IDEF0 Modeling Notes

Robert E. Young
Department of Industrial Engineering
North Carolina State University
Raleigh, NC 27695

September 2005

Table of Contents

INTRODUCTION AND OVERVIEW OF IDEF0

IDEF0 (Integration DEFinition language 0) is based on SADT™ (Structured Analysis and Design Technique™), developed by Douglas T. Ross and SofTech, Inc. In its original form, IDEF0 includes both a definition of a graphical modeling language (syntax and semantics) and a description of a comprehensive methodology for developing models.

During the 1970s, the U.S. Air Force Program for Integrated Computer Aided Manufacturing (ICAM) sought to increase manufacturing productivity through systematic application of computer technology. The ICAM program identified the need for better analysis and communication techniques for people involved in improving manufacturing productivity. ICAM program personnel recognized the potential of SADT to fulfill this need and purchased the rights to SADT from SofTech, placing the technology in the public domain. It was the first of several modeling methodologies that came from the ICAM program and was renamed IDEF0.

The ICAM program developed a series of techniques known as the IDEF (ICAM Definition) techniques which included the following:

- IDEF0, used to produce a "function model". A function model is a structured representation of the functions, activities or processes within the modeled system or subject area.
- IDEF1, used to produce an "information model". An information model represents the structure and semantics of information within the modeled system or subject area.
- IDEF2, used to produce a "dynamics model". A dynamics model represents the time-varying behavioral characteristics of the modeled system or subject area.

In 1983, the U.S. Air Force Integrated Information Support System program enhanced the IDEF1 information modeling technique to form IDEF1X (IDEF1 Extended), a semantic data modeling technique.

Currently, IDEF0 and IDEF1X techniques are widely used in the government, industrial and commercial sectors, supporting modeling efforts for a wide range of enterprises and application domains. IDEF2 is a discrete event simulation modeling tool built by Pritsker and Associates. It is not used today because it has been subsumed by commercial simulation tools such as SLAM and ARENA.

In 1991 the National Institute of Standards and Technology (NIST) received support from the U.S. Department of Defense, Office of Corporate Information Management (DoD/CIM), to develop one or more Federal Information Processing Standards (FIPS) for modeling techniques. The techniques selected were IDEF0 for function modeling and IDEF1X for information modeling. These FIPS documents are based on the IDEF manuals published by the U.S. Air Force in the early 1980s.

There is also an IEEE Standard for IDEF0 and IDEF1X.

IDEF DOCUMENTATION

IDEF0

- *Integration Definition for Function Modeling (IDEF0)*, Publication FIPS 183, June 1994, National Institute of Standards and Technology (NIST), U.S. Department of Commerce, Gaithersburg, MD, USA.

- *IEEE Standard for Functional Modeling Language – Syntax and Semantics for IDEF0*, IEEE Std 1320.1-1998, Software Engineering Standards Committee of the IEEE Computer Society, IEEE-SA Standards Board, June 1998, The Institute of Electrical and Electronics Engineers, Inc., 345 E. 47th ST, New York, NY, USA, ISBN 0-7381-0340-3.

IDEF1X

- *Integration Definition for Information Modeling (IDEF1X)*, Publication FIPS 184, June 1993, National Institute of Standards and Technology (NIST), U.S. Department of Commerce, Gaithersburg, MD, USA.

- *IEEE Standard for Conceptual Modeling Language – Syntax and Semantics for IDEFiX$_{97}$ (IDEF$_{object}$)*, IEEE Std 1320.2-1998, Software Engineering Standards Committee of the IEEE Computer Society, IEEE-SA Standards Board, June 1998, The Institute of Electrical and Electronics Engineers, Inc., 345 E. 47th ST, New York, NY, USA, ISBN 0-7381-0341-1.

IDEF0 Approach

IDEF0 may be used to model a wide variety of automated and non-automated systems. For new systems, IDEF0 may be used first to define the requirements and specify the functions, and then to design an implementation that meets the requirements and performs the functions. For existing systems, IDEF0 can be used to analyze the functions the system performs and to record the mechanisms (means) by which these are done.

The result of applying IDEF0 to a system is a model that consists of a hierarchical series of diagrams, text, and glossary cross-referenced to each other. The two primary modeling components are functions (represented on a diagram by boxes) and the data and objects that inter-relate those functions (represented by arrows).

IDEF0 APPROACH

CONTINUED ...

As a function modeling language, IDEF0 has the following characteristics:

1. It is comprehensive and expressive, capable of graphically representing a wide variety of business, manufacturing and other types of enterprise operations to any level of detail.

2. It is a coherent and simple language, providing for rigorous and precise expression, and promoting consistency of usage and interpretation.

3. It enhances communication between systems analysts, developers and users through ease of learning and its emphasis on hierarchical exposition of detail.

4. It is well-tested and proven, through many years of use in Air Force and other government development projects, and by private industry.

5. It can be generated by a variety of computer graphics tools; numerous commercial products specifically support development and analysis of IDEF0 diagrams and models.

USING IDEF0 FOR DATABASE SYSTEM DESIGN

IDEF0 is usefull to us because it allows us to define what the system is doing that we will support with our database application. It also defines what activities our form set is supporting and allows us to see how our application works in the context of the overall system.

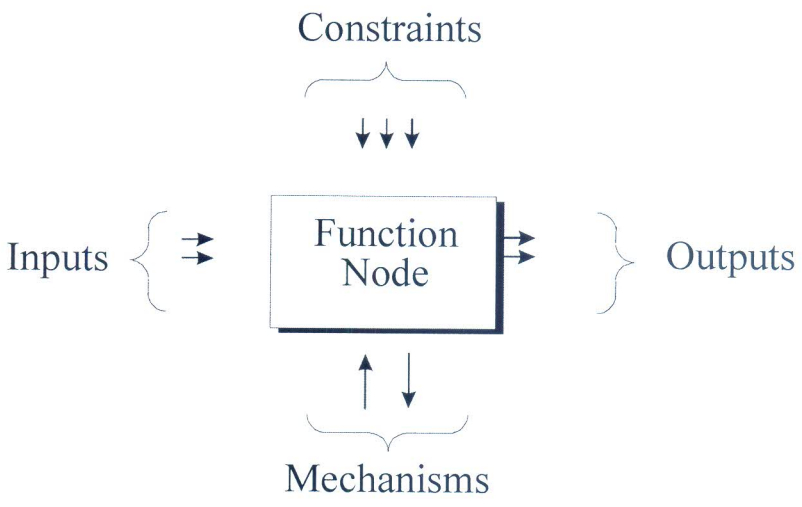
To use IDEF0 in the development of a database system we need to do the following:

1. Create an IDEF0 model of the system from the perspective(s) we are supporting in our database application. These perspectives may be that of a customer, an employee, a manager, etc.

