title, and node number to see how the diagram fits into the model.
- Check the status area to determine the level of approval this diagram has received.
- Finally, place your initials and the date of your review in the Reader Area.

After reading and commenting on several kits you will find these actions become automatic.

Simple agreement is recorded with a red check mark (√). Similarly, simple disagreement is noted with a red X. These marks tell the author that the diagram or page was read and that the reader generally agrees or disagrees with the message presented. More substantive comments should also be included on each page of the kit.

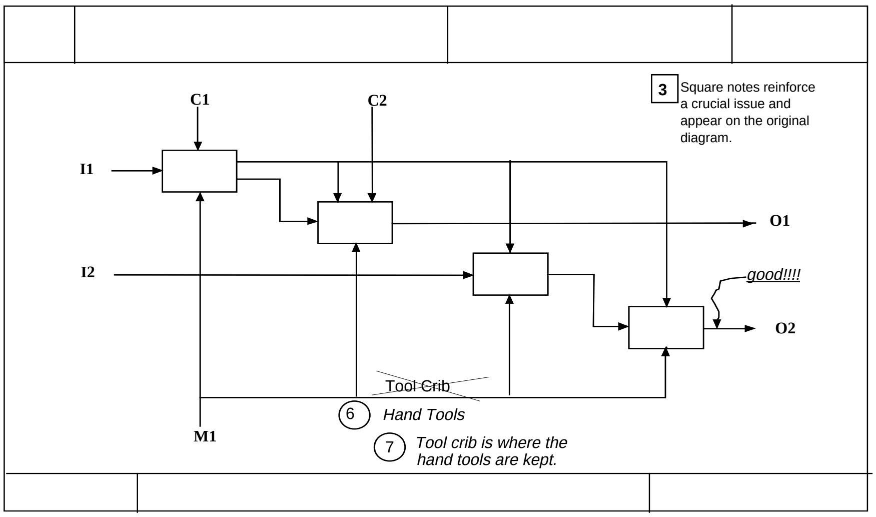
3 Square notes reinforce a crucial issue and appear on the original diagram.
C1
C2
I1
I2
O1
O2
good!!!!
~~Tool Crib~~
6 Hand Tools
7 Tool crib is where the hand tools are kept.
M1

Figure 9: Commenting Examples

Making IDEF Notes - The IDEF methodology distinguishes between "Round" ⓝ notes — those added to diagrams by the reader/commenter to explain disagreement with point(s) the Author has made on the diagram, and "Square" $\boxed{n}$ notes — those included by the Author on the orignial diagram to highlight or explain specific details or issues.

To make a round note ⓝ, first make a tick mark over the next available Note Number on the Kit Cover Page then write that number on the diagram and circle it. Finally, write down your issue next to the circled number. If necessary, connect the note to the appropriate part of the diagram a squiggle. Figure 9 provides examples of round and square notes as well as reinforcing comments.

It is also a good practice to acknowledge agreement on important issues with comments such as "good!!".

Rereading the Kit - After a kit has been commented, it is wise to go back and reread it. Rereading isn't busywork. As with any written document, IDEF first drafts are rarely good enough to publish. So by reviewing your comments, you can mark important notes with a check, cross out irrelevant notes with an X, clarify weak notes, add new notes, and summarize your review on the cover page. Experience has shown that the quality of review is greatly increased if several minutes are taken to reread the entire kit.

SOME FINAL THOUGHTS ON MAKING CONSTRUCTIVE CRITICISM

No one likes to receive one's work that criticicism without suggestions of ways to improve what was done. The IDEF methodology strongly advocates a *positive* and *constructive* commenting style. Constructive comments should be brief, clear, positive and specific.

It is especially important to be helpful when an author makes a large oversight, has misunderstood basic terminology, or has drawn a diagram outside the purpose or viewpoint of the model. In such cases you should make a note on the Kit Cover Page describing the problem as you see it and arrange to meet with the author to discuss the matter.

Good IDEF readers are not satisfied if they see nothing wrong - they must also see everything right. Important points may have been overlooked, parts of the diagram could be irrelevant, or a diagram could even be drawn from the wrong point of view. That is why readers need to point out the good as well as the bad. Authors need to know they are on the right track.

BIBLIOGRAPHY

1. Marca, D.A. and McGowan, C.L., *Structured Analysis and Design Technique*, McGraw-Hill, Inc., 1988

2. *Integrated Computer-Aided Manufacturing (ICAM) Part II - Function Modeling Manual (IDEF0)*, Contract no. AFWAL-TR-81-4023, Vol. IV, June 1981.

3. Blanchard, Benjamin and Wolter, Fabrycky, *Systems Engineering and Analysis*, Printice Hall, 1990

4. Colquhoun, G.J, Baines, R.W, Crossley, Roger, *A State of the Art Review of IDEF0*, International Journal of Computer Integrated Manufacturing, Vol.6, No.4, 1993, pp.252-264.