2. Use the *Mechanism Call* (arrows coming into the bottom side of the function box) to identify the forms in our database application that support the activities in the function box.

3. Use the IDEF0 model to "walk" someone through the system operation. When a function block is supported by a form (the function box has an arrow coming into its bottom side) then you can show them how the form supports the activities in the function. In this way, a client gets a sense of what the system is doing and how your application works to support the system's overall needs.

IDEF0 SYNTAX

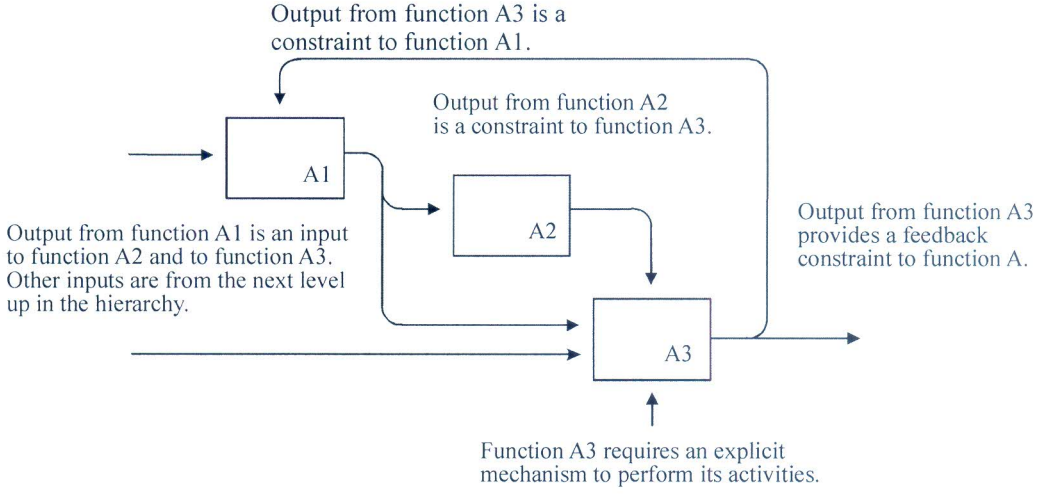


File: FIG2.cdr

- A node is an activity of function that transforms inputs into outputs under constraints using mechanisms.
- The inputs, outputs, constraints, and mechanisms are directed information flows that connect functions in IDEF0 models.
- Often the mechanisms are not shown because they are understood by the audience.
- Sometimes mechanisms arrows point away from the note to indicate that the mechanism is represented by another IDEF0 model or functions in another IDEF0 model.

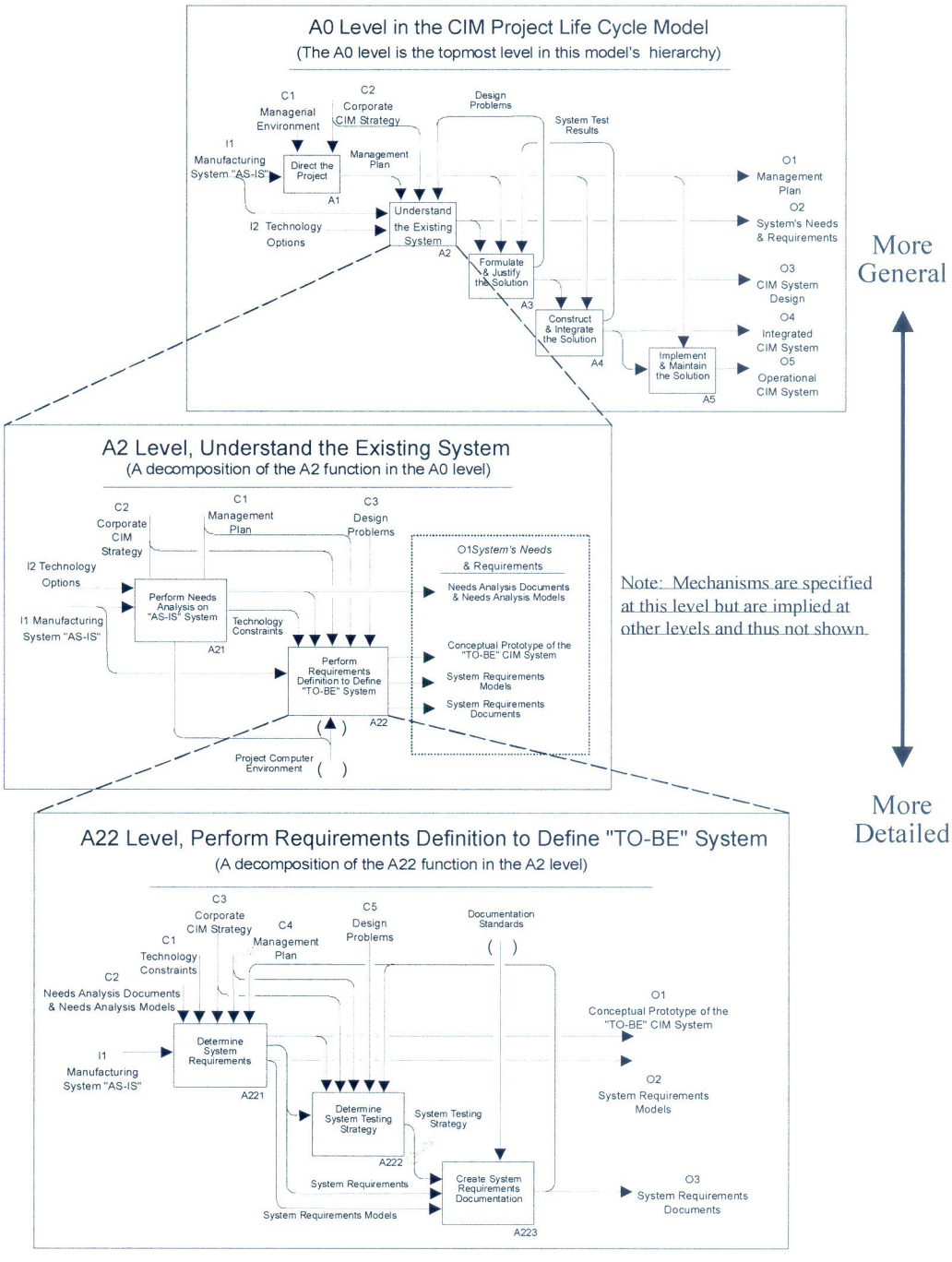
IDEF0 SYNTAX

CONTINUED ...



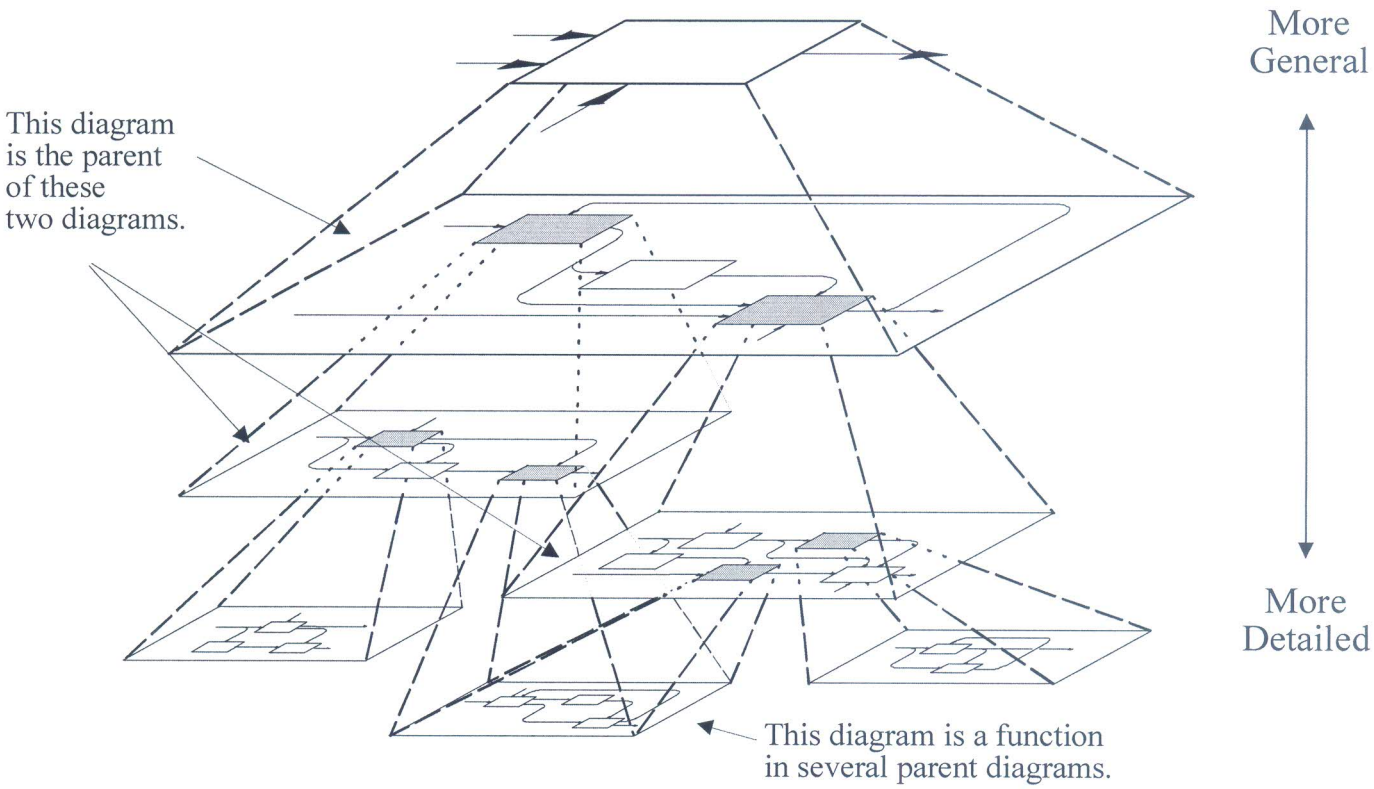
File: FIG3.cdr

USING IDEF0 TO DECOMPOSE ACTIVITIES



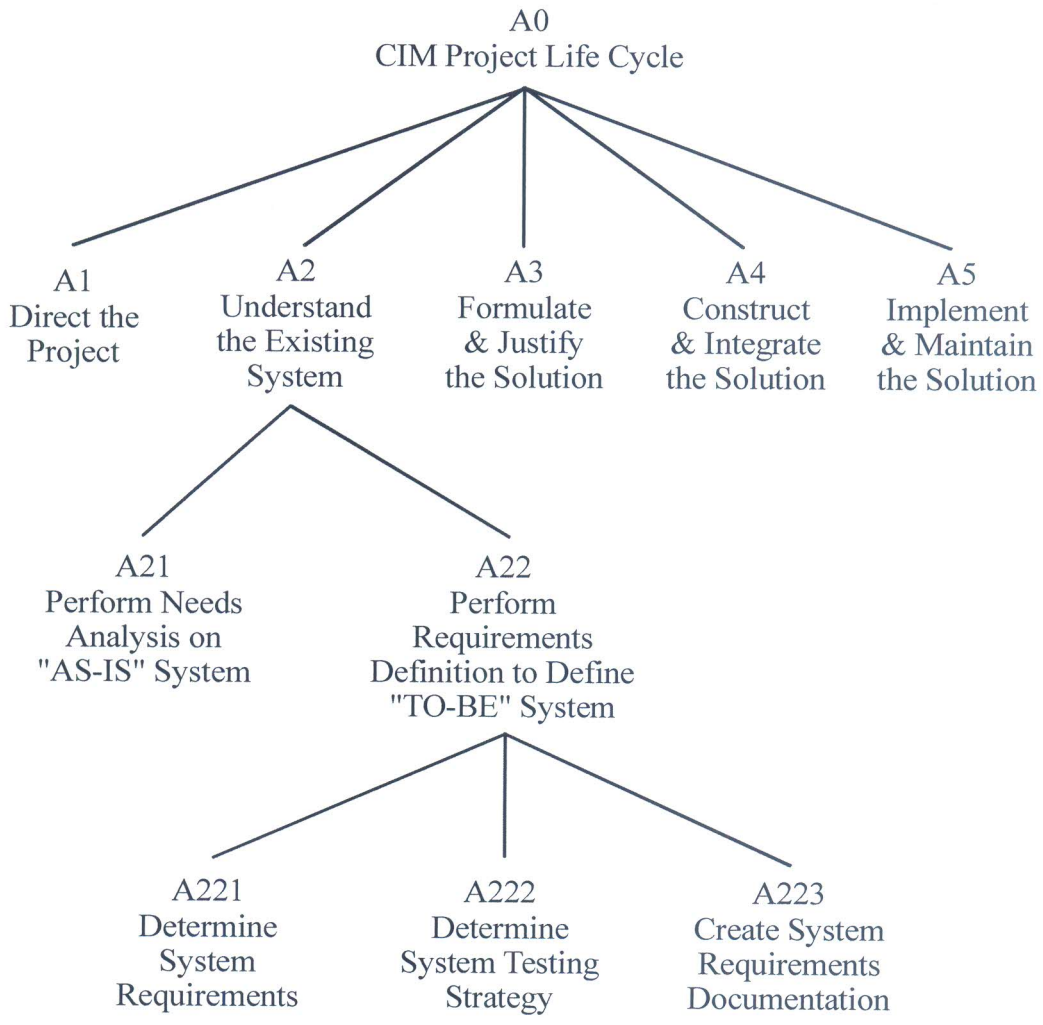
 File: FIG5.cdr

IDEF0 Hierarchical Structure



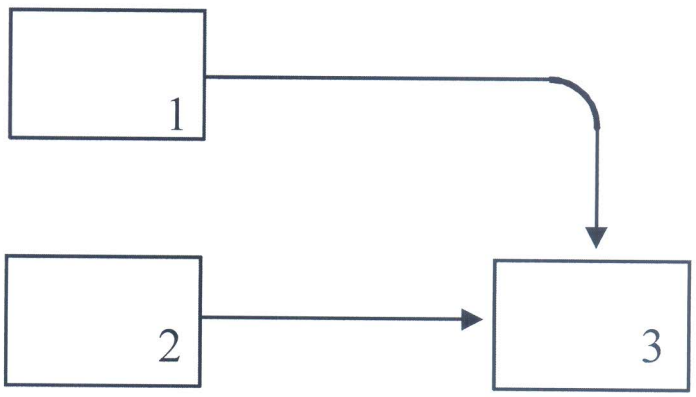
File: FIG4.cdr

NODE TREE DIAGRAM
SHOWING THE IDEF0 MODEL HIERARCHY



 File: FIG6.cdr

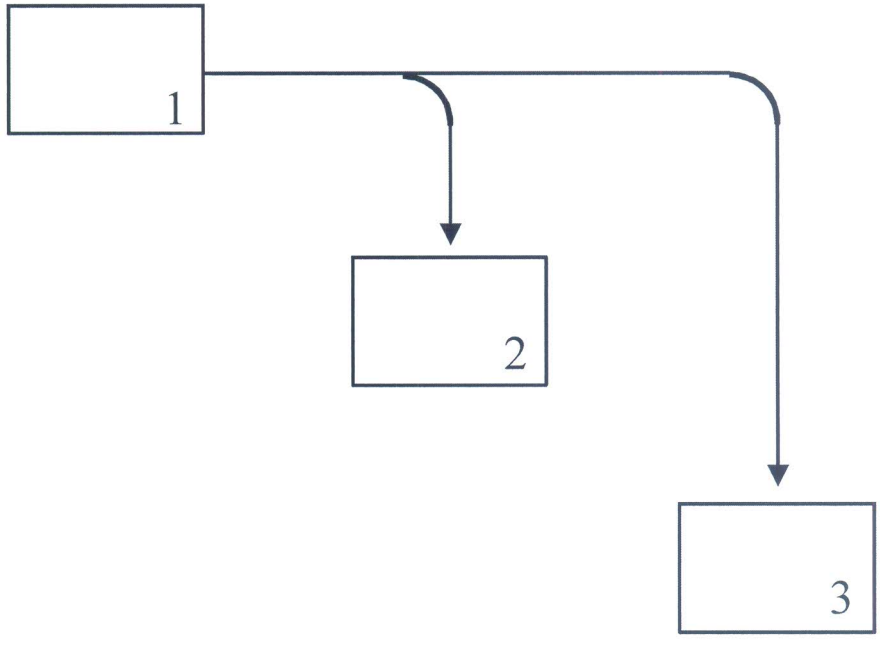