APPENDIX A

Below is a short description of key terms used in the IDEF$_0$ methodology:

KEY TERM OR FIELD NAME	DESCRIPTION
Author/ Date/ Project	This field appears at the top of each IDEF$_0$ diagram. It tells who created the diagram, the date it was first drawn, and the project title.
C-Number	This field is located at the bottom right of all IDEF$_0$ diagrams. It is typically composed of two or three letters of the author's initials followed by a number sequentially assigned by the author. Every diagram form used by an author receives a unique (sequentially assigned) C-number which uniquely identifies each diagram and to trace the history of each diagram.
Glossary	This is a specialized diagram form used to providedetailed definitions of the *Inputs*, *Controls*, *Outputs*, and *Mechanisms*.
Librarian	The person responsible for routing and tracking of kits and for project files.
Message	This is the predominant field occupying most of the diagram form. It's purpose is to convey the Author's primary message. It is normally used for diagramming, but can be used for other purposes such as glossary, checklist(s), notes, sketches, etc.
Node Number	The number associated with an IDEF$_0$ box or diagram. Each activity may be shown once as a box and once as diagram.
Node List	A listing, often indented, showing all nodes in an IDEF$_0$ model in outline order.
Notes Field	This provides a check-off for round ⓝ notes written on the diagram sheet(s). As comments are made on a page, the notes are successively crossed out. The crossing out provides a quick check for the number of comments, while the circled number provides a unique reference to the specific comment.
Reader / Date	This is the area where the reader will initial and date <u>*each*</u> form.

APPENDIX A (CONTINUED)

KEY TERM OR FIELD NAME — **DESCRIPTION**

Status — The status classification provides a ranking of approval. Categories include:

WORKING - diagram is a major change, regardless of prior status.

DRAFT - minor change from previous diagram.

RECOMMENDED - Both diagram and supporting text have been reviewed and approved by design and customer teams and is not expected to change.

PUBLICATION - This page may be forwarded as-is for publication.

Title — This field is located in the center of the bottom of the diagram form. It contains the name of the material presented on the Diagram Form. If the Message field contains a diagram, then the contents of the Title field must precisely match the name written in the parent box.

Inputs — The arrow class associated with the left hand side of the IDEF$_0$box. Inputs are the things being transformed by the activity of the box.

Controls — The types of arrows associated with the top of an IDEF$_0$ box. Provides guidance to the transformation or controls the activity.

Outputs — The class of arrows associated with the right hand side of the IDEF$_0$ oxes. Outputs are the result of an IDEF0 transformation — the things into which inputs are transformed.

Mechanisms — The arrow class associated with the bottoms of IDEF$_0$ boxes. These are things used or consumed by the activity of the box.

Kit Cover Page — The first page of a standardized package of diagrams which contain portions of models to be reviewed. It contains descriptive information on model as well as documentation of ongoing comments engendered by the Author/Reader Cycle.

APPENDIX B

The following three tables provide a complete list of the kinds of questions IDEF readers must ask when criticizing the author’s message. The types of questions are grouped into three distinct areas:

1. Questioning Syntax
2. Questioning Your Understanding of the Message
3. Agreeing with the Author’s Message

1. QUESTIONING SYNTAX

- Are boxes all numbered correctly?
- Do all boxes have verbal names?
- Are there missing arrowheads?
- Are all labels clearly connected to one arrow?
- Are there extra labels on long arrows?
- Are there unlabeled arrows?
- Are the labels of external arrows compatible with the labels of boundary arrows on the parent?
- Is tunneling (parenthesized arrows) used excessively or incorrectly?

APPENDIX B (CONTINUED)

2. QUESTIONING YOUR UNDERSTANDING OF THE MESSAGE

- What is the role of each box in the diagram?
- How is each box activated?
- Is the role of each box clear?
- How does the box transform its inputs to outputs?
- Is important error handling clear?
- Is the main story-line clear?
- Are alternate data flow paths understood?
- Does the terminology support the story line?
- How do the boxes decompose the parent box?
- What are the source and target of all external arrows?
- Are the main inputs, controls, and outputs clear?
- Are there too many or too few boxes?
- Should the boxes be reordered?
- Is part of the diagram unusually busy or inactive?
- Are there too many arrows?
- Are arrow crossings confusing?
- Are labels too long or too wordy?
- Is jargon used heavily?
- Does the terminology match the viewpoint of the intended audience?

APPENDIX B (CONTINUED)

3. AGREEING WITH THE AUTHOR'S MESSAGE

- Is the decomposition complete?
- Is there a missing box?
- Is there a box that doesn't belong?
- Are there any surprises in the decomposition?
- Would I have come up with a radically different decomposition?
- What questions will this diagram answer?
- Does this match the purpose of the model?
- From whose viewpoint is the system being described?
- Does this viewpoint match the viewpoint of the model?
- Is the diagram too vague or detailed to answer the question set implied by the purpose of the model?
- Does the diagram answer questions outside the purpose of the model?
- Are all terms used from the same viewpoint?
- Are the facts relevant to the viewpoint of the model?
- Does the model reflect reality?
- Are boxes properly placed in order of dominance?
- Are there extraneous or missing arrows between boxes?
- Are titles and labels misleading?
- Do arrow branches contain only data which is required by a box?
- Do "normal" data flow paths seem to work?
- How will wrong data affect a box?
- Are essential error paths accounted for?
- Should a function do more work than is implied by the arrows touching it?