MEANING OF A CONSTRAINT



 File: FIG7.cdr

The box 3 activity requires the data provided as input by box 2. However, the box 3 function cannot be performed until the data is available from box 1. In this way, the output of box 1 *constrains* the activity of box 3.

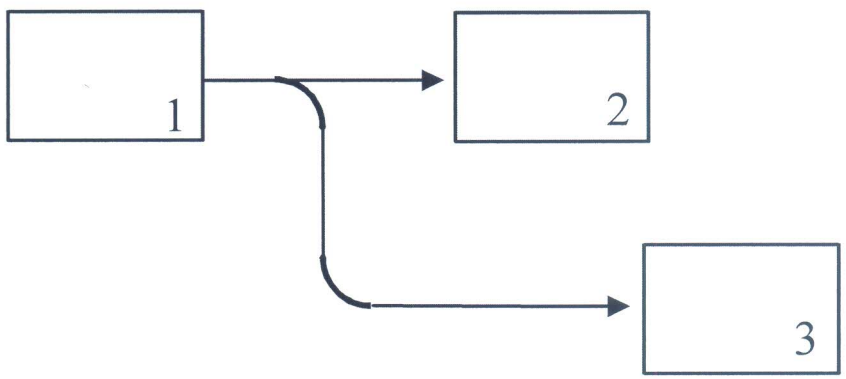
MEANING OF A CONSTRAINT
CONTINUED ...



 File: FIG8.cdr

The activity from box 1 produces data as its output that is needed by box 2 and by box 3 in order for them to perform their functions. . In this way, the output of box 1 *constrains* the activity of box 2 and the activity of box 3.

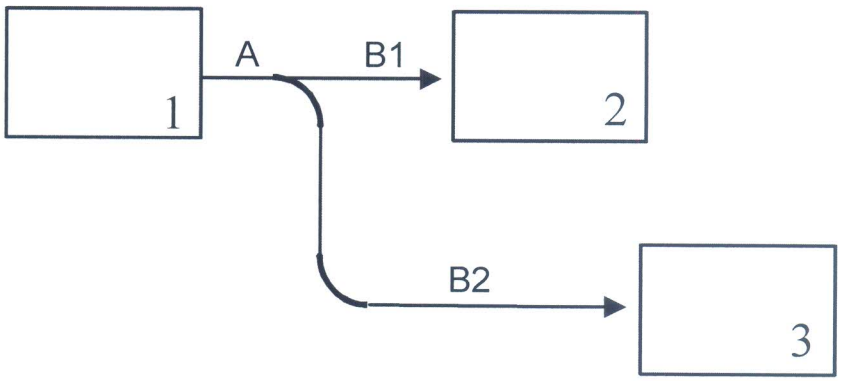
SIMULTANEOUS ACTION WITH THE SAME INFORMATION



File: FIG9.cdr

Once the data is supplied by box 1 the functions in boxes 2 and 3 may operate in parallel. Unless otherwise labeled, the same data flows from box 1 to both box 2 and box 3.

SIMULTANEOUS ACTION
WITH THE DIFFERENT INFORMATION
-- ARC BRANCHING --

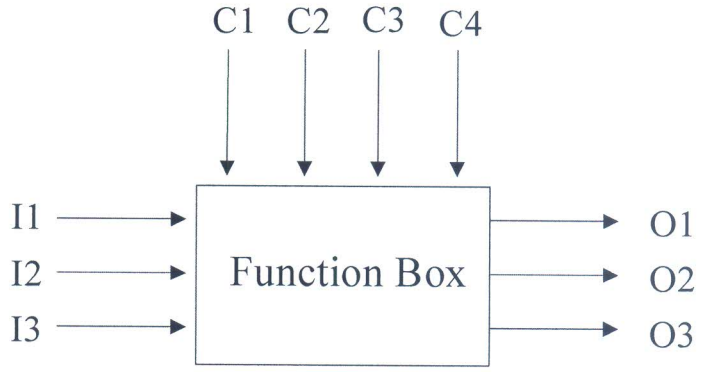


File: FIG9.cdr

A = B1 ∪ B2

Athough boxes 2 and box 3 may operate in parallel, the information A output by box 1 is split between them. In this way, the information A coming from box 1 is a *pipeline*. From this pipeline a subset goes to box 2 (information B1) and a subset goes to box 3 (information B2). These subsets are not necessarily mutually exclusive and may overlap. We identify the information split by the different labels on the arcs. A before the split, B1 on one fork and B2 on the other fork. A split can be any number of forks.

MULTIPLE INPUTS, CONTROLS AND OUTPUTS

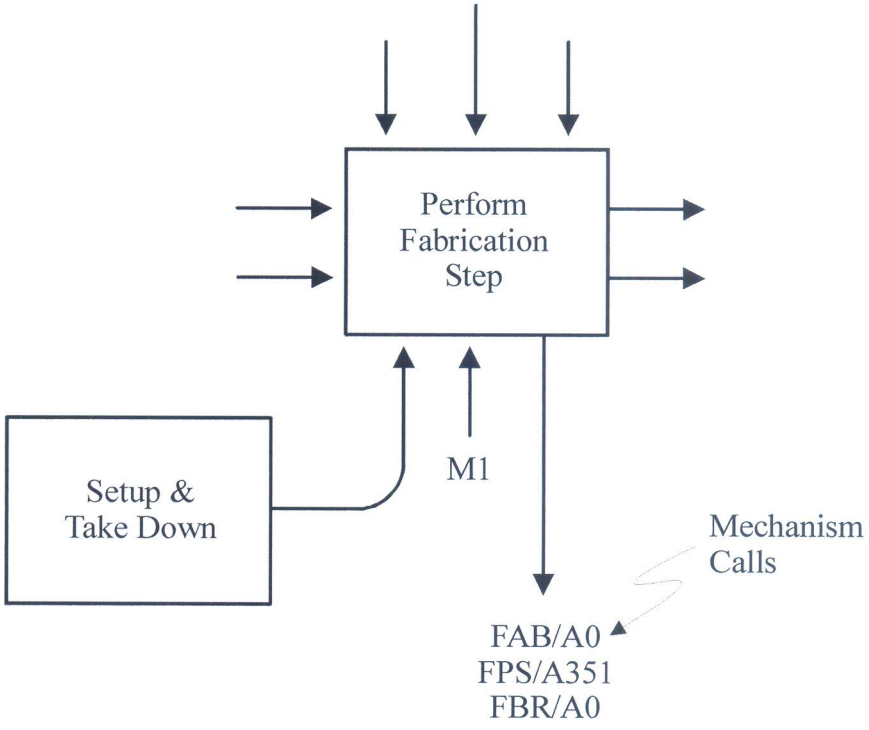


File: FIG20.cdr

In order to produce an subset of the outputs (O1, O2, and O3), any subset of the inputs (I1, I2, and I3) and constraints (C1, C2, C3, and C4) may be required. However, without further decomposition it cannot be assumed that:

1. **Any output can be produced without all inputs and constraints.**
2. **Any output requires all inputs and constraints for its production.**

MECHANISM CALLS OR MECHANISM ARCS



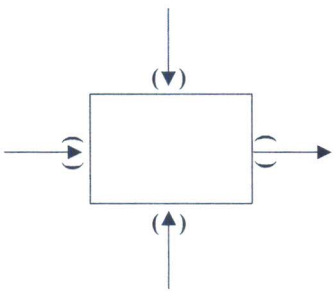
 File: FIG18.cdr

Mechanisms can be information from a previous level, output from another function on the same level, or a downward pointing arc. A *downward pointing arc* indicates that the mechanism is described by another IDEF0 model or is described by a part of another IDEF0 model. The downward pointing arc is a *mechanism call*.

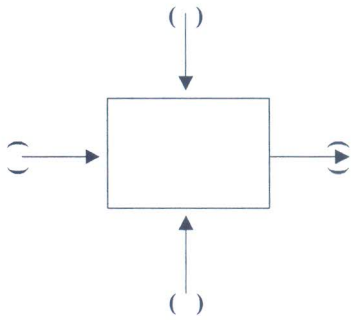
In the example, three different IDEF0 models provide mechanisms for the function "Peform Fabrication Step".

The A0 level of the FAB model, the A351 level of the FPS model, and the A0 level of the FBR model.

TUNNELED ARCS



Tunneling at the Connected End



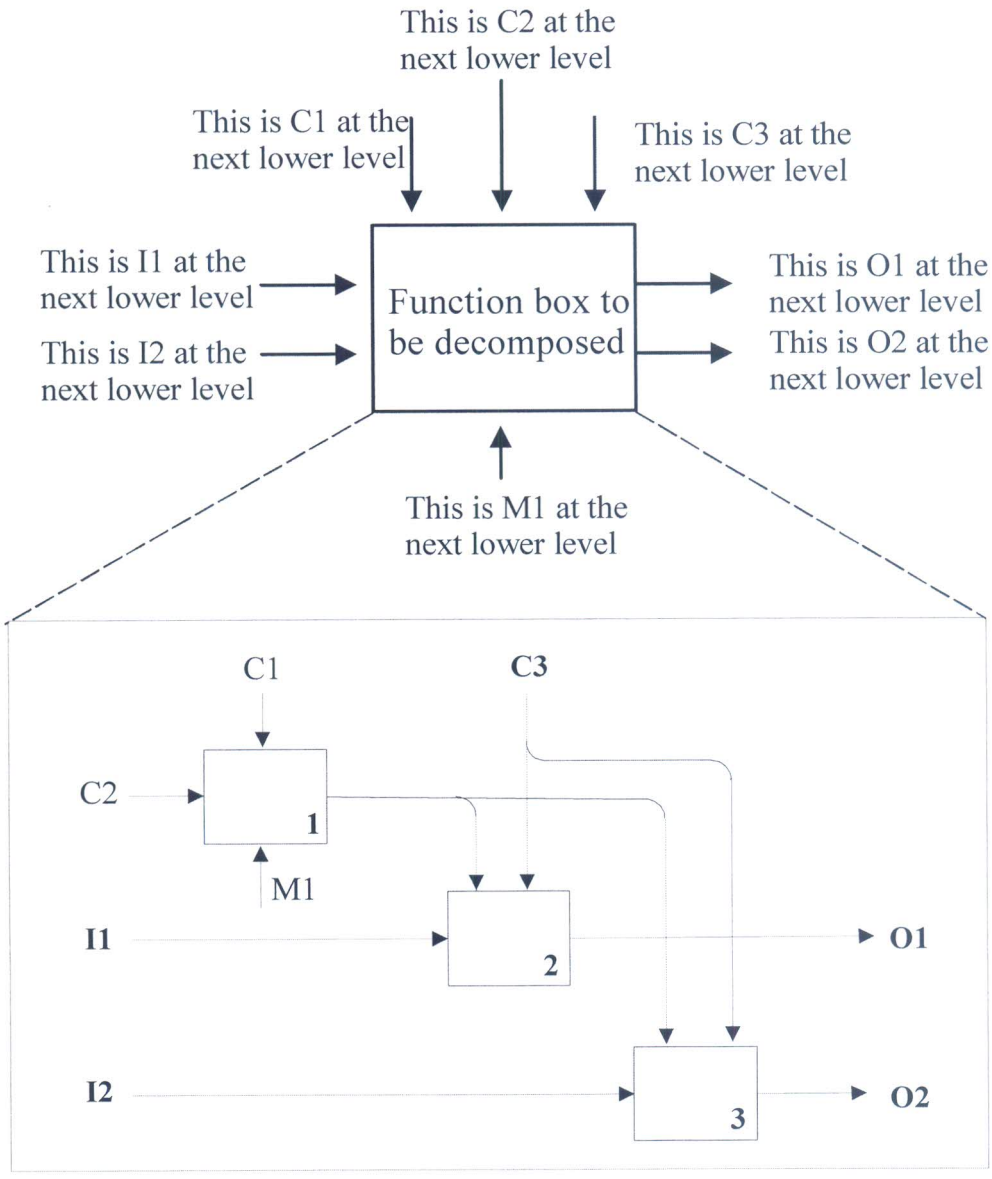
Tunneling at the unconnected End

File: FIG16.cdr

Parentheses indicate that you are *tunneling an arc*.

- **Tunneling an arrow where it connects to a function or activity box indicates that the information does not appear at the next level down in the decomposition.**
- **Tunneling an arrow at the end of an arc indicates that the information does not appear at the next level up in the decomposition.**

ICOM CODES
INPUTS, CONSTRAINTS, OUTPUTS, AND MECHANISMS



File: FIG17.DRW

CONTEXT, VIEWPOINT AND PURPOSE

CONTEXT

The context establishes the subject of the model as a part of a larger whole. It creates a boundary with the environment by describing external interfaces.

VIEWPOINT

The viewpoint determines through whose eyes the system is to be seen. It states the author of the model's position as an observer or participant in the system. This allows others who use the model to understand how they should interpret the model. For example, within a manufacturing facility the job scheduling engineer sees the system as a series of processes and queues connected by material handling. The maintenance engineer sees the system as a series of maintenance operations that need to be scheduled as well as a series of unanticipated failures that need to be repaired immediately. A different models of the manufacturing system would be built with each viewpoint.

PURPOSE

The purpose establishes the intent of the model or the goal it serves. Purpose embodies the reason why the model was built. Examples would be for functional specifications for a system upgrade, to locate system bottlenecks, to correct congestion problems, etc.

THE A-0 DIAGRAM
PROVIDING CONTEXT

The A-0 diagram is a single function box diagram that bounds the context for the entire model and forms the basis for further model decomposition.

Arcs entering and leaving the function box identify significant information flowing into and out-of the system being modeled. These are the interactions between the system being modeled and the outside environment.

EXAMPLE A-0 DIAGRAM

 File: FIG21.cdr

All inputs, constraints and outputs of the model can be related to the inputs, constraints and outputs shown in this diagram.

SUPPORTING MATERIAL

Each IDEF0 model includes supporting material necessary to understand the model. This material includes the following:

GLOSSARY

The glossary defines all function boxes and the labels used in each diagram. These definitions should be concise and consistent with the common usage in the system being modeled.

TEXT

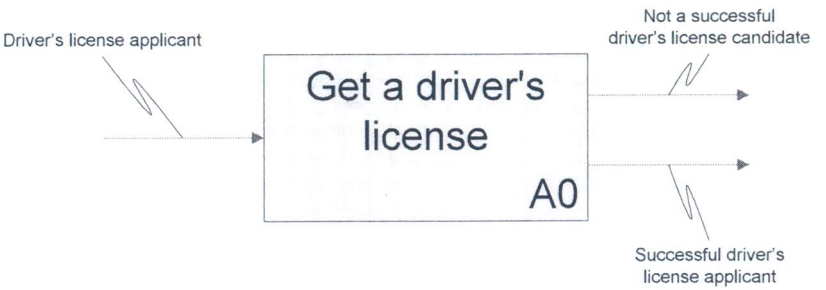
Text may accompany a diagram as a supplement to explain what is contained in the diagram.

GUIDELINES FOR CREATING AN IDEF0 MODEL

- Use a *landscape* page layout and not a *portrait* page layout for the model.
- Each page should be framed in the *Title block*.
- Layout the function boxes diagonally so that information flows from upper left to lower right.
- The maximum number of function boxes in a diagram using a standard 8.5 × 11 sheet should be about six.
- Try to arrange the function boxes to minimize the arc crossings. Large numbers of arc crossings make a diagram confusing and difficult to follow.
- Number boxes sequentially beginning with the upper left box.
- Associate database application forms with activities by attaching the forms to an activity as a mechanism.

EXAMPLE IDEF0 MODEL

The example IDEF0 model is a model of an individual getting a driver's license in North Carolina. It includes database application forms as mechanisms to various nodes. In this example, the forms are conjectured. Thus, the model could be considered a design specification in that it identifies the activities that the database application will support and the general nature of the form set for that support.



Context

Someone getting a driver's license in North Carolina. This includes a person who:

* is renewing a license
* has never had a license in North Carolina but has a valid license from elsewhere
* has never held a driver's license anywhere

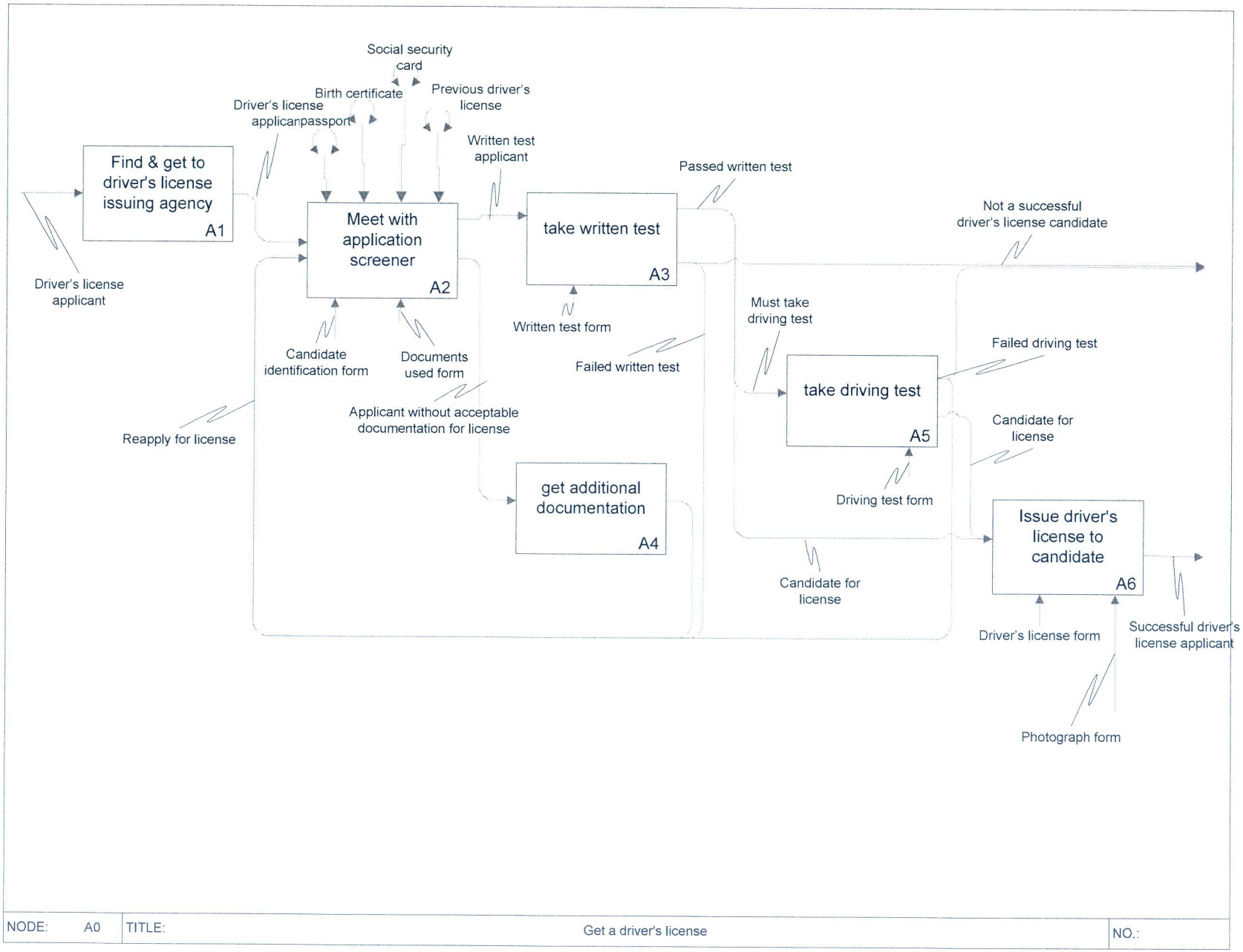
Viewpoint

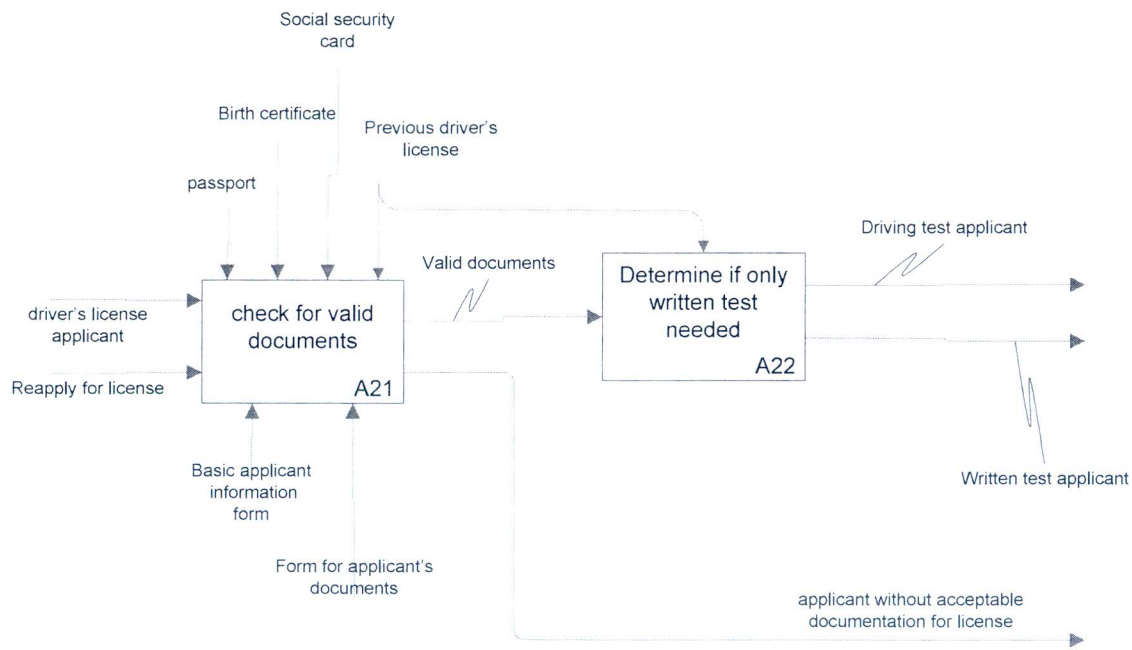
The viewpoint is from the person who is getting the license. It is not from the examiners perspective, nor from a driver's license administrative perspective.

Purpose

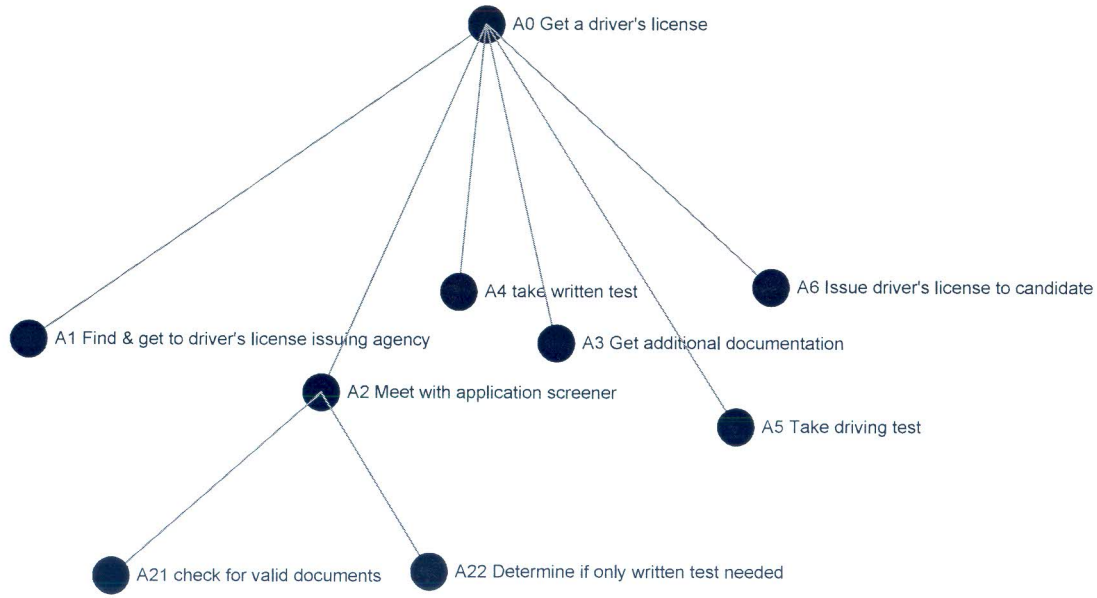
The purpose is to show the steps necessary to obtain a driver's license in North Carolina and to identify the forms that support the different activities.

NODE: A-0	TITLE: Get driver's license	NO.:


Social security card
Birth certificate
Previous driver's license
Driver's license applicanpassport
Written test applicant
Passed written test
Find & get to driver's license issuing agency
A1
Meet with application screener
A2
take written test
A3
Not a successful driver's license candidate
Driver's license applicant
Written test form
Must take driving test
Failed written test
Failed driving test
Candidate identification form
Documents used form
take driving test
A5
Reapply for license
Applicant without acceptable documentation for license
Candidate for license
get additional documentation
A4
Driving test form
Issue driver's license to candidate
A6
Candidate for license
Driver's license form
Successful driver's license applicant
Photograph form
NODE: A0
TITLE:
Get a driver's license
NO.:



NODE: A2	TITLE: Meet with application screener	NO.:



NODE: Tree	TITLE: Get a driver's license	NO.: