IE 469 Manufacturing Systems

469 صنع نظم التصنيع

FUNCTION MODELING USING IDEF-0

What is IDEF?

- ***Definition*: IDEF is the common name referring to classes of enterprise modeling languages.**
- ***Objective*: IDEF is used for modeling activities necessary to support system analysis, design, improvement or integration.**
- ***Originally*, IDEF was developed to enhance communication among people trying to understand the system. *Now*, IDEF is being used for documentation, understanding, design, analysis, planning, and Integration.**

IDEF History

- **In the 1970's, IDEF0 originated in the U.S. Air Force under the Integrated Computer Aided Manufacturing(ICAM) program from a well-established graphical language, the Structured Analysis and Design Technique (SADT).**

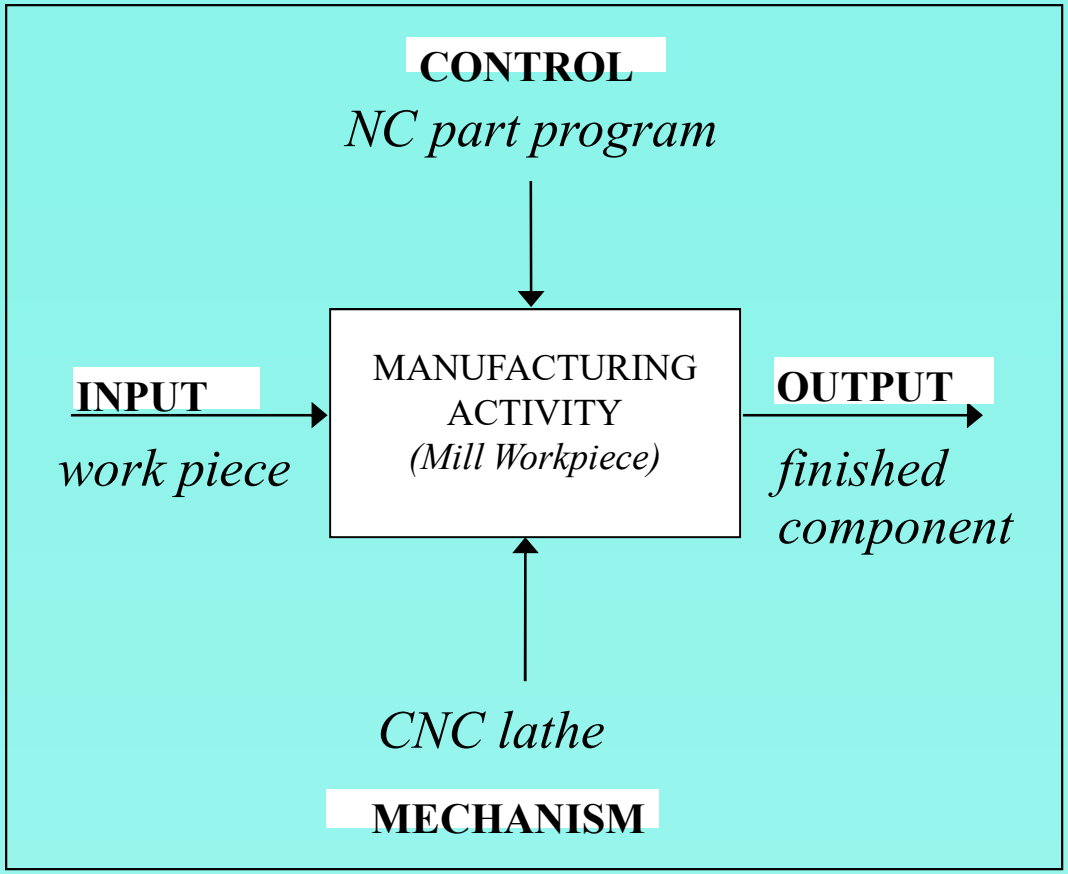
IDEF Family

- **IDEF Family of Methods:**
  - **IDEF0: for Function Modeling (purpose:description)**
  - **IDEF1: for Information Modeling. (purpose:description)**
  - **IDEF1x: for Data Modeling. (purpose:design)**
  - **IDEF3: for Process Modeling. (purpose:description)**
  - **IDEF4: for Object-Oriented Design. (purpose:design)**
  - **IDEF5: for Ontology Description Capture. (purpose:description)**

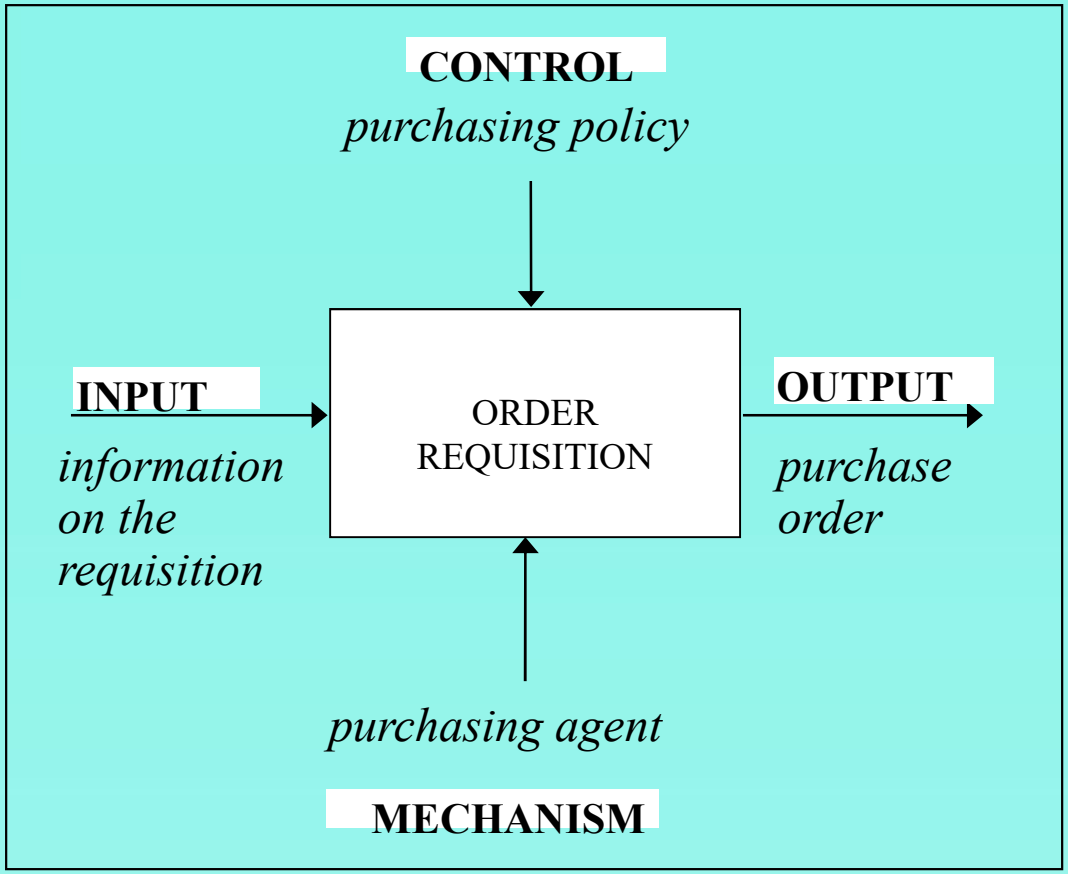
IDEFØ

The IDEF Function Modeling Method

IDEF0:Example 1

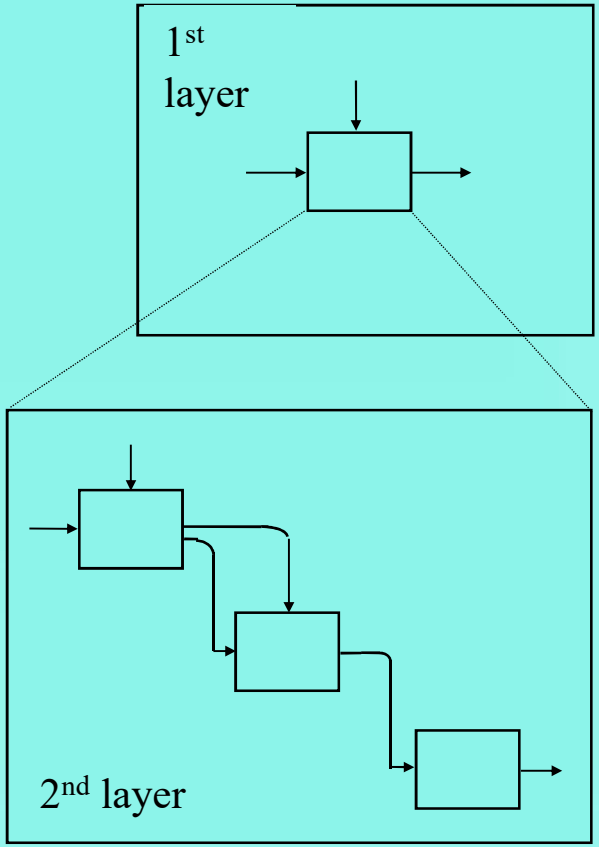


IDEF0:Primitives Example 2



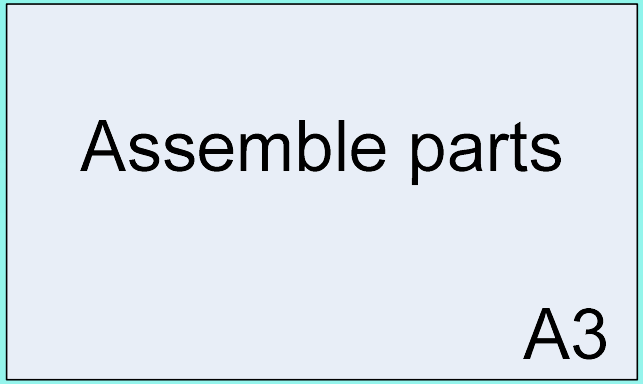
IDEF0 Methodology

- **The activity box and four arcs provide a concise expression: an *input* is transformed into an *output* by an *activity* (function) performed by a *mechanism* and governed by a *control*.**
- **Top down modeling approach**
- **1st layer is a single activity box that describes the function or process that is the subject of the model**
- **2nd layer is the decomposition of the first layer into major sub-activities.**

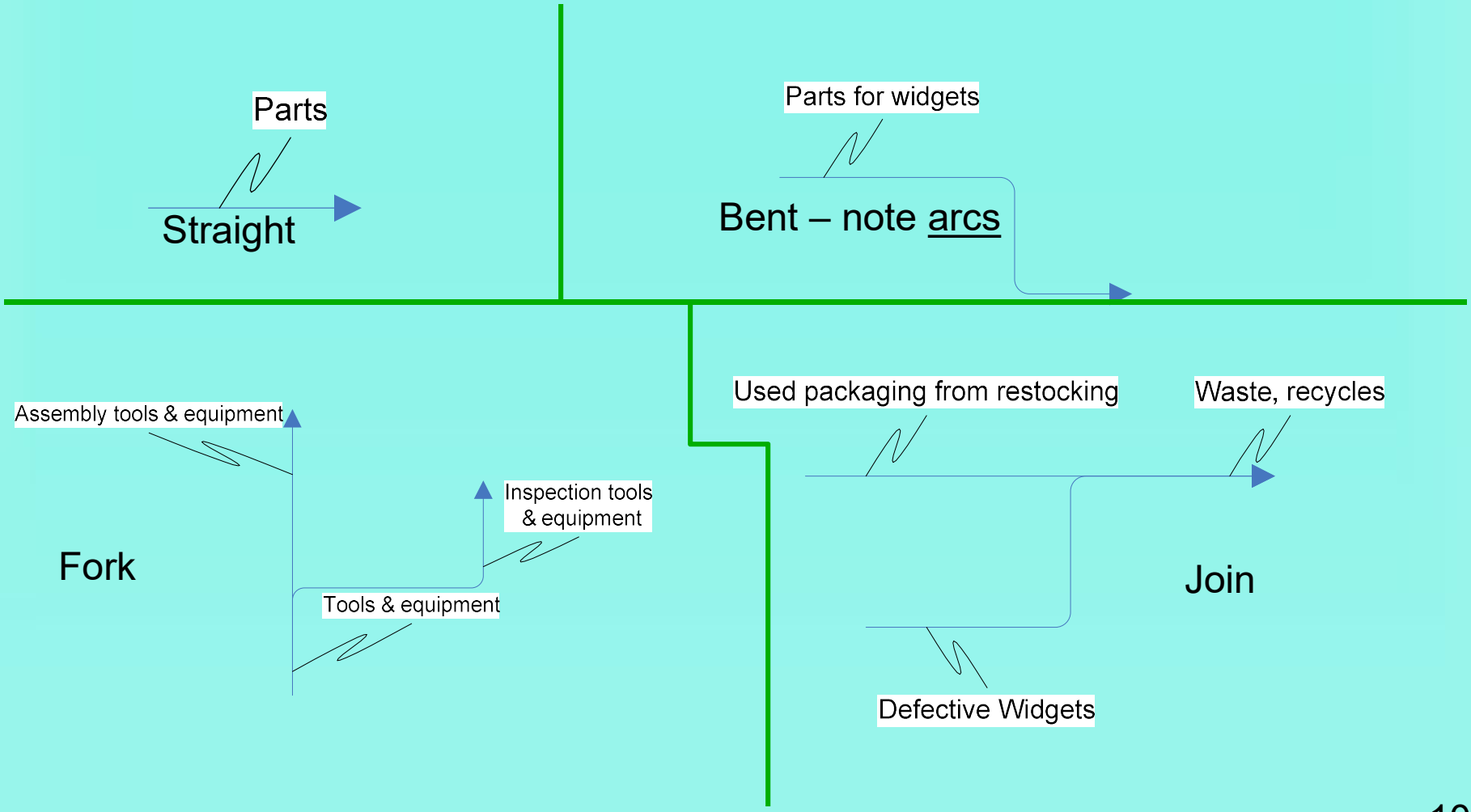


Syntax: Boxes

- **Solid lines**
- **Verb or verb phrase**
- **Box number**



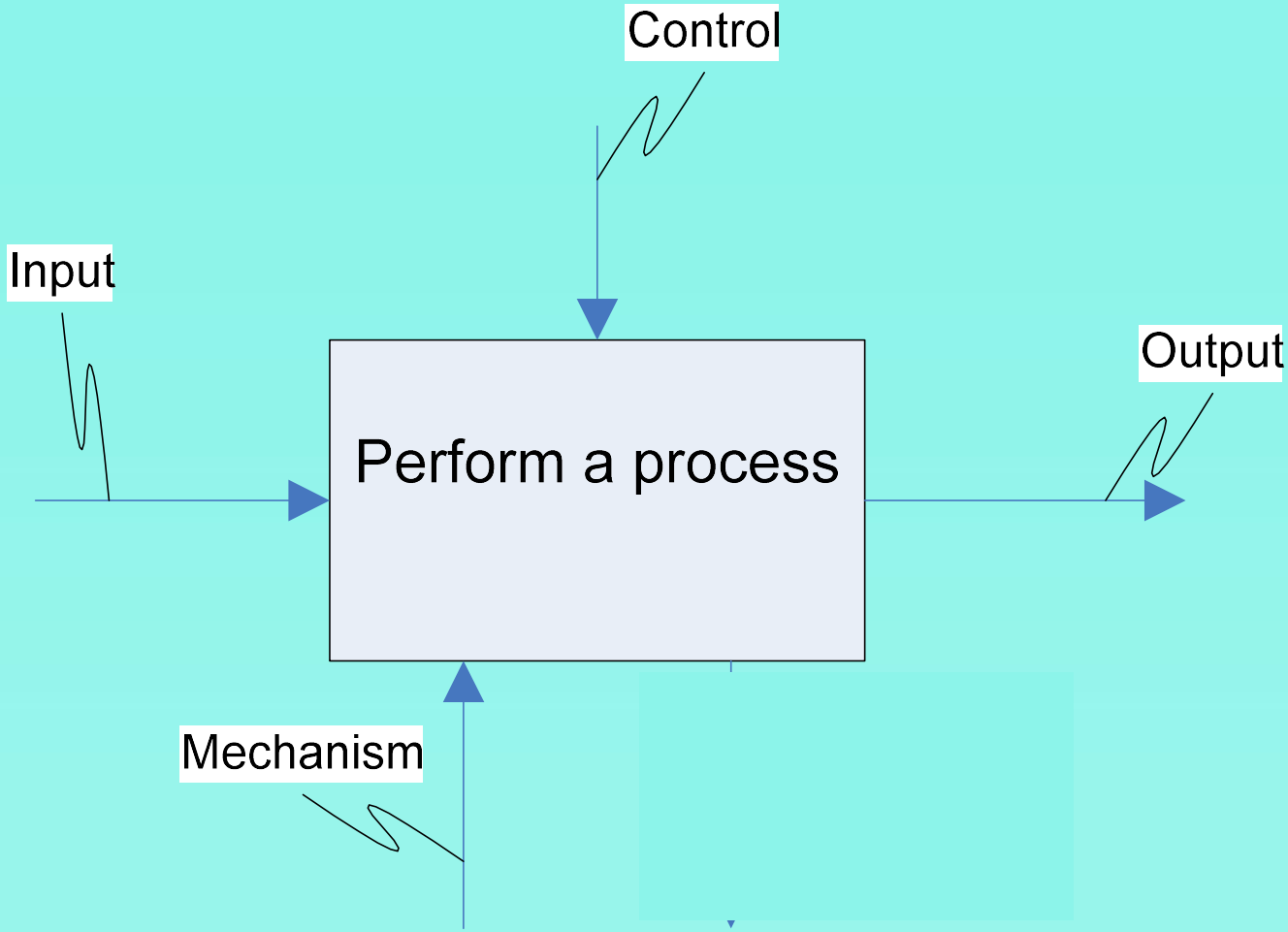
Syntax: Arrows



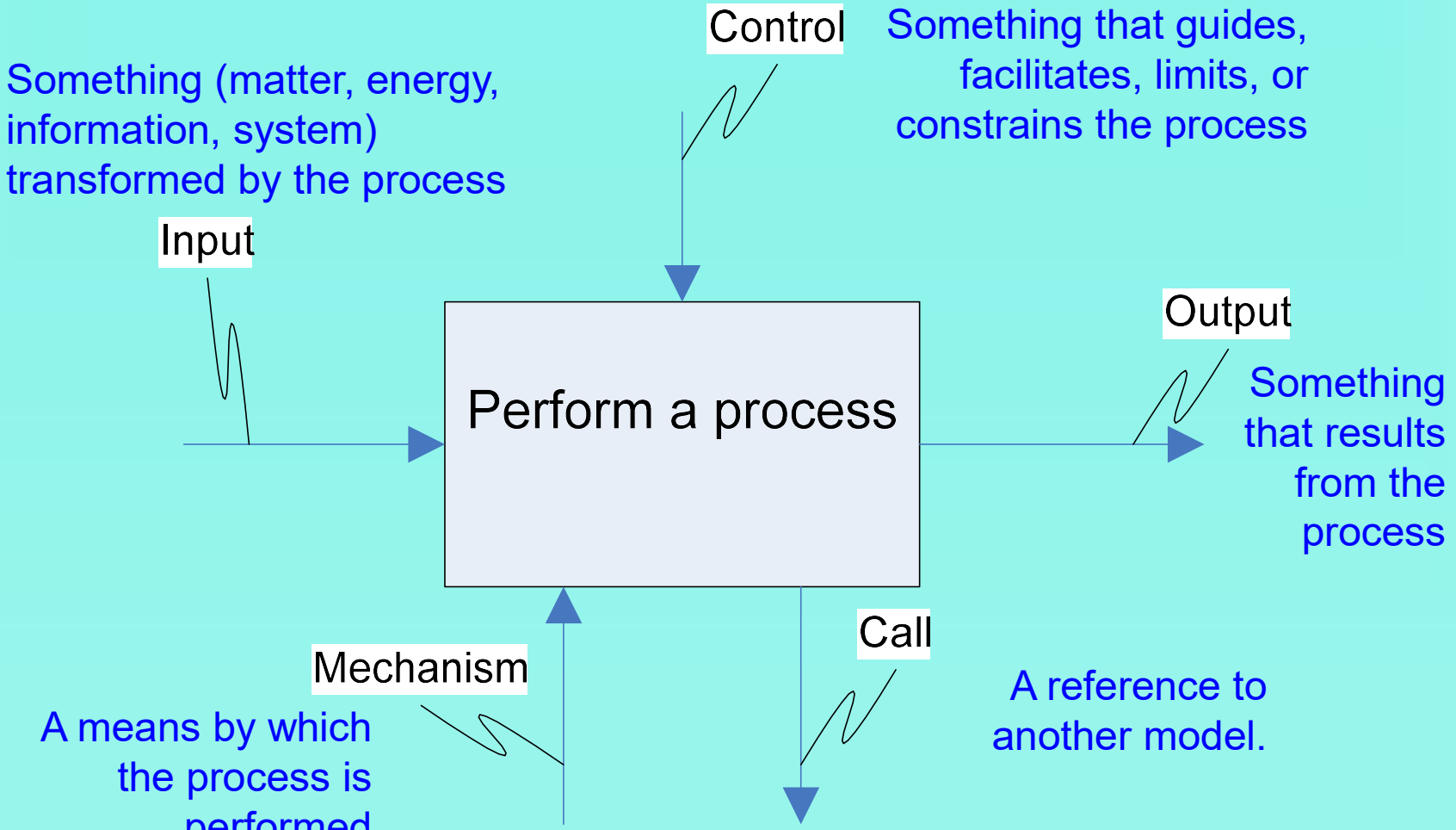
Box and Arrow Syntax Rules

- **Boxes**
  - **Boxes shall be sufficient in size to insert box name.**
  - **Boxes shall be rectangular in shape, with square corners.**
  - **Boxes shall be drawn with solid lines.**
- **Arrows**
  - **Arrows that bend shall be curved using only 90 degree arcs.**
  - **Arrows shall be drawn in solid line segments.**
  - **Arrows shall be drawn vertically or horizontally, not diagonally.**
  - **Arrow ends shall touch the outer perimeter of the function box and shall not cross into the box.**
  - **Arrows shall attach at box sides, not at corners.**

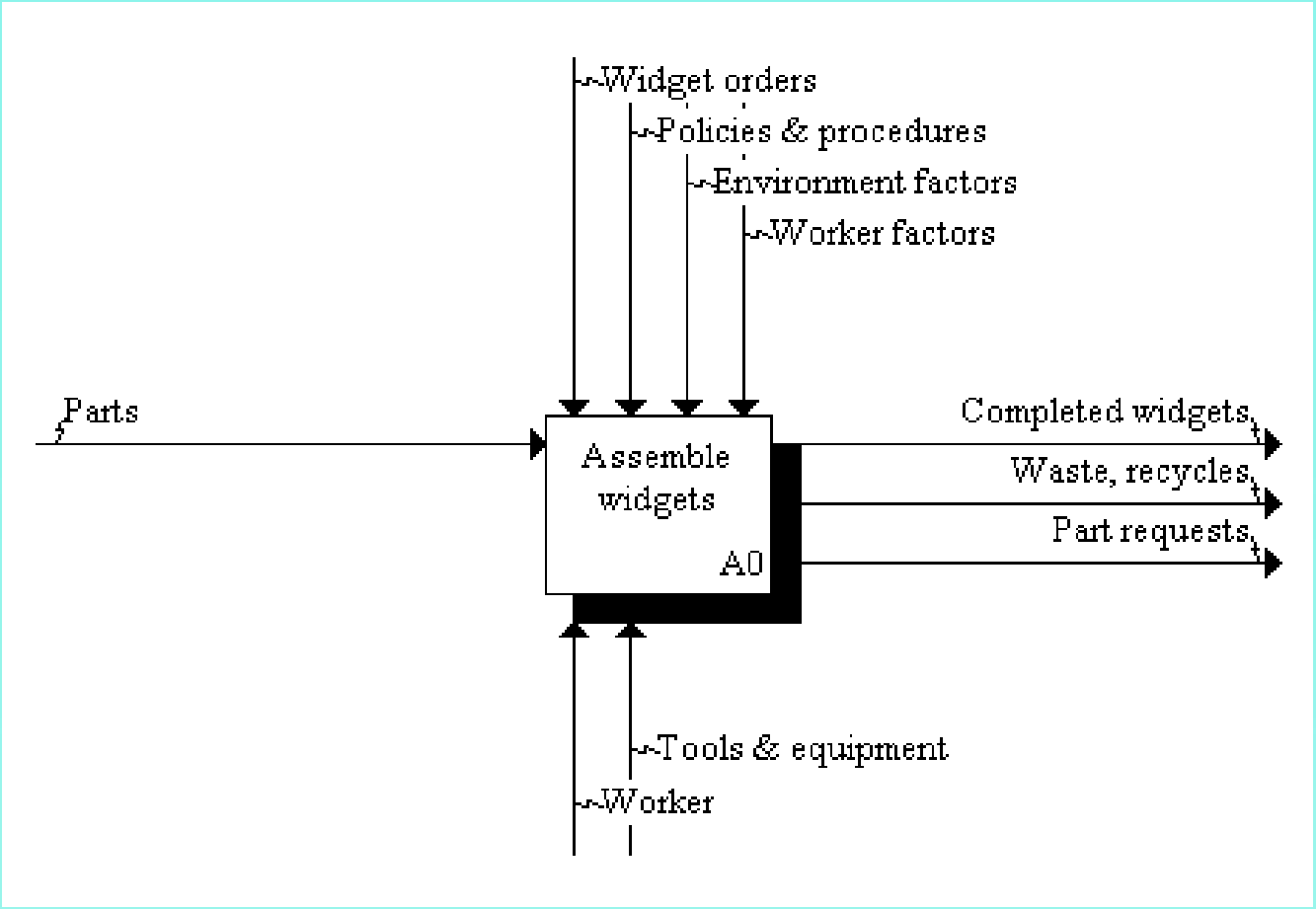
Semantics



Semantics


Control
Something that guides, facilitates, limits, or constrains the process
Something (matter, energy, information, system) transformed by the process
Input
Perform a process
Output
Something that results from the process
Mechanism
A means by which the process is performed
Call
A reference to another model.

Example


Widget orders
Policies & procedures
Environment factors
Worker factors
Parts
Assemble widgets
A0
Completed widgets
Waste, recycles
Part requests
Tools & equipment
Worker

More Box and Arrow Syntax Rules

1. **A box shall be named with an active verb or verb phrase.**
2. **Each side of a function box shall have a standard box/arrow relationship:**
   a. **Input arrows shall interface with the left side of a box.**
   b. **Control arrows shall interface with the top side of a box.**
   c. **Output arrows shall interface with the right side of the box.**
   d. **Mechanism arrows (except call arrows) shall point upward and shall connect to the bottom side of the box.**
   e. **Mechanism call arrows shall point downward, shall connect to the bottom side of the box, and shall be labeled with the reference expression for the box which details the subject box.**
3. **Arrow segments, except for call arrows, shall be labeled with a noun or noun phrase unless a single arrow label clearly applies to the arrow as a whole.**
4. **A "squiggle" (⫽) shall be used to link an arrow with its associated label, unless the arrow/label relationship is obvious.**
5. **Arrow labels shall not consist solely of any of the following terms: function, input, control, output, mechanism, or call.**

IDEF0 Diagrams and Text

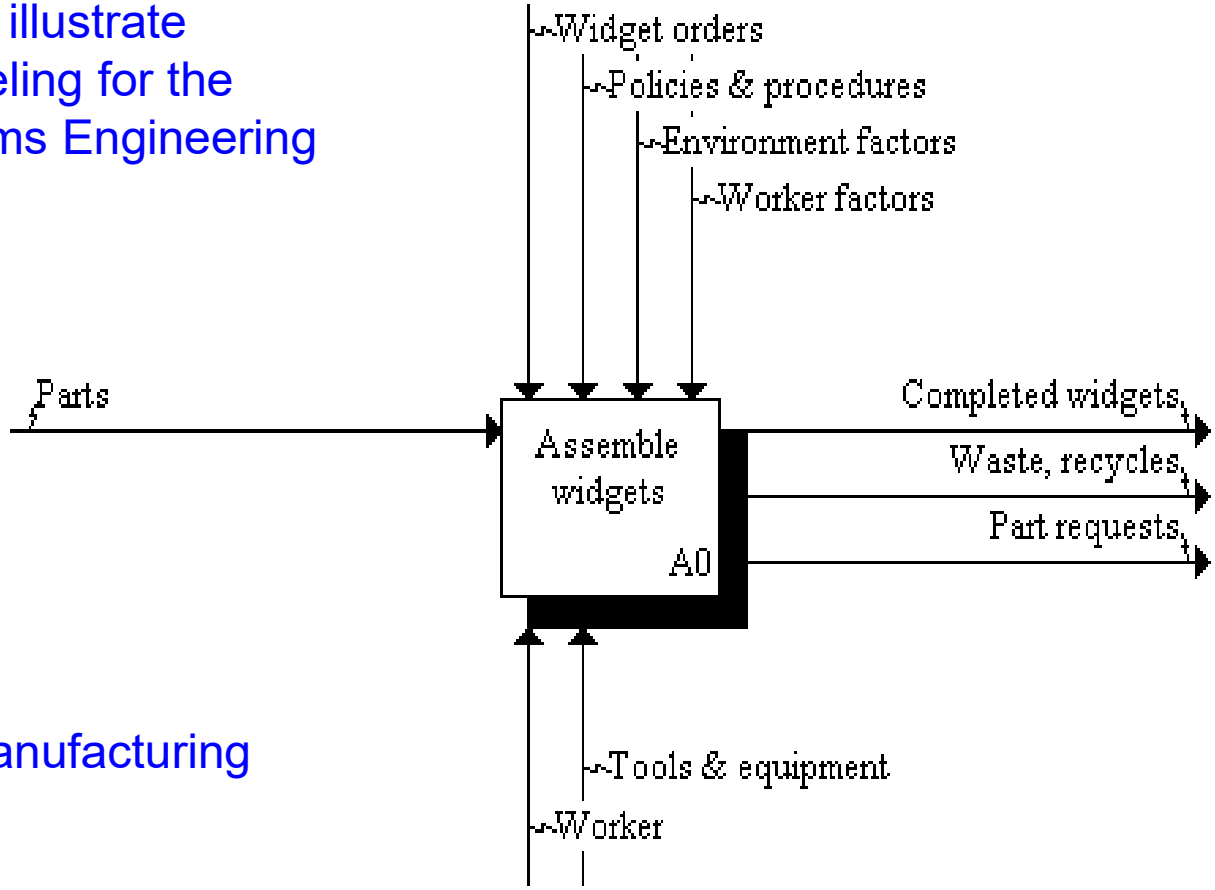
- **Top-Level Context Diagram**
- **Child Diagram**
- **Parent Diagram**
- **Text and Glossary**
- **For Exposition Only Diagrams**

Top-Level Context Diagram

- **Subject of model represented by single box with bounding arrows.**
- **Called A-0 (“A minus zero”)**
- **Box and arrows are very general**
- **Sets model scope or boundary and orientation.**
- **Should include**
  - **Purpose**
  - **Viewpoint**

Example Context Diagram: A-0 Assemble widgets

Purpose: To illustrate IDEF0 modeling for the Work Systems Engineering process.



Viewpoint: Industrial/manufacturing engineer.

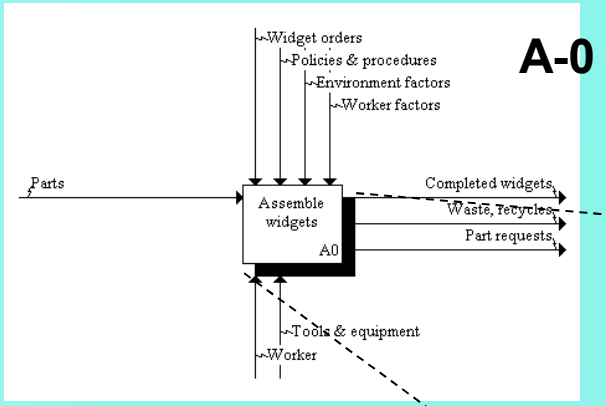
Child Diagram

- **Single process in Context Diagram (A-0) may be decomposed into subprocesses and modeled in a child (A0) diagram.**
- **Each process in the A0 diagram may be decomposed further into subprocesses and modeled in (grand-) child (A1, A2, … A6) diagrams.**
- **Each (grand-) child process may be decomposed further into subprocesses and modeling (great-grand-) child diagrams.**
- **And so on …**

Parent Diagram

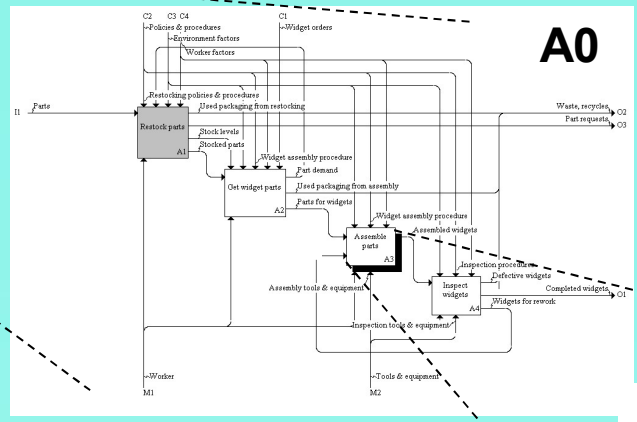
- **Diagram that contains one or more parent boxes, i.e., boxes detailed on child diagrams.**

Process Decomposition



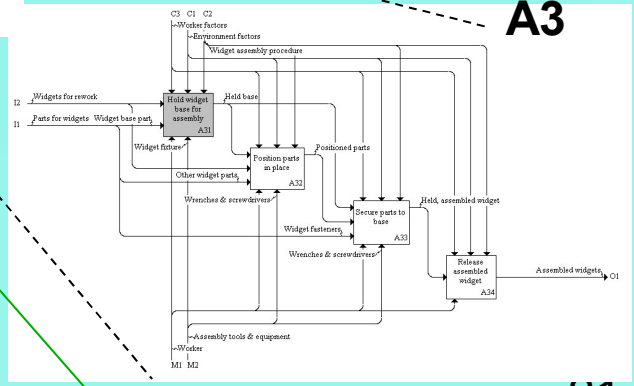
parent

child



parent

child



Text and Glossary

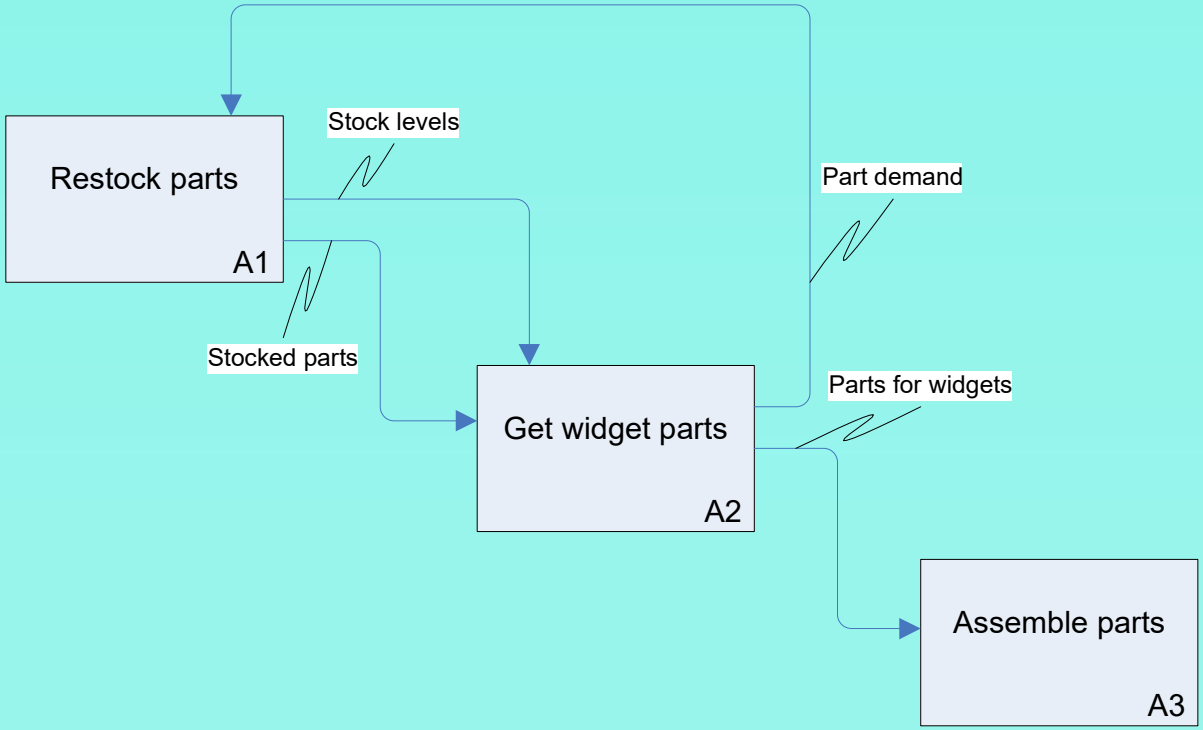
- **Text**
  - **Associated textual information used to clarify model.**
- **Glossary**
  - **Definitions of**
    - » **processes (activities, functions)**
    - » **inputs**
    - » **controls**
    - » **outputs**
    - » **mechanisms**
  - **Examples**
    - » **Get widget parts (process)**
      - **The process of getting widget parts from the stock areas so that widgets may be assembled.**
    - » **Parts for widgets (output)**
      - **Parts retrieved from the workstation stock areas and ready to be used in assembly.**

Diagram Features

- **Arrows As Constraints**
- **Concurrent Operation**
- **Arrows As Pipelines**
- **Branching Arrows**
- **Inter-Box Connections**
- **Boundary Arrows**
- **Tunneled Arrows**
- **Call Arrows**

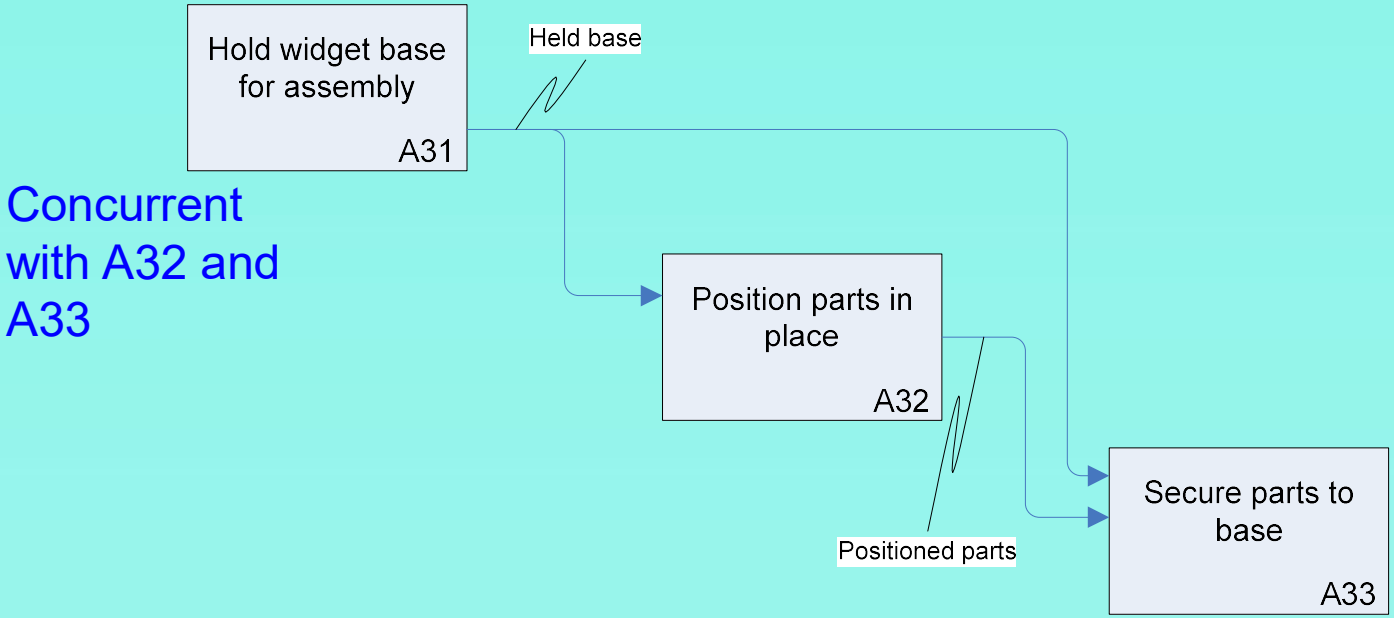
Arrows As Constraints

- **Connecting output of a box representing a process that is input/control/mechanism to another box means that the second process is constrained by the first.**



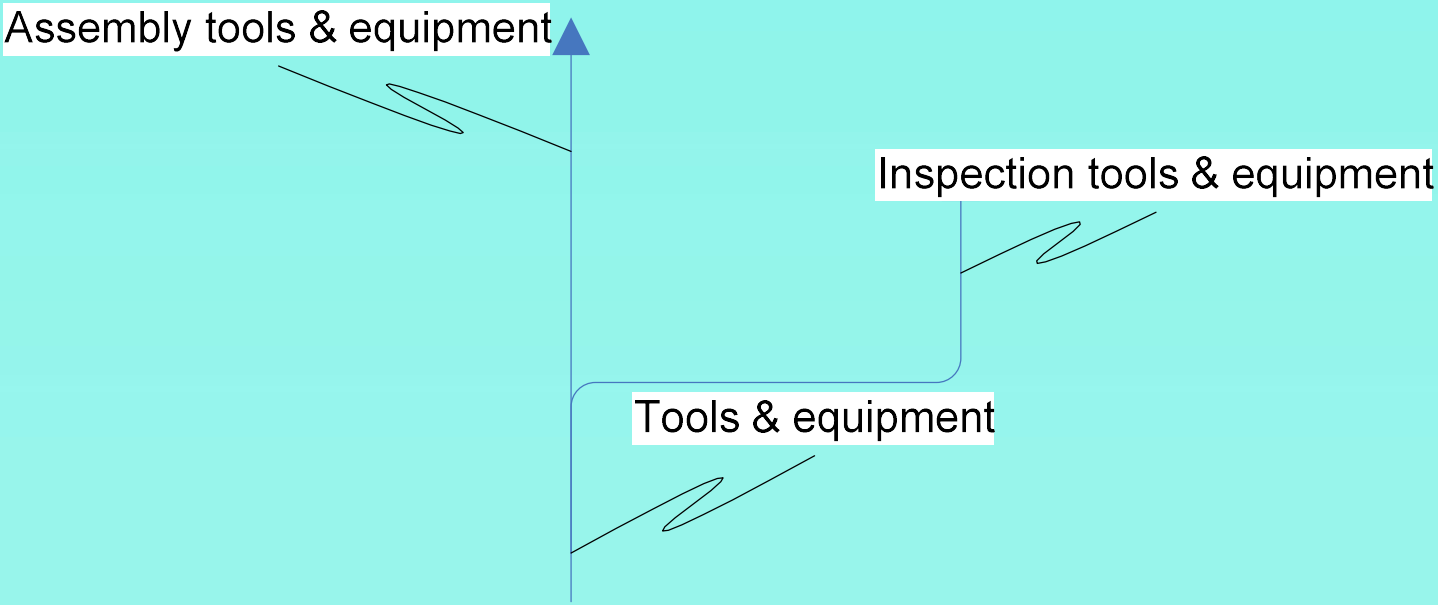
Concurrent Operation

- **Box order and connections do not necessarily imply sequence!**
- **Processes may proceed concurrently.**

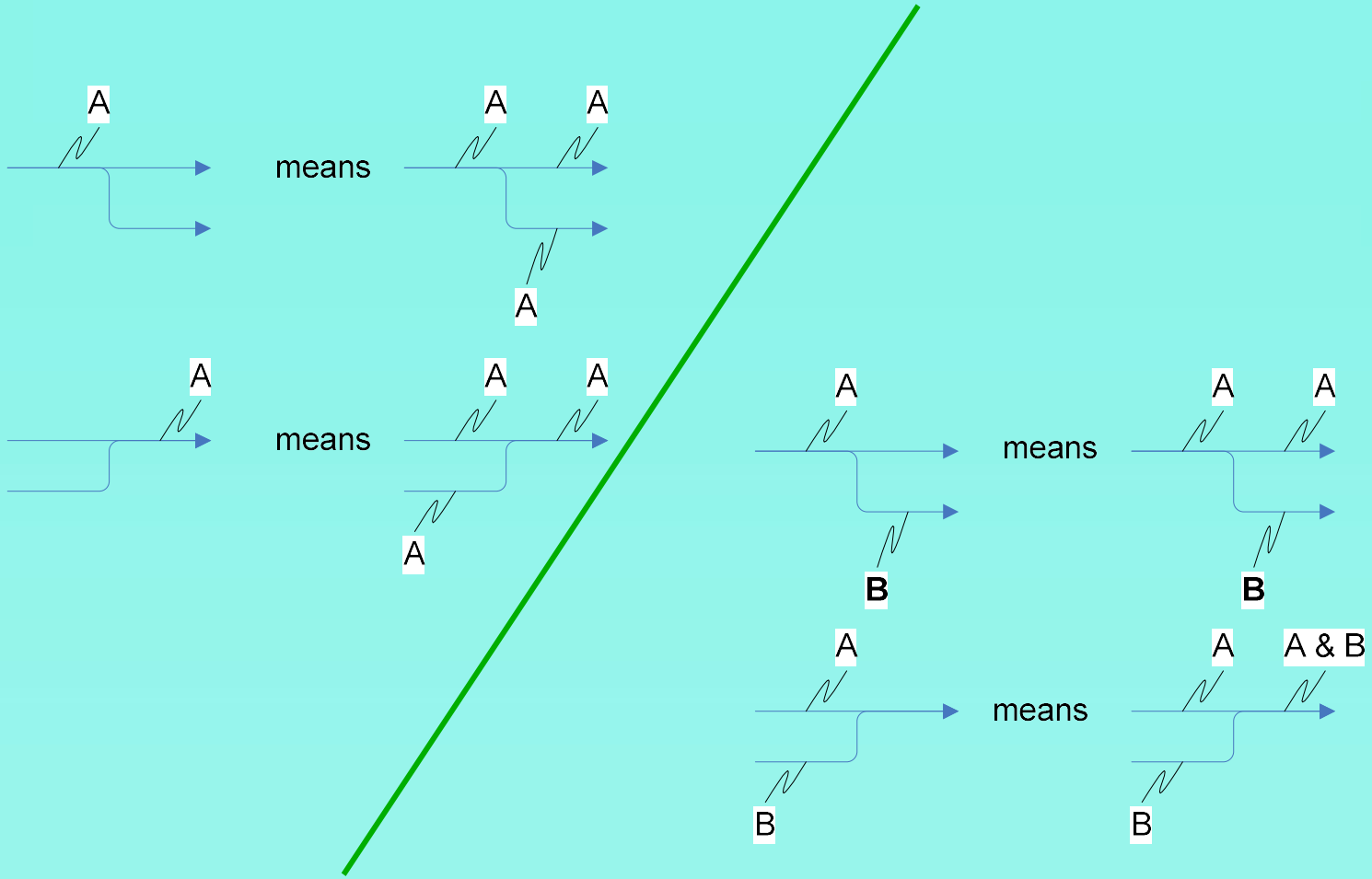


Arrows As Pipelines

- **Think of arrows as pipelines or conduits.**
- **High-level arrows have general labels.**
- **Low-level arrows have specific labels.**
- **If an arrow forks, the branches may have more specific**



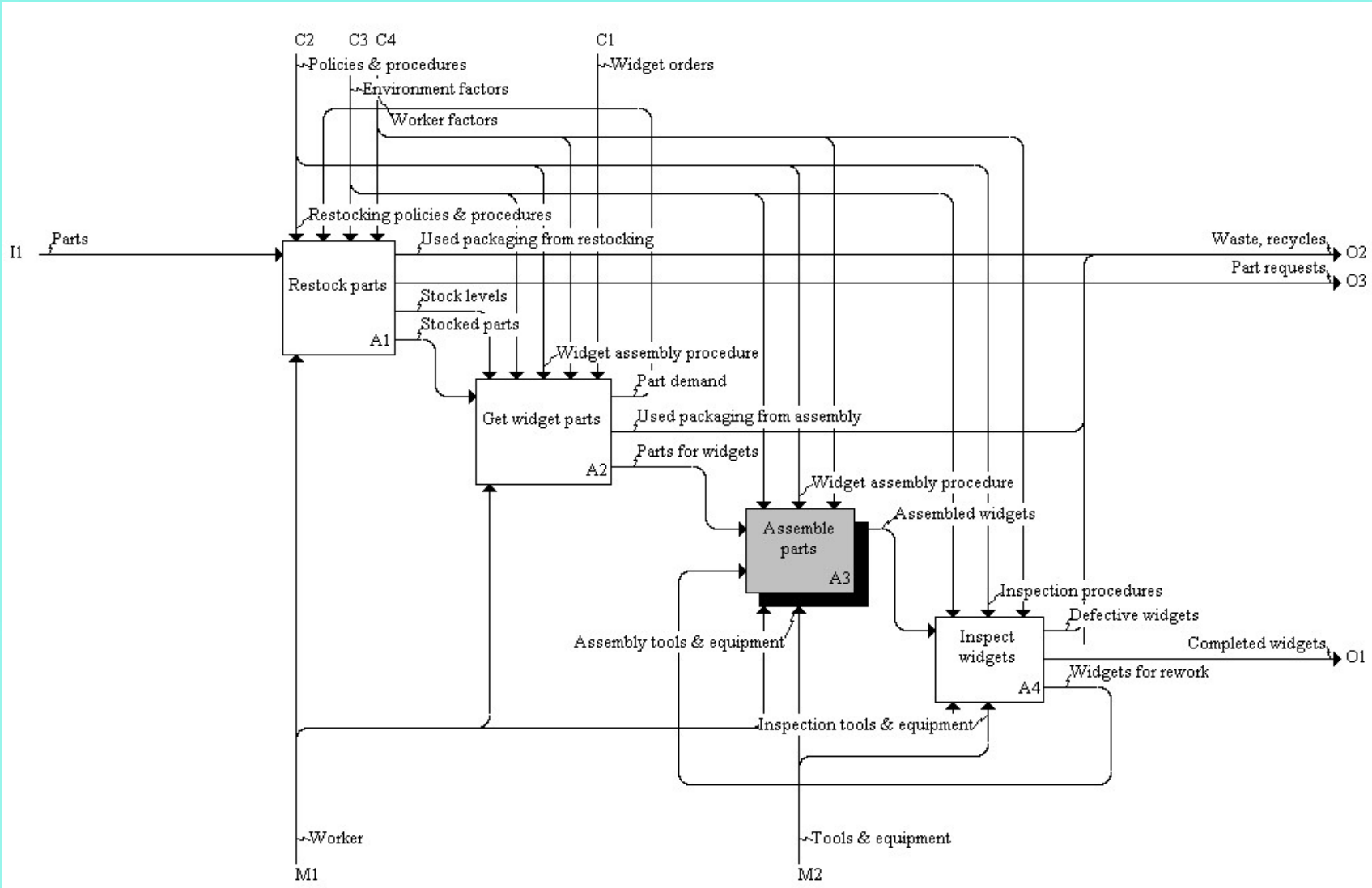
Branching Arrows



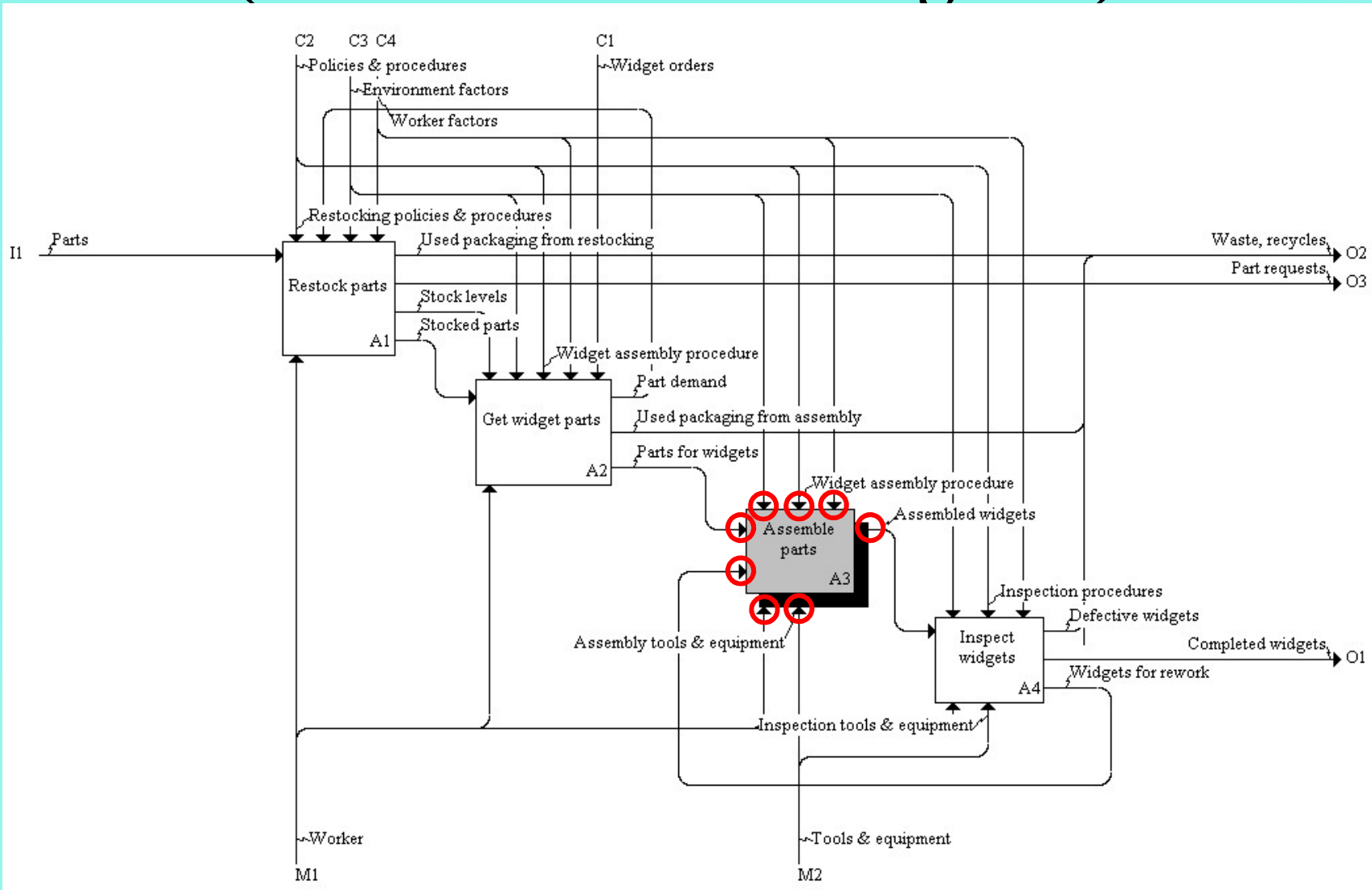
Inter-Box Connections

- **Except for A-0, diagrams contain 3 – 6 boxes.**
- **Normally organized on diagonal (“staircase”).**
- **Any output of one box may be input, control, or mechanism of another box.**
- **If box is detailed on child diagram, every arrow connected to the box appears on the child diagram (unless it is tunneled).**

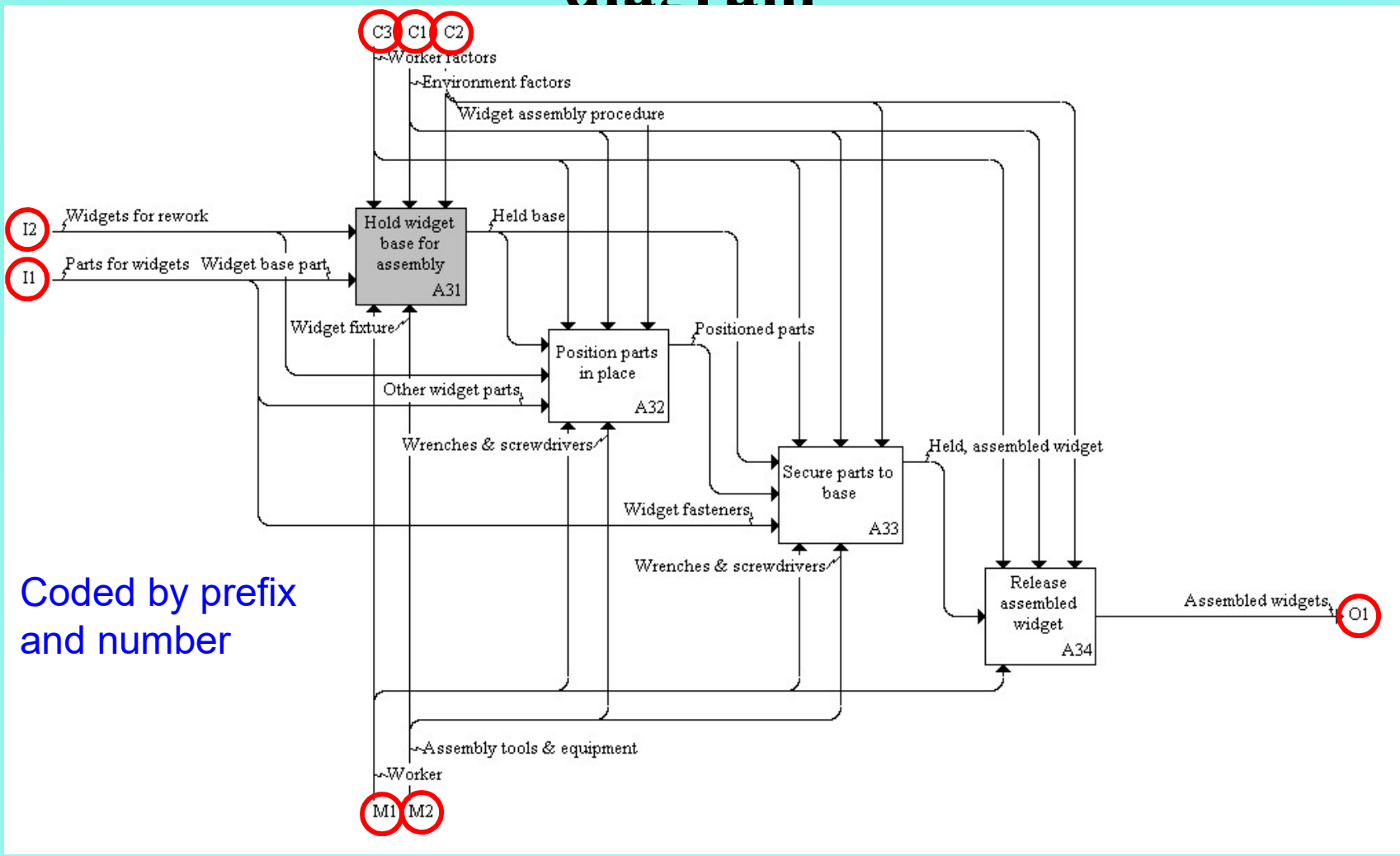
Inter-Box Connections



Inter-Box Connections (arrows for child diagram)



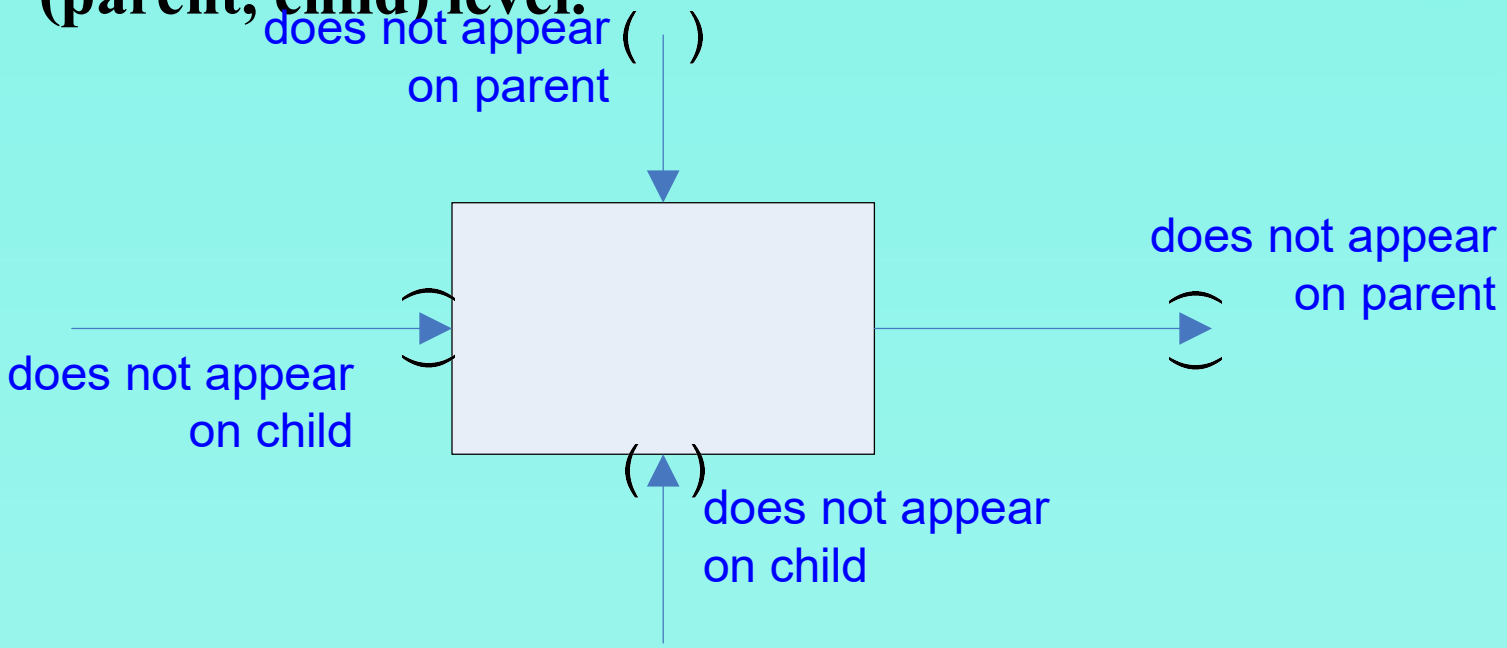
Boundary Arrows:
Arrows from parent box on parent diagram



Coded by prefix and number

Tunneled Arrows

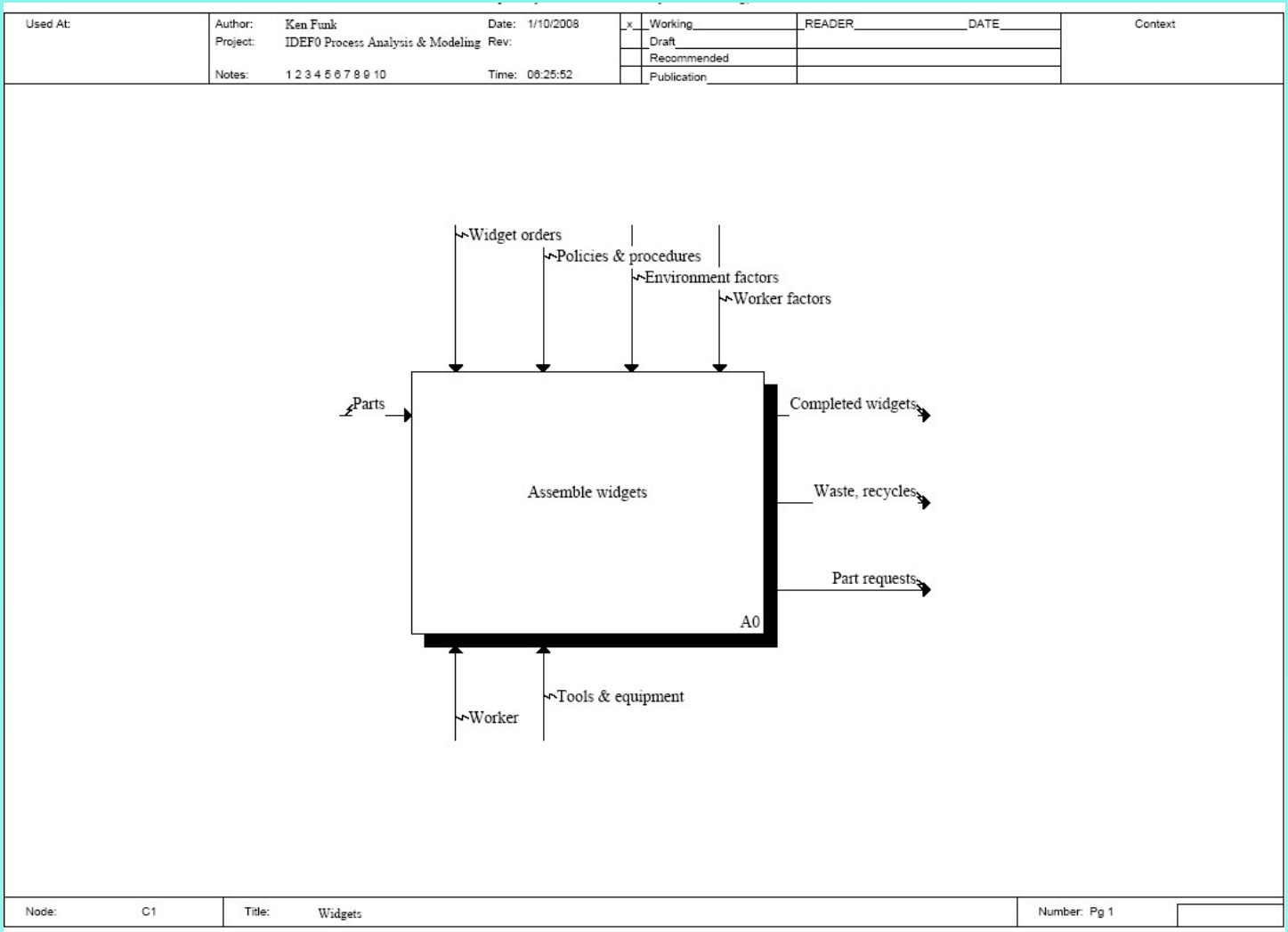
- **Arrows that provide information at one level of decomposition but are not needed at another (parent, child) level.**



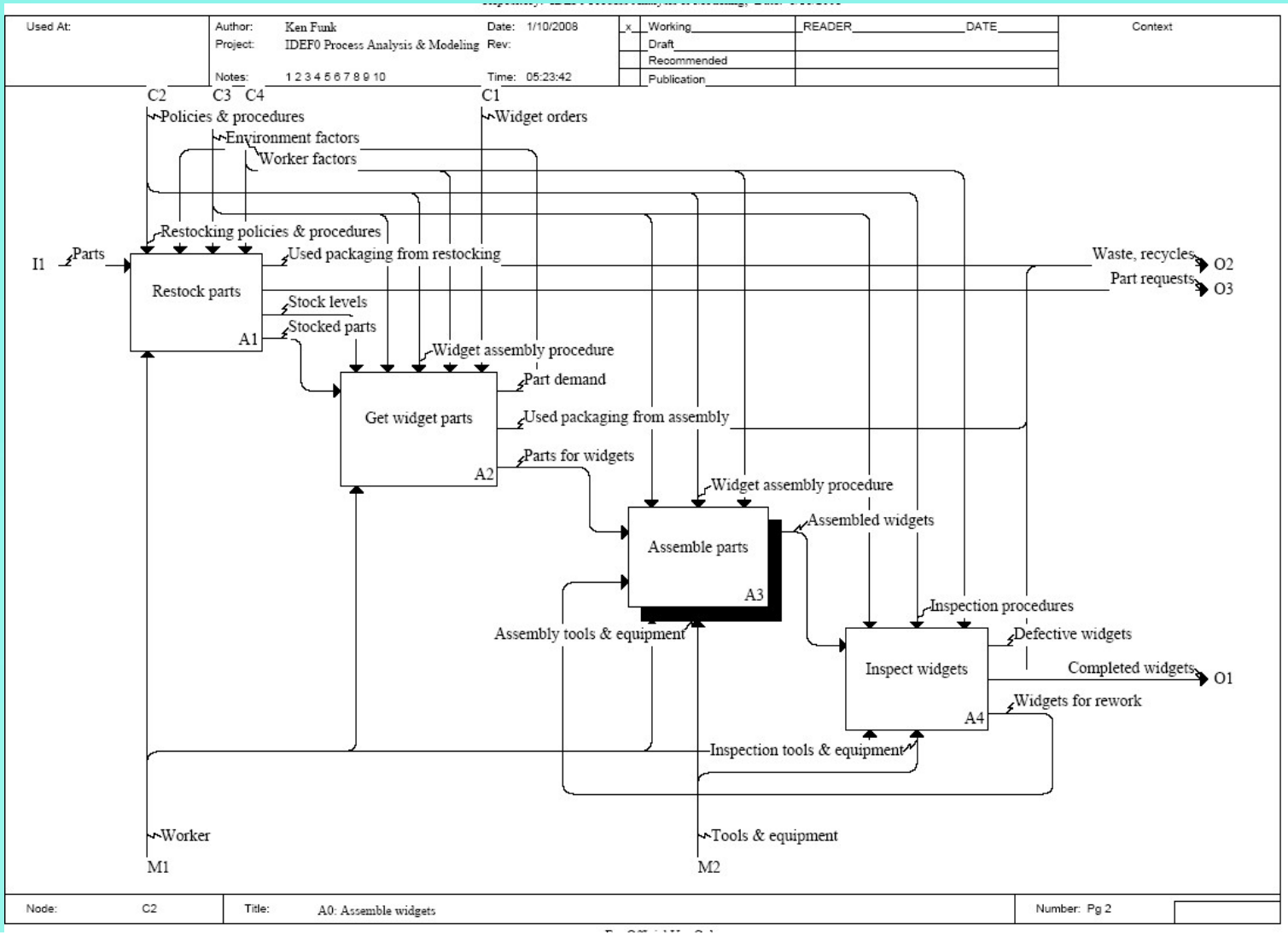
Box Numbers and Node Numbers

- **Box numbers**
  - **Single box in context (A-0) diagram numbered A0 ("Activity" 0).**
  - **Boxes in context diagram's child numbered A1, A2, A3, … [A6].**
  - **Boxes in A1's child diagram numbered A11, A12, …**
  - **Boxes in A2's child diagram numbered A21, A22, …**
  - **Boxes in A21's child diagram numbered A211, A212, …**
  - **and so on …**
- **Node – for our purposes, another name for a diagram**
- **Node numbers**
  - **Context diagram is node A-0**
  - **A-0's child node is node A0**
  - **A0's children are nodes A1, A2, …**
  - **In general, a node bears the same number as the box in the parent node it details.**

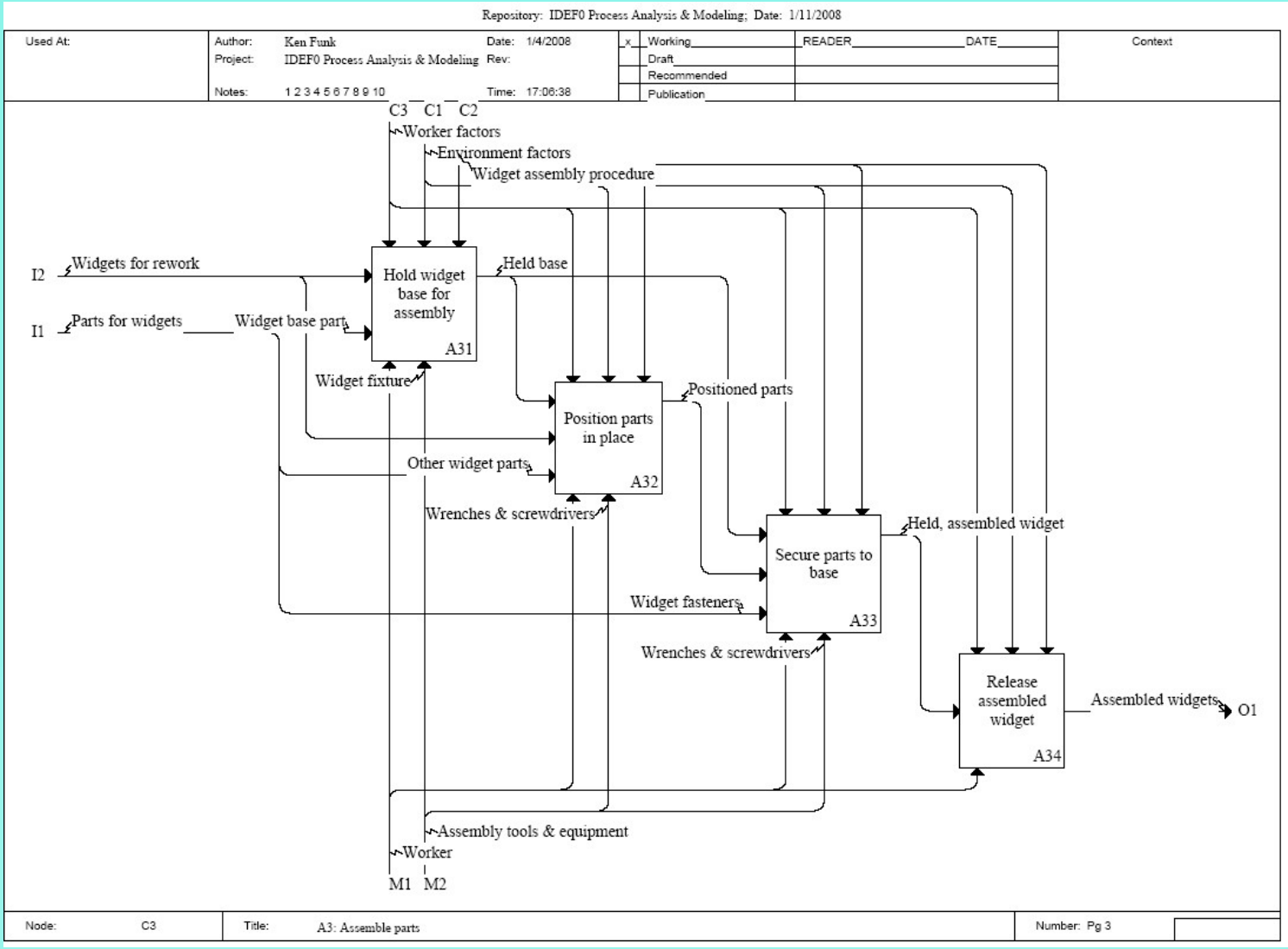
Node (Diagram) A-0 (Context)



Node (Diagram) A0



Node (Diagram) A3



Terminology of IDEFØ

- **Function Modeling**
- **Functions and activities**
- **Diagrams, Boxes, and Arrows**
- **ICOMs: Inputs, Controls, Outputs, and Mechanisms**
- **Arrows, links, relationships, and concepts**
- **Splits, Joins, Unbundling, Bundling, and Branching**
- **Decompositions**
- **Viewpoint, Purpose, and Context**
- **NIST (FIPS) standard**

What is a Function Model?

A Representation of the Activities and Relationships Between Activities in an Existing or Planned System.

What is IDEFØ?

- **An IDEF method for modeling functions**
  - **Graphics (diagrams)**
  - **Text (glossary & narrative)**
- **Provides both a process and a language for constructing a model of the decisions, actions, and activities in an organization**

What is an IDEFØ Model?

- **A definition of activities and information**
  - **Within a particular *Context***
  - **Having a consistent *Viewpoint***
  - **For a particular *Purpose***
- **Series of diagrams (that decompose a subject into manageable chunks)**
- **A foundation for requirements specification, design, and programming**
- **A useful record throughout the life-cycle of an enterprise**

IDEFØ Diagram

- **Definition of activities performed**
- **Definition of information "Surrounding" the functions**

Example IDEFØ Diagram

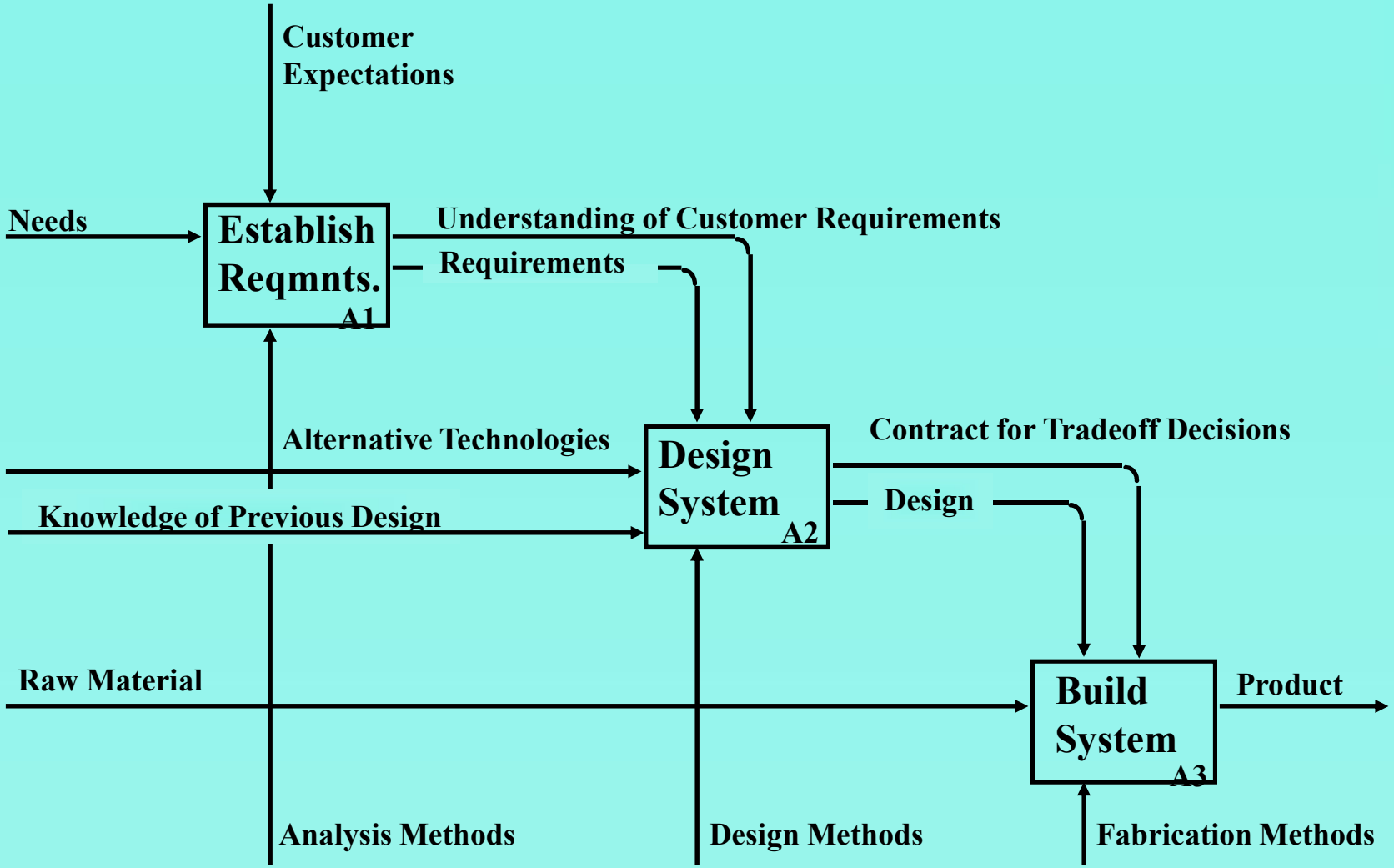
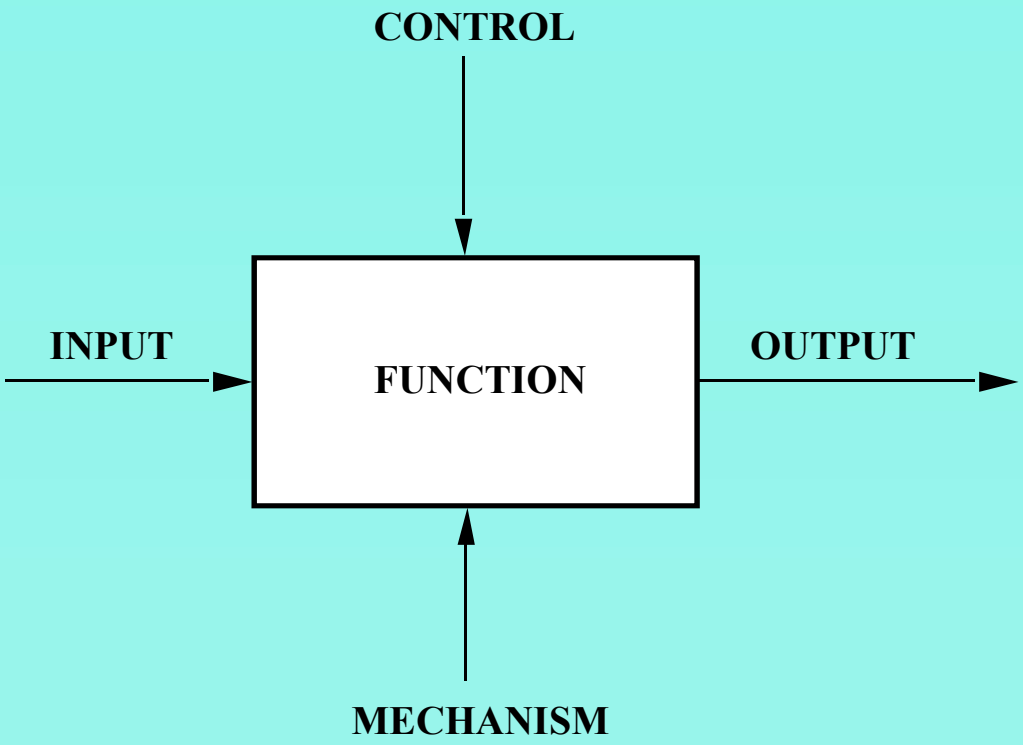


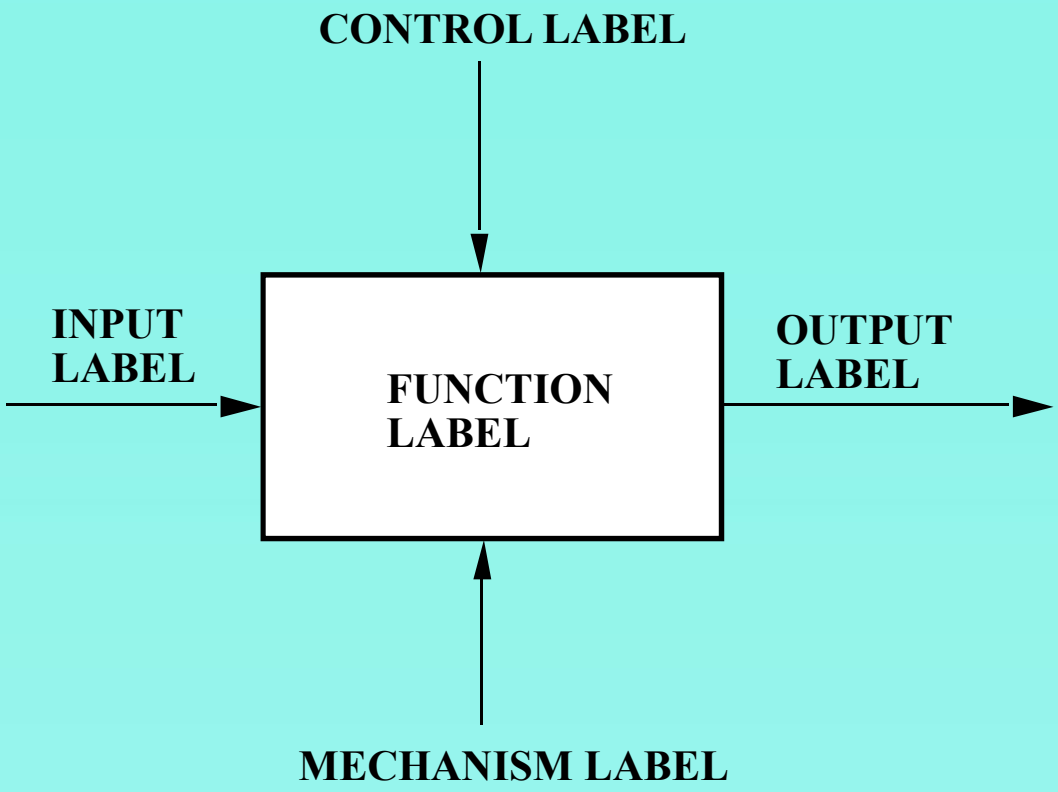
Diagram Construction (1)

- **Boxes represent functions**

- **Arrows represent real objects or data**



Labels


CONTROL LABEL
INPUT LABEL
FUNCTION LABEL
OUTPUT LABEL
MECHANISM LABEL

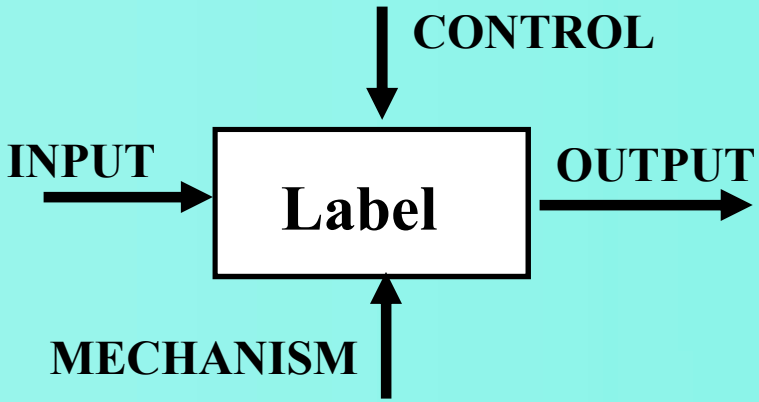


Diagram Construction (2)

- Labels are words that name functions and data/real objects
- Function labels are verbs or verb phrases and are put in the center of the function box
- Data labels are nouns or noun phrases
- Data labels name the input, control, output, and mechanism arrows

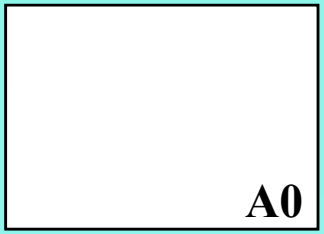
IDEFØ Function

- **An Activity, Action, Process, or Operation**
- **A Description of "What Happens" in a Particular Environment**
- **Accomplished by People, Machines, Computers**
- **Labeled with an Active Verb or Verb Phrase**

Function Label

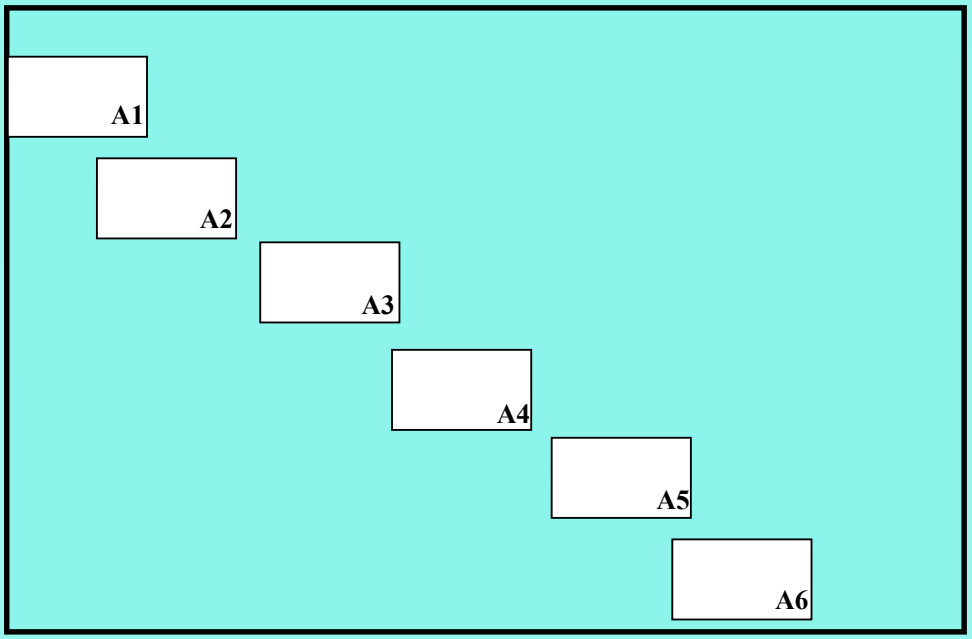
IDEFØ Functions (Activities)

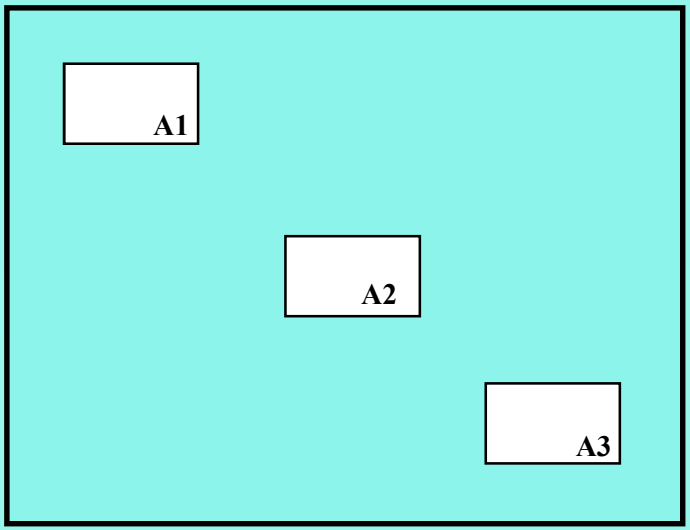
Represented as a box in an IDEF0 Model.



First diagram has one Function which bounds the context of the Model. (A - 0 diagram)

Diagram has a maximum of 6 functions & a minimum of 3





IDEFØ Relationships (Between Functions)

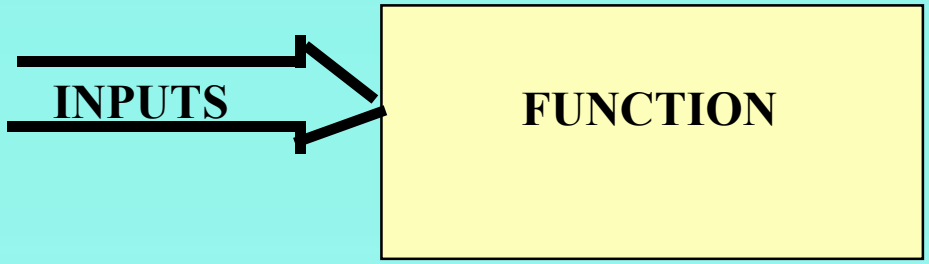
- Represented as arrows
- AKA concepts
- Real objects, data, people, machines, and computers

ICOMs

- **Inputs**
- **Controls**
- **Outputs**
- **Mechanisms**

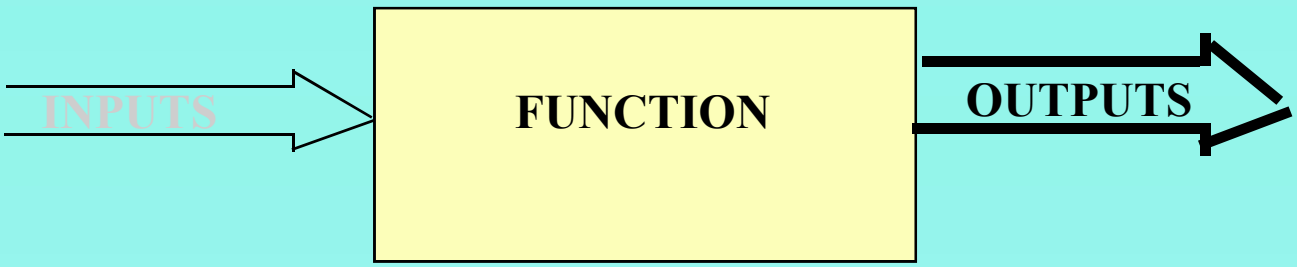
Inputs

- **Real Objects or Data Needed to Perform a Function**
- **Objects or Data Transformed by a Function**
- **Labeled with a Noun or Noun Phrase**



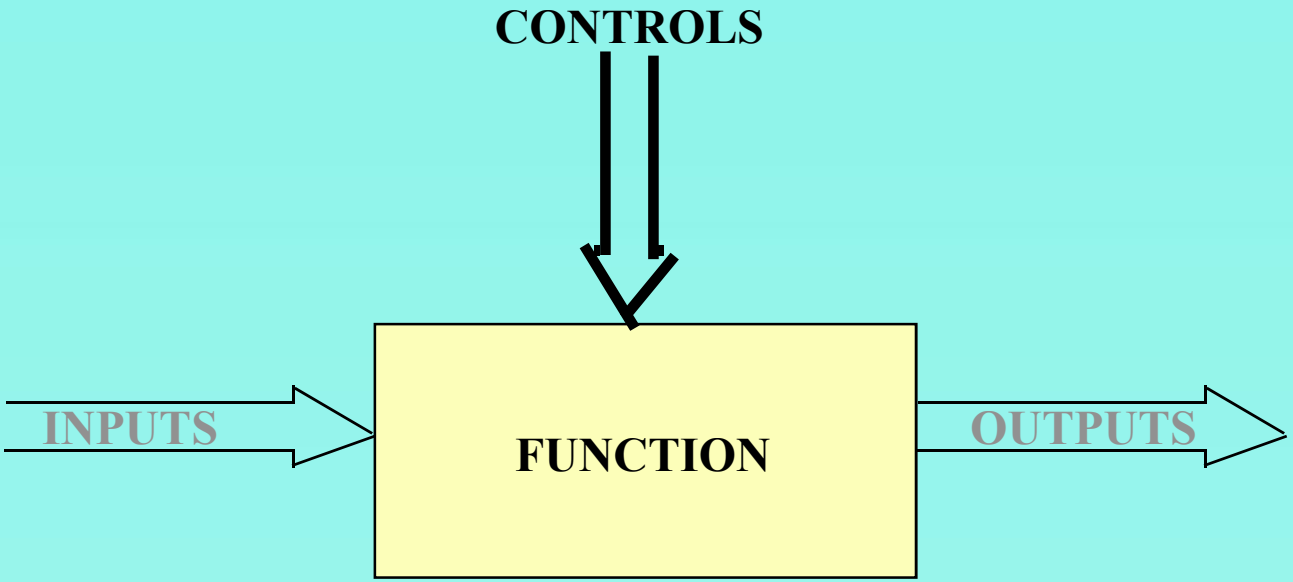
Output

- **Objects or Data Produced as a Result of the Function**
- **Labeled with a Noun or Noun Phrase**



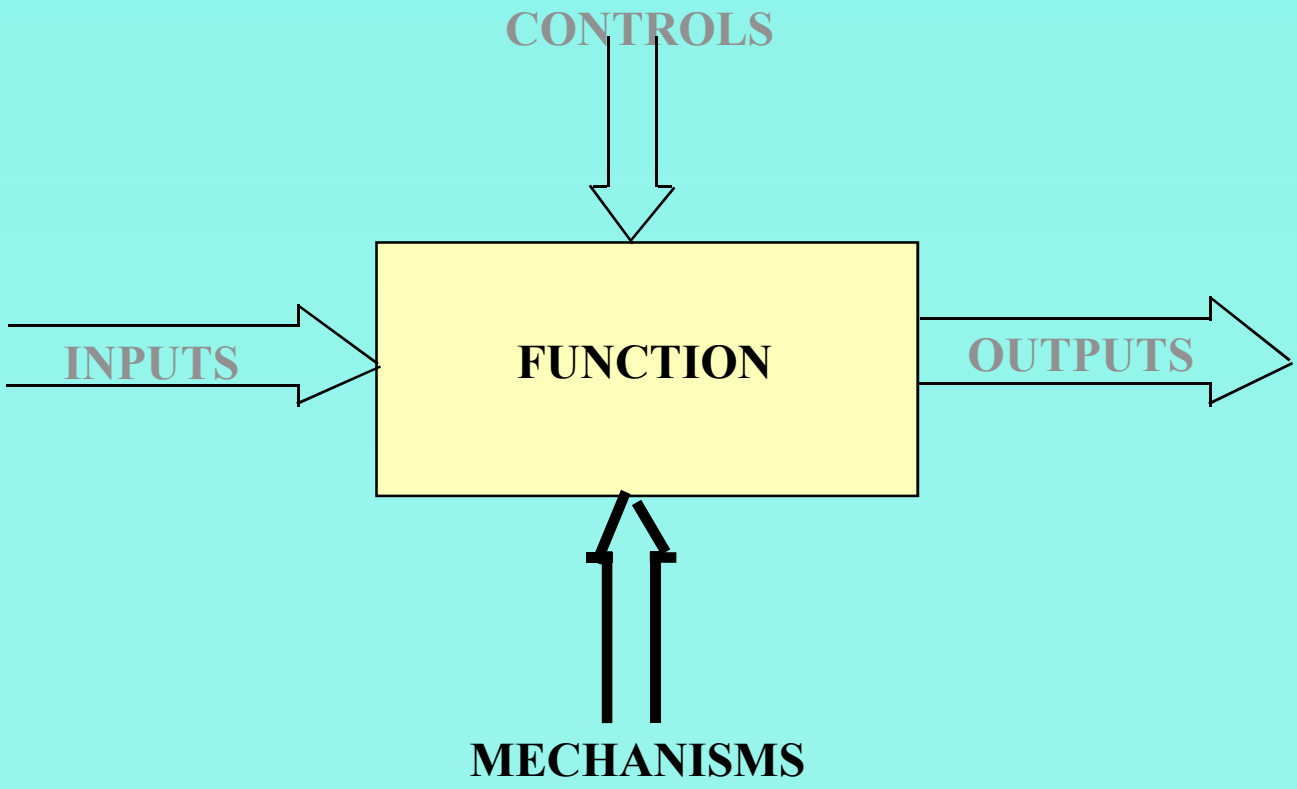
Control

- That which Governs the Accomplishment of the Function
- Things that Influence or Determine the Outputs
- Labeled with a Noun or Noun Phrase

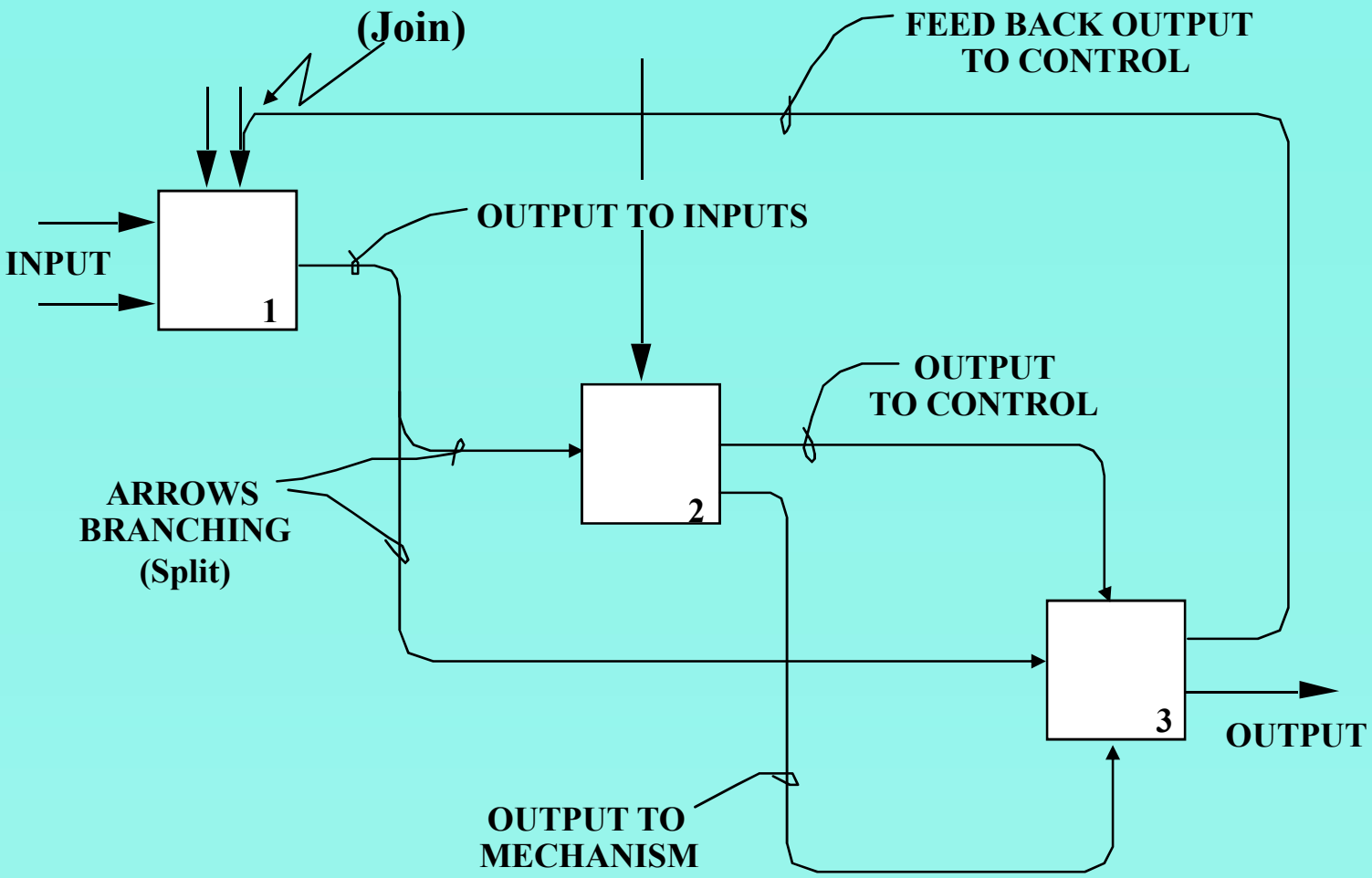


Mechanism

- **Person, Device, or Data which Carries out the Function**
- **The Means by which the Function is Performed**
- **Labeled with a Noun or Noun Phrase**

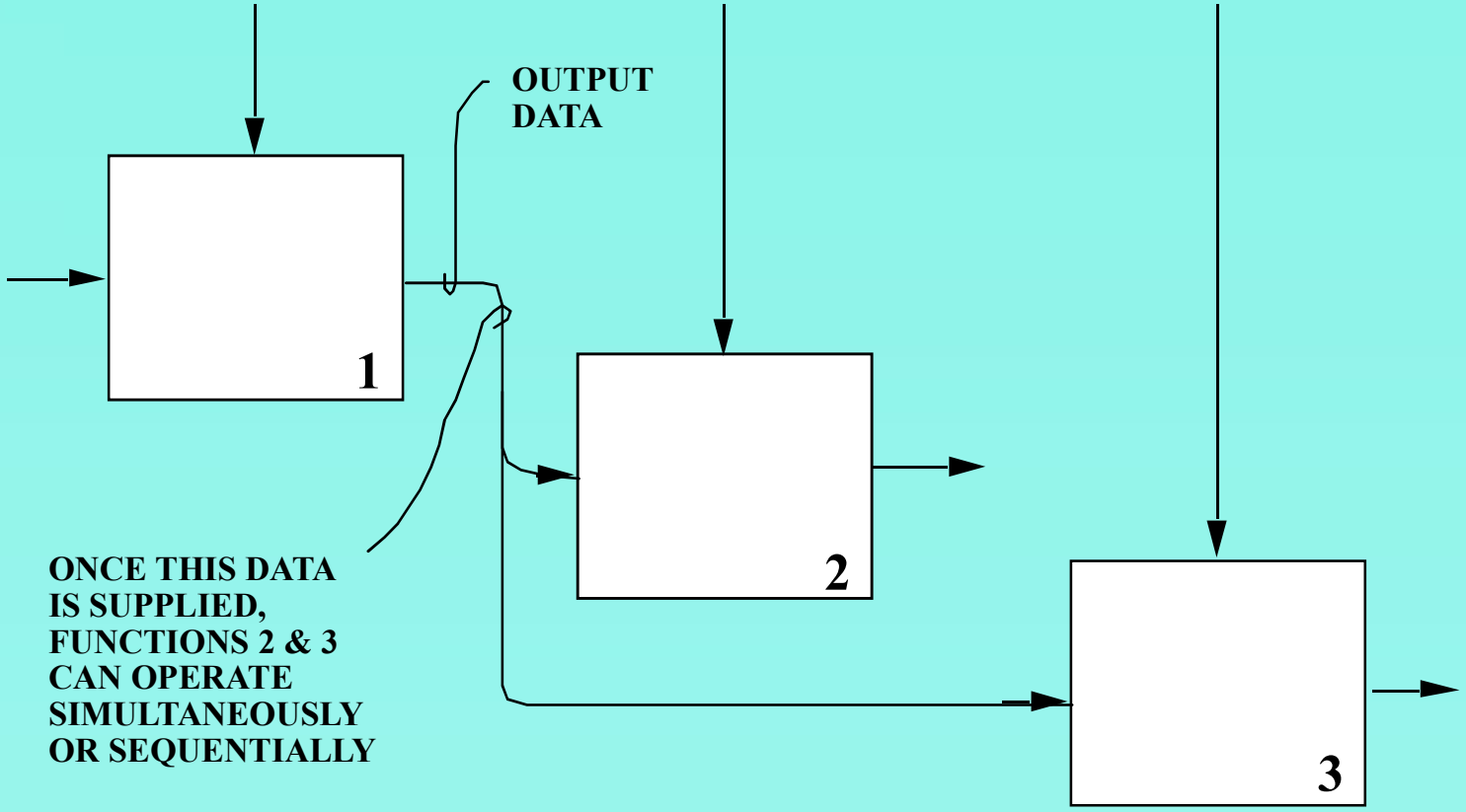


Box and Arrow Relations in a Diagram



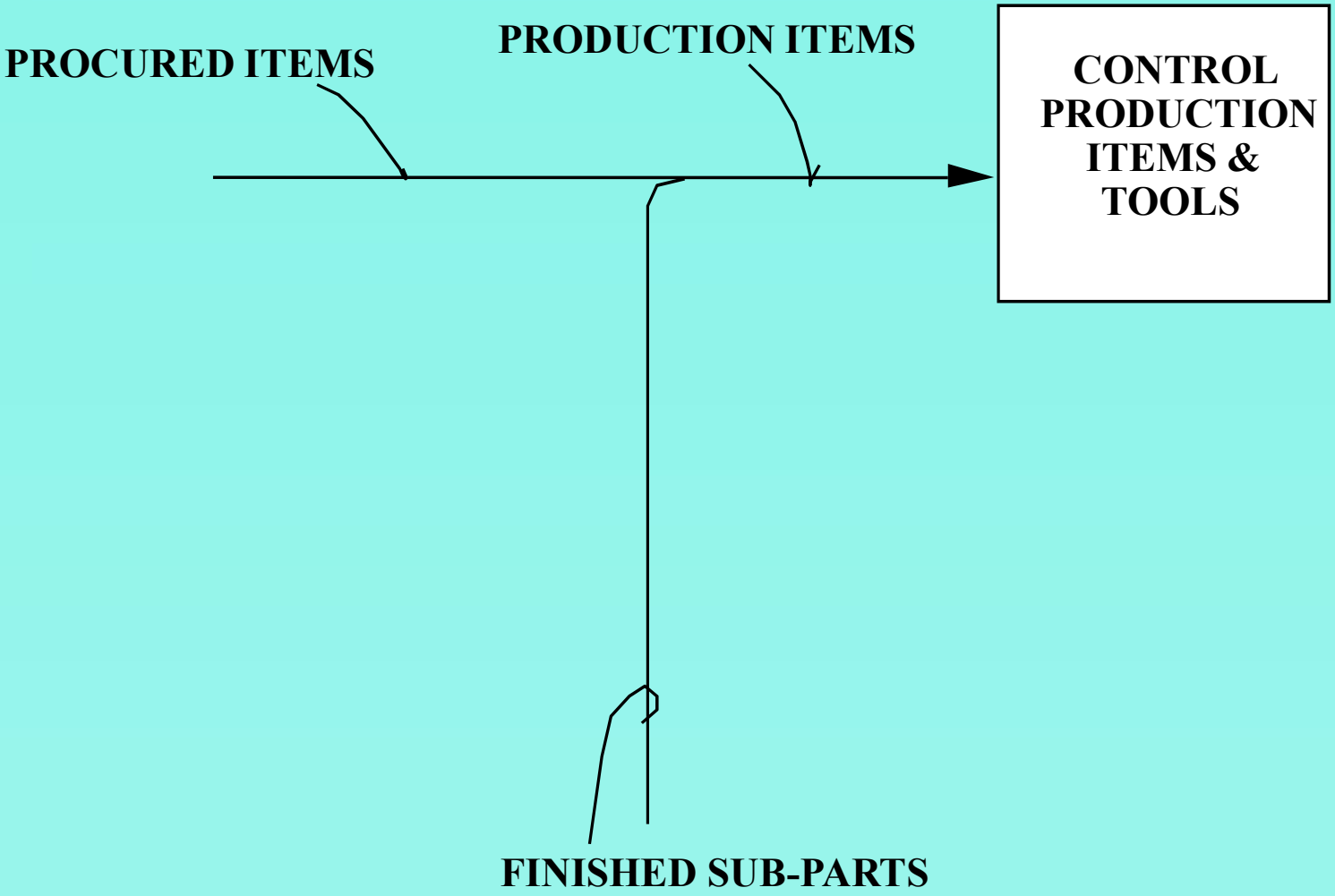
Arrows: "Branching"

Output can branch and be used by two functions simultaneously or sequentially

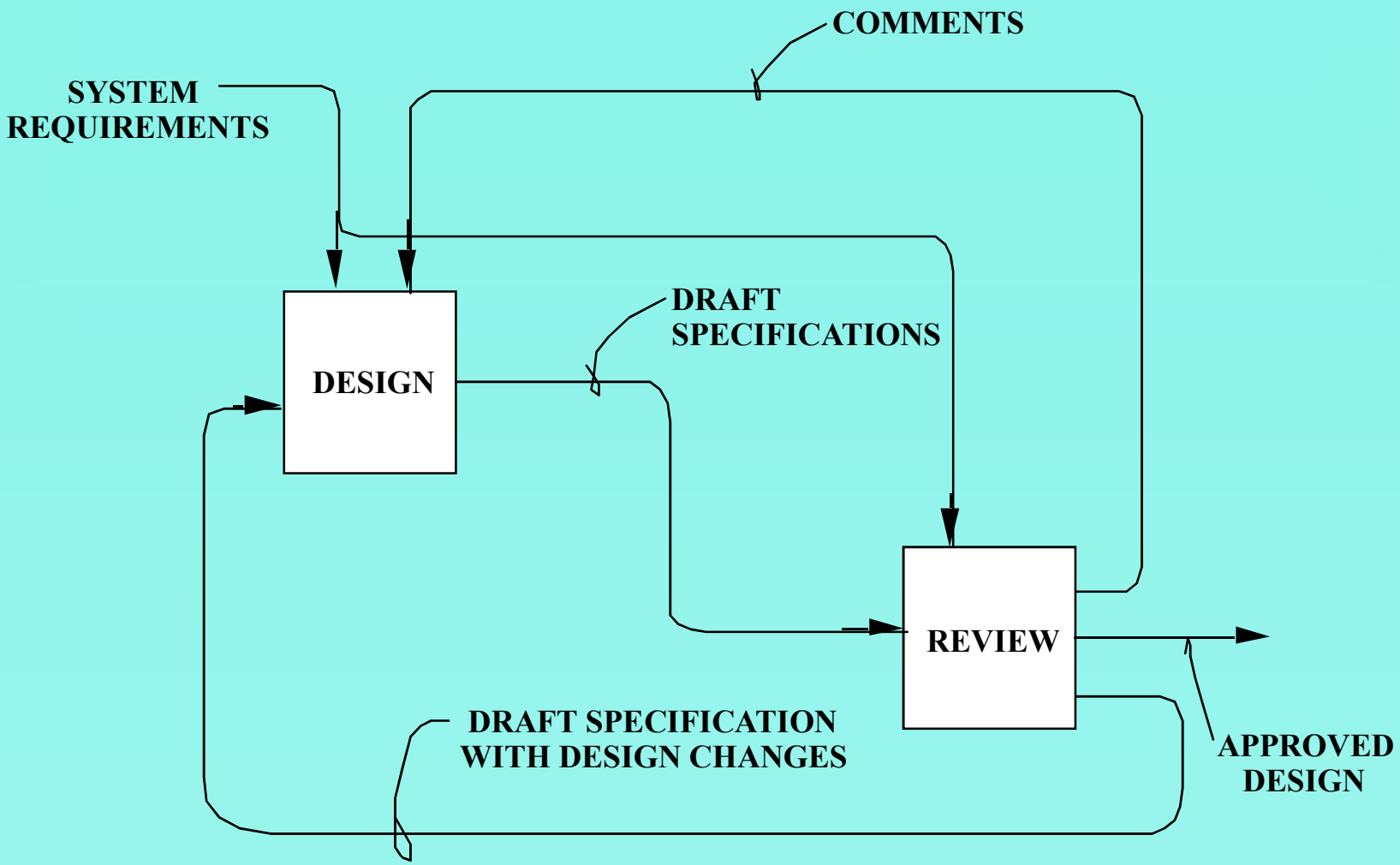


Without labels we cannot tell how the branching occurs

Arrows: "Joining"

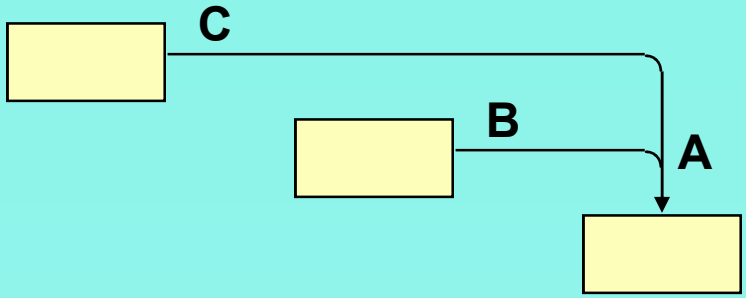


Arrows: "Feedback"



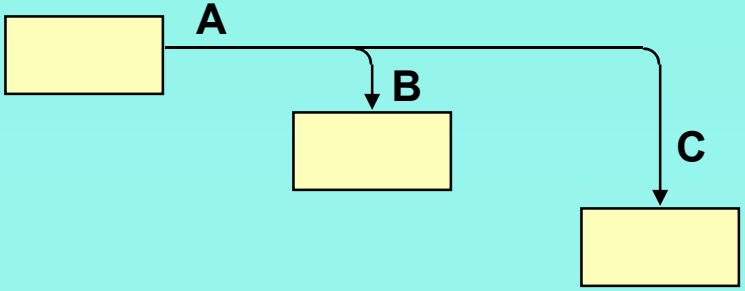
Bundling and Unbundling

Bundle: Concepts B and C are bundled to form concept A.



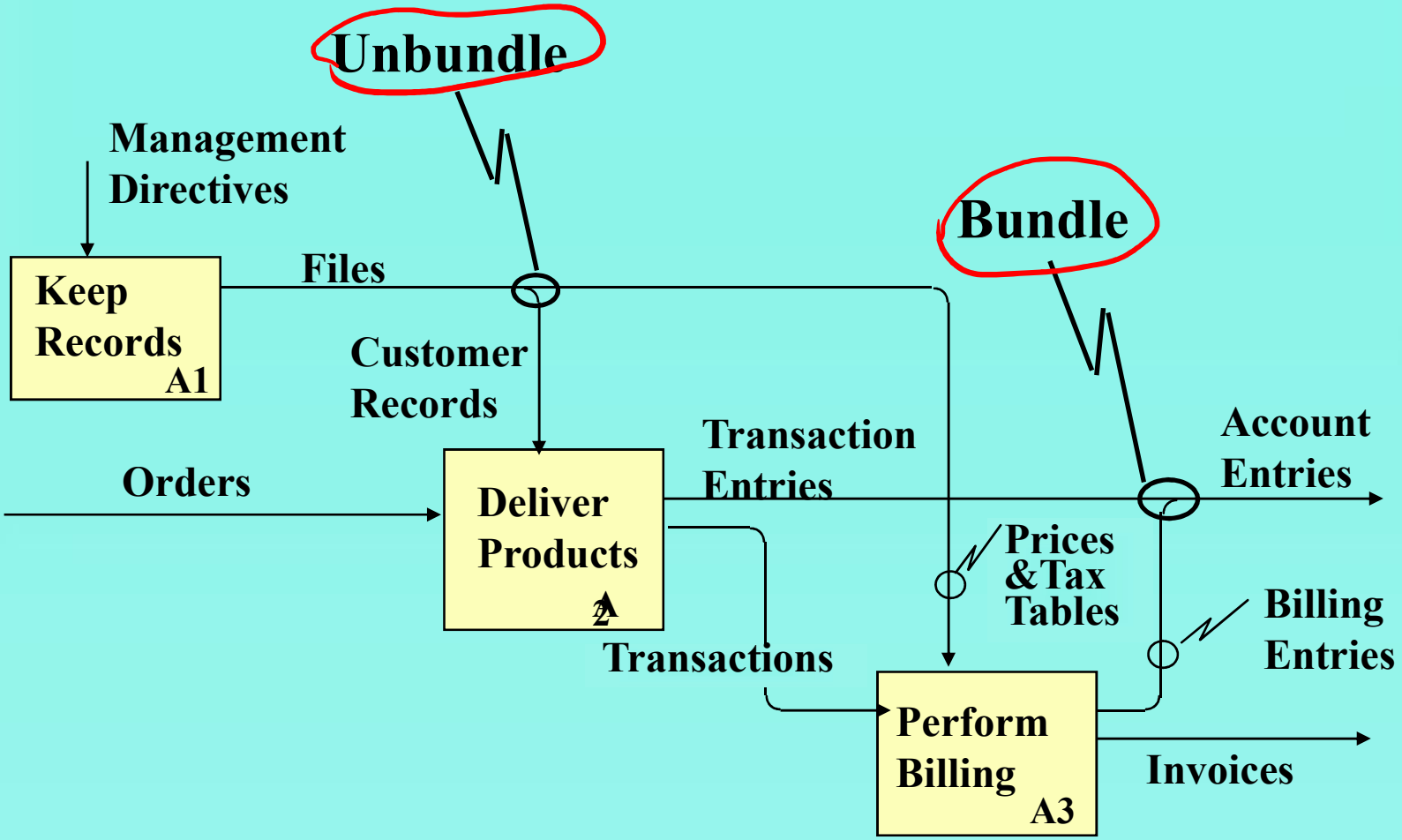
2 join into 1

Unbundle: Concept A is unbundled into concepts B and C.



1 split to 2

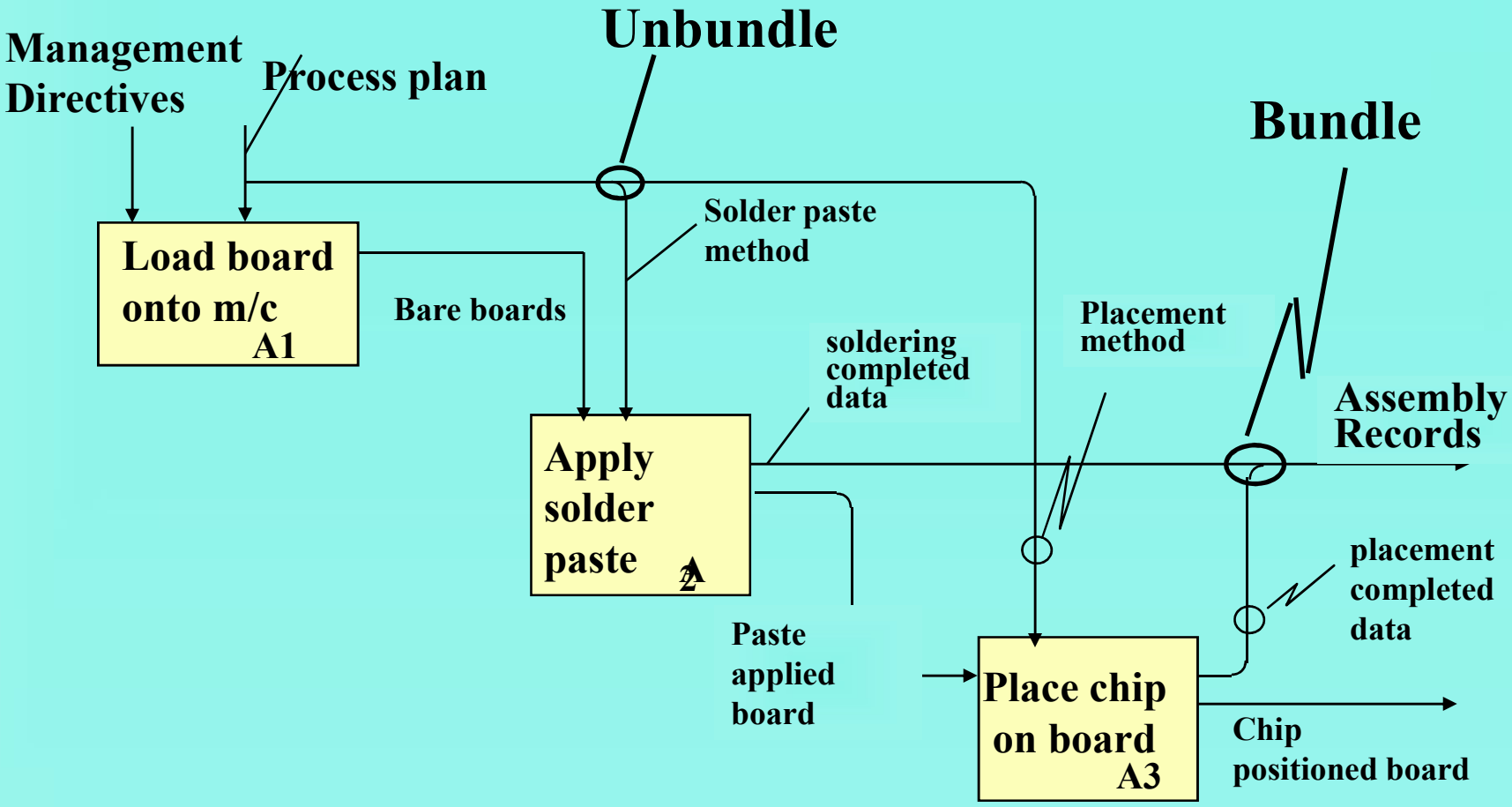
Bundles and Unbundles



Files = Customer Records + Price & Tax Tables

Account Entries = Transaction Entries + Billing Entries

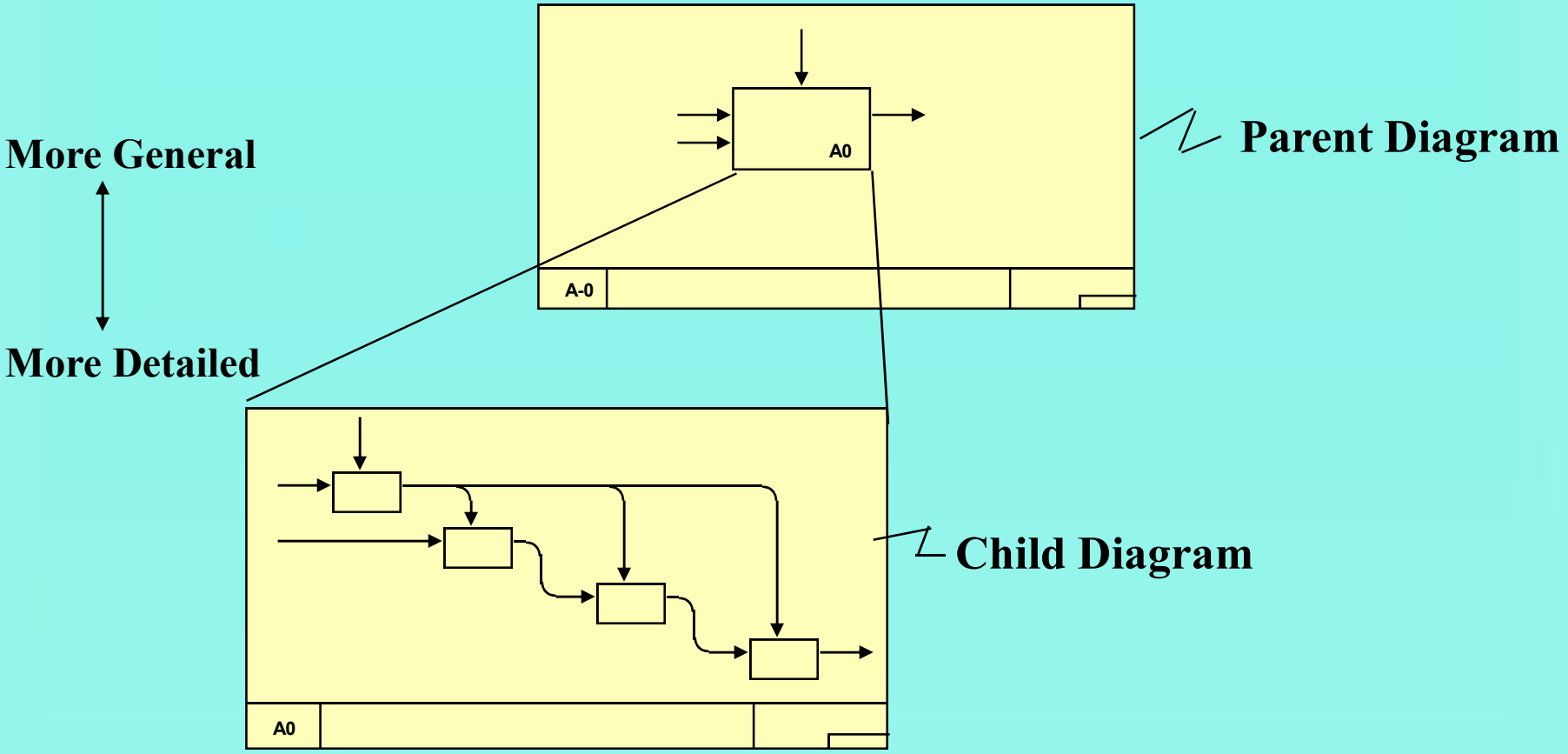
Bundles and Unbundles: PCB ASSEMBLY



Process Plan = loading details + solder paste details + chip placement method

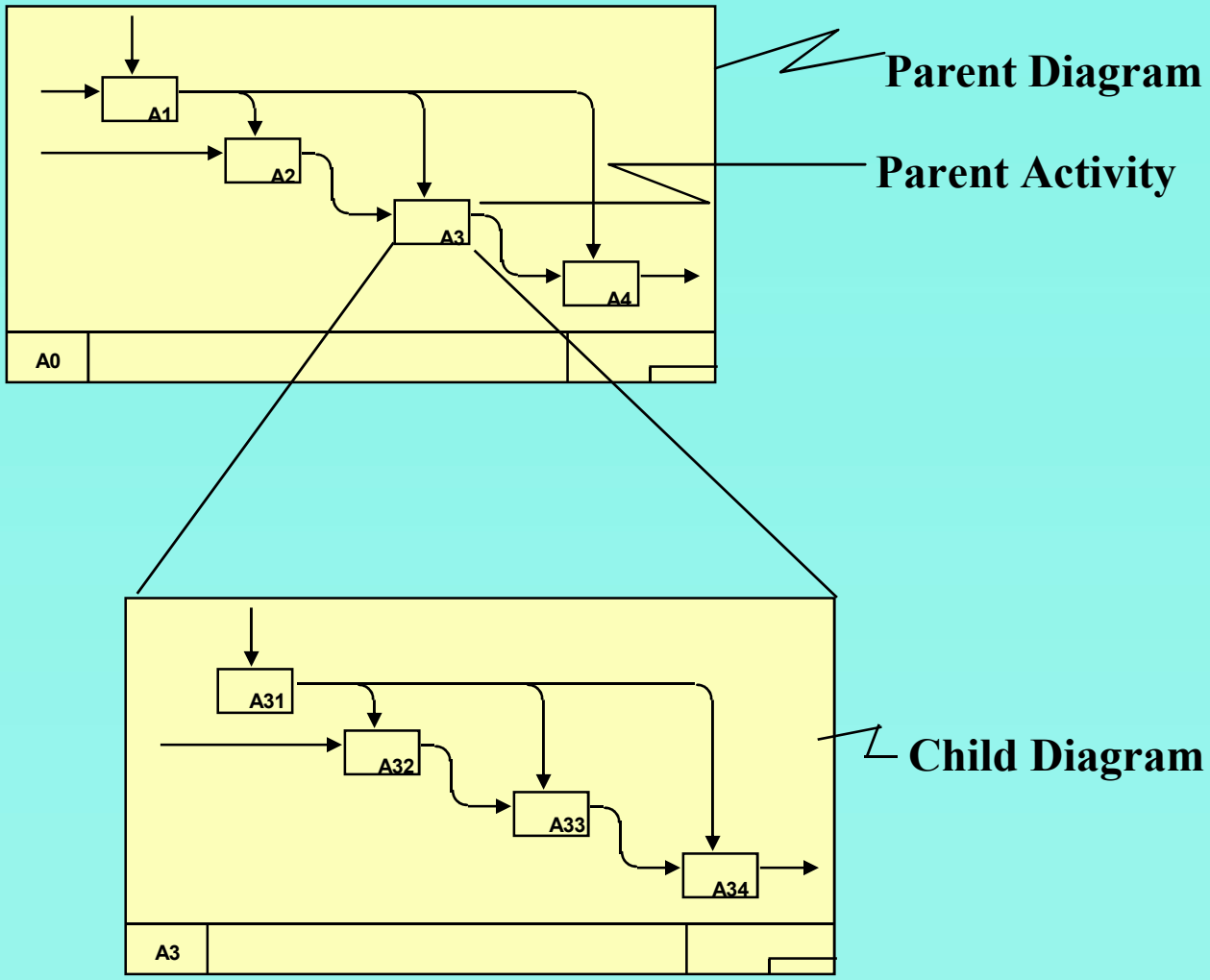
Assembly Records = soldering completed data + placement completed data

Function Decomposition



"Parent" Activities Represent a Higher Level of Abstraction than that of Their "Children"

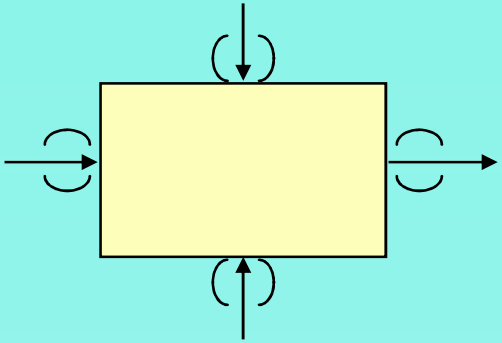
Further Decomposition



Decomposition

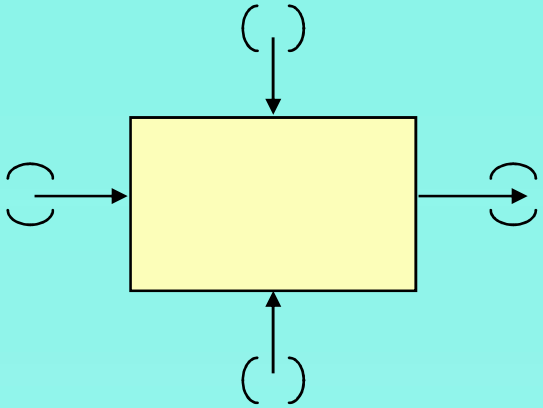
- **Establishes model hierarchy**
- **Functions are comprised of other functions**
- **Decompositions is a process of breaking down of the functions (level-by-level)**
- **Data consistency is required throughout the level-by-level decomposition breakdown**

Complexity Simplification Technique
Tunneled Arrows



Tunneled Arrows at Connected Ends

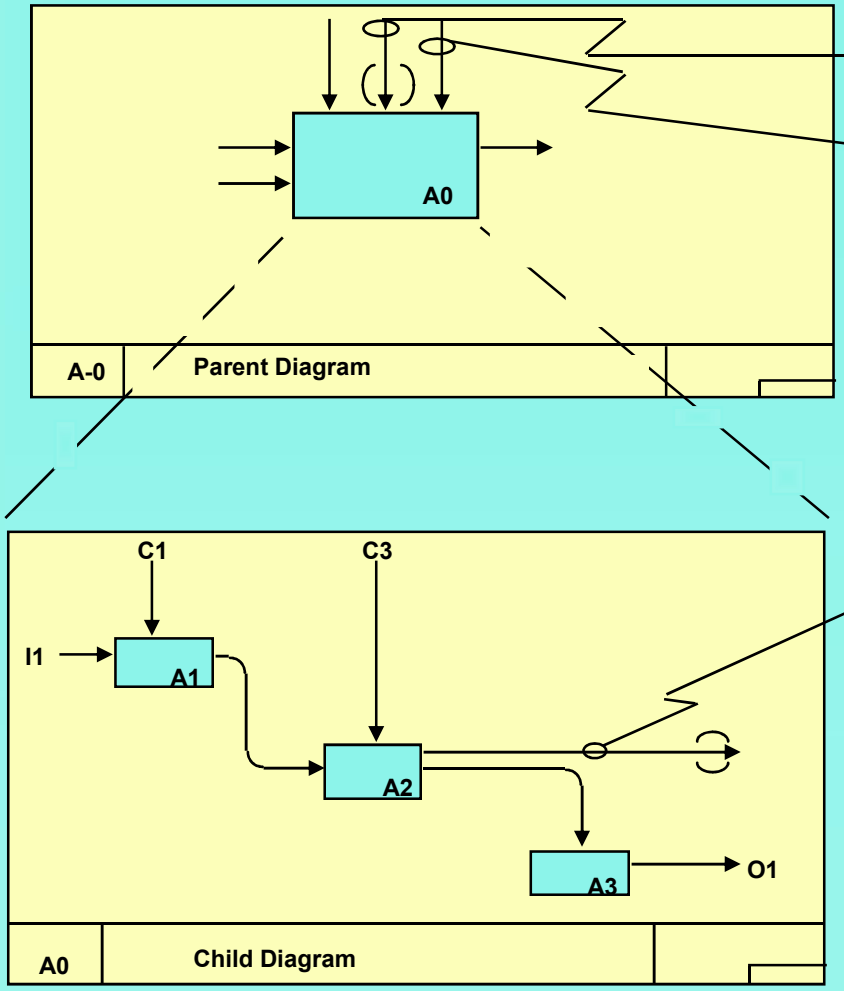
(Concept Does Not Appear on the Next Lower Level.)



Tunneled Arrows at Unconnected Ends

(Concept Does Not Appear on the Next Higher Level.)

Tunneling Example



This control will not appear on child diagram.

This control will still be designated as C3 on child diagram.

This output will not be shown on parent diagram.

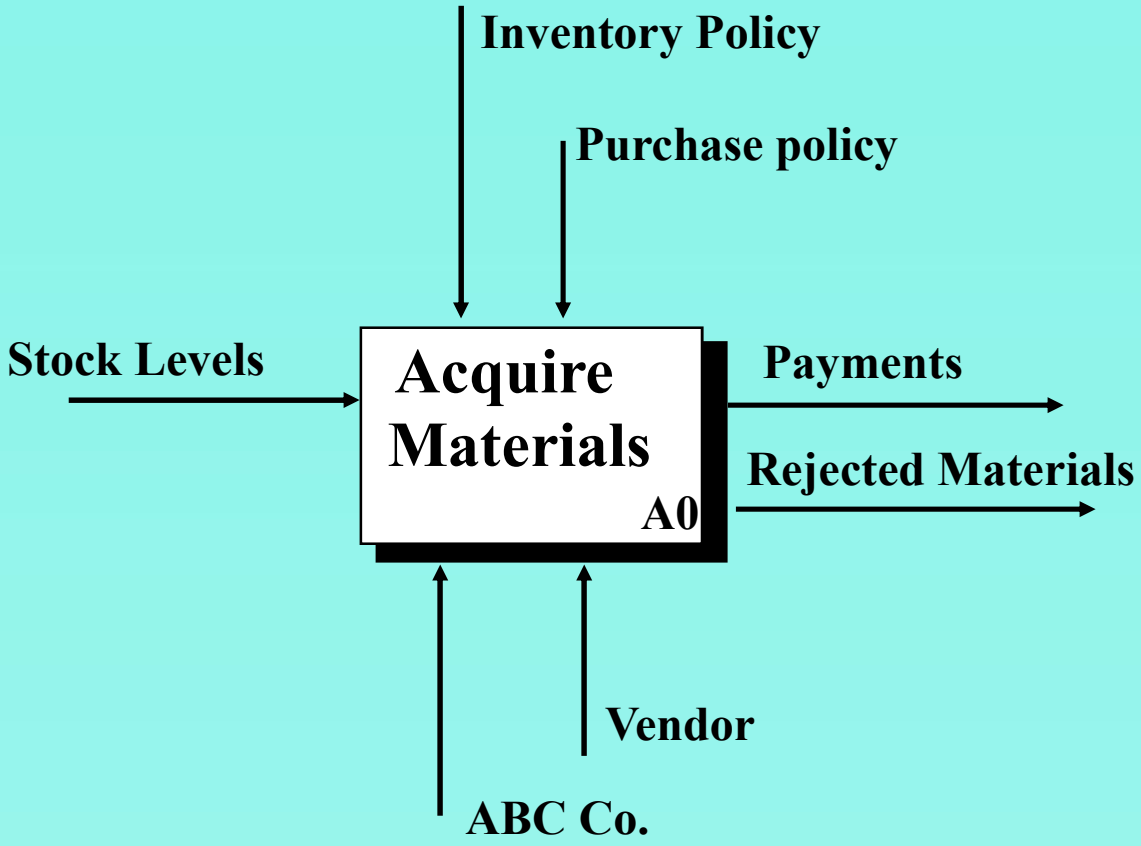
Steps in Building a Model

- **1. Define Viewpoint, Purpose, and Context**
- **2. Develop the Context Diagram (Putting the situation in context)**
- **3. Decompose activities to fit scope of modeling task (complete modeling per rules, etc)**
- **4. Develop glossary**

Model Orientation!!!!

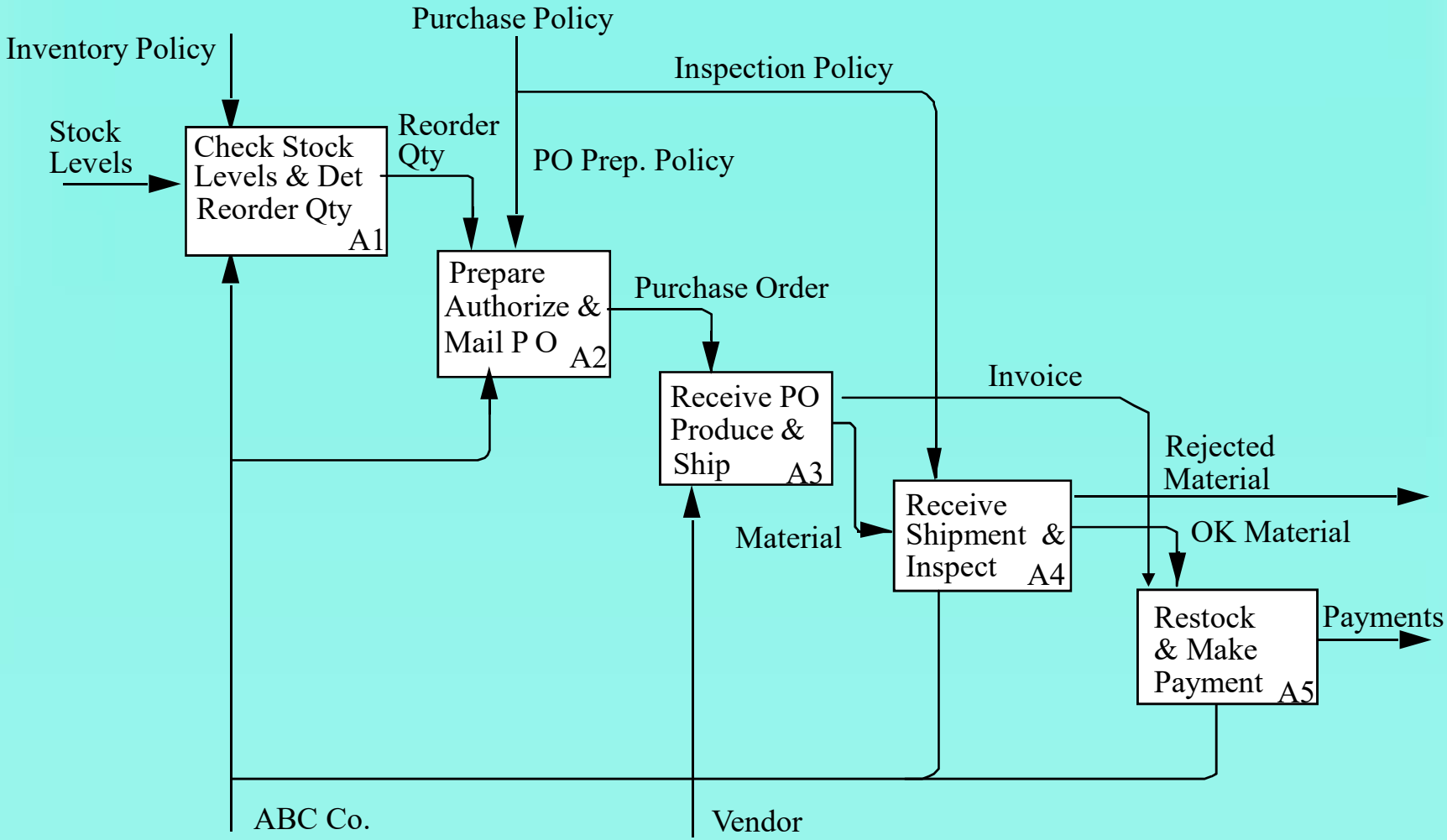
- **Context (Subject)**
  The Boundaries of the Subject Matter
- **Viewpoint (Bias)**
  The Perspective from which a Subject is Analyzed
- **Purpose (Objective)**
  The Reason(s) a Model is Created

Example - Context Diagram



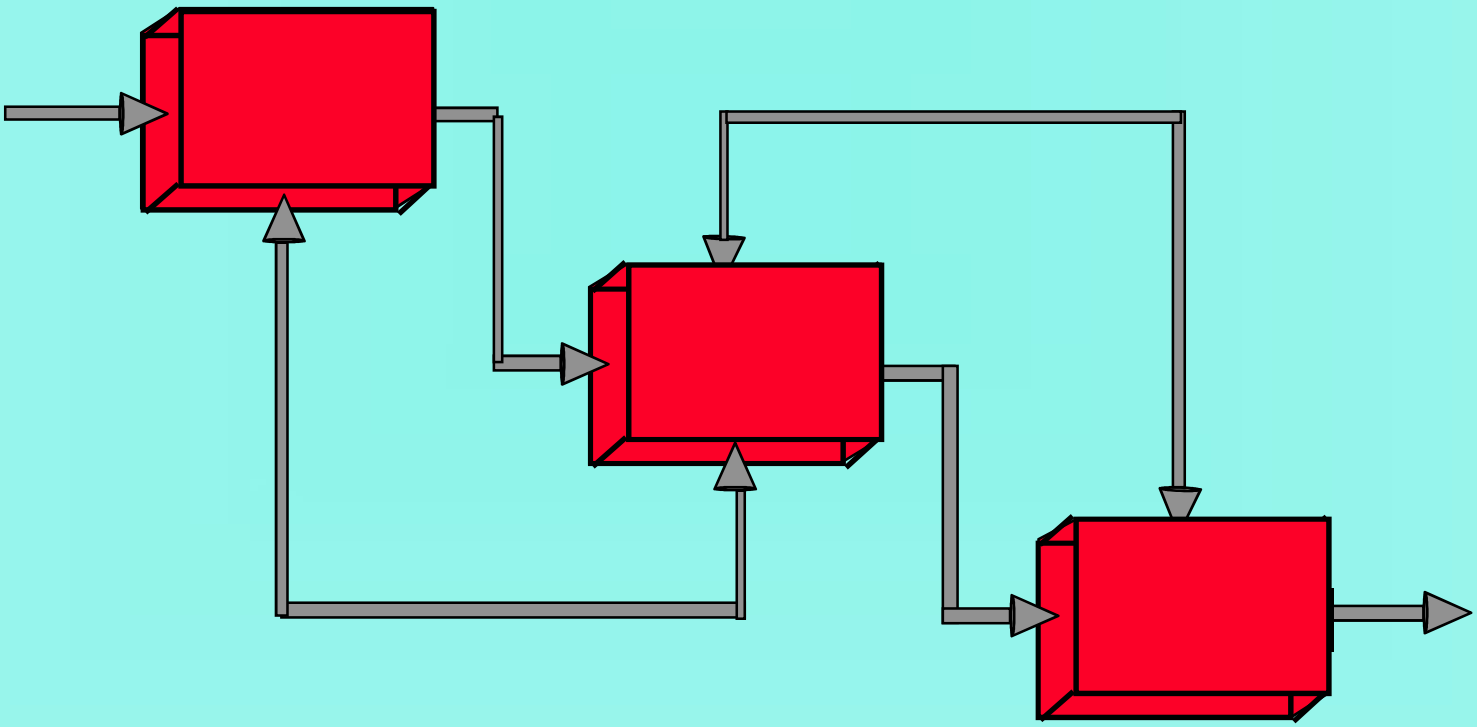
A-0 Diagram

Example - Decomposition of the Context Diagram



A0 Diagram

IDEFØ Captures What an Enterprise Does



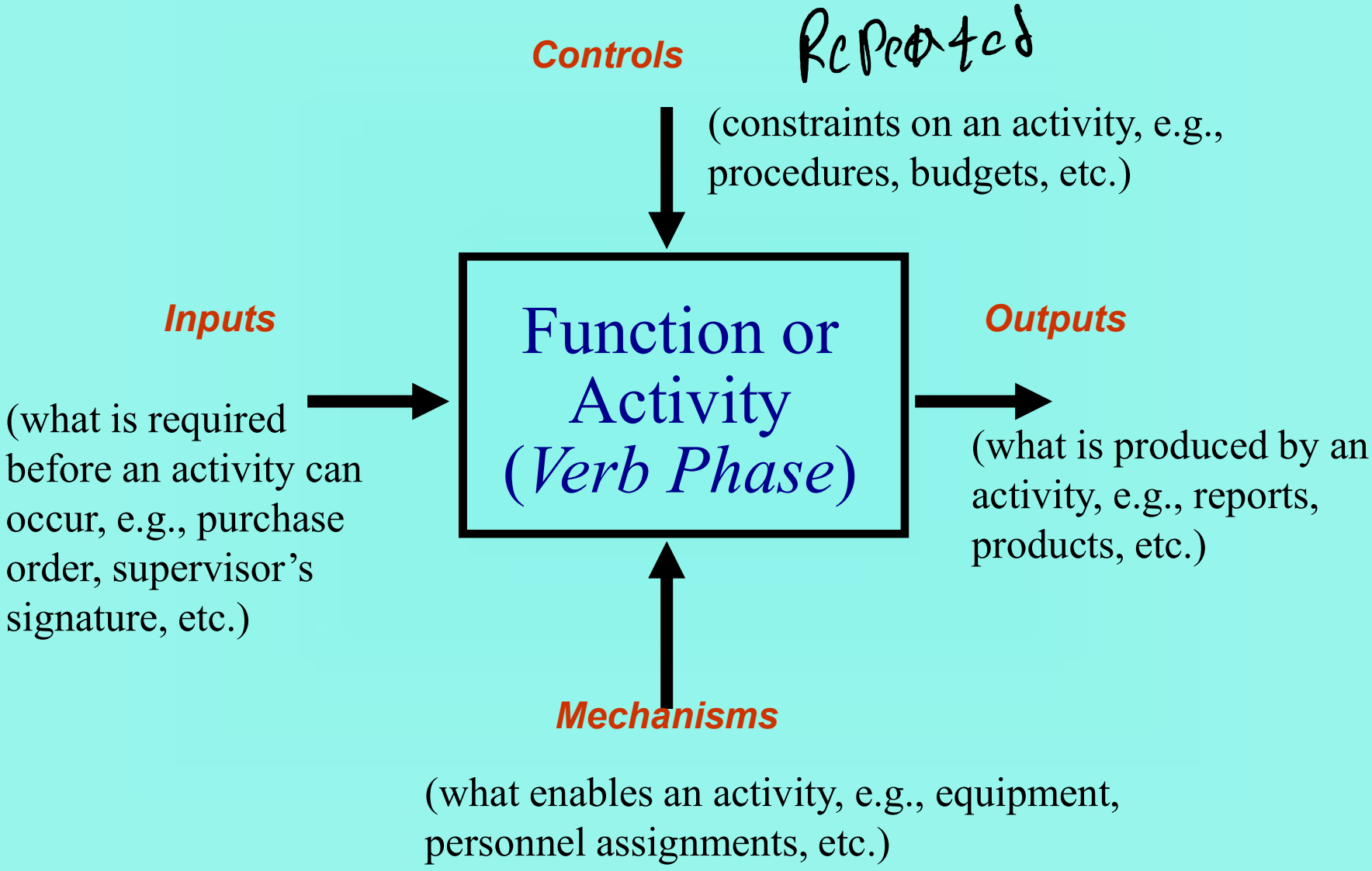
Why Develop An IDEFØ Activity Model?

- **To identify, document, and communicate an enterprise's core activities.**
- **To understand how activities relate to one another.**
- **To identify value-added and non-value-added activities.**
- **To identify activities that need to be improved.**

Benefits of Activity Modeling

- **Documents current activities.**
- **Reduces the learning curve for new activity users.**
- **Captures and analyzes As-Is activities.**
- **Facilitates the design/redesign of activities for To-Be scenarios.**

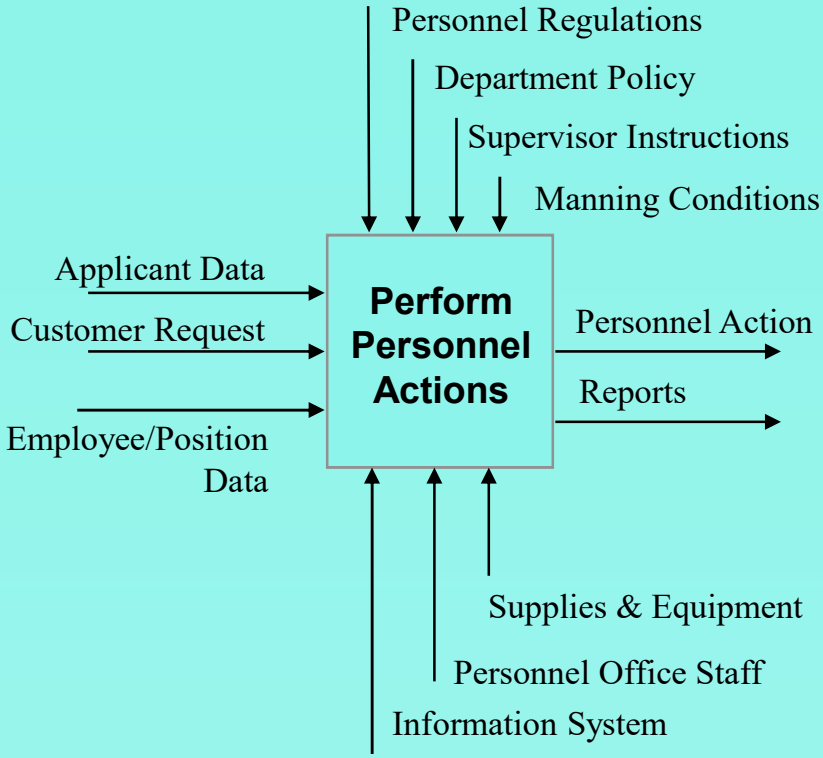
An IDEFØ Activity Box



Context, Purpose, and Viewpoint: CONTEXT

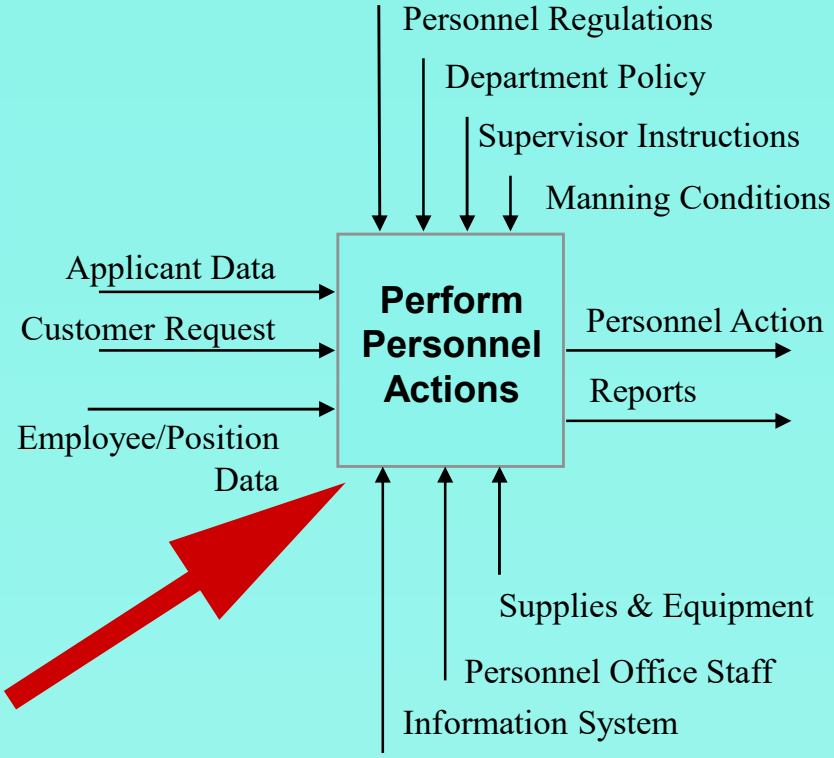
The context defines the boundaries of your model—i.e., what will be included in the model.

For example, Employee/Position Data comes from outside the model.



Context, Purpose, and Viewpoint:

PURPOSE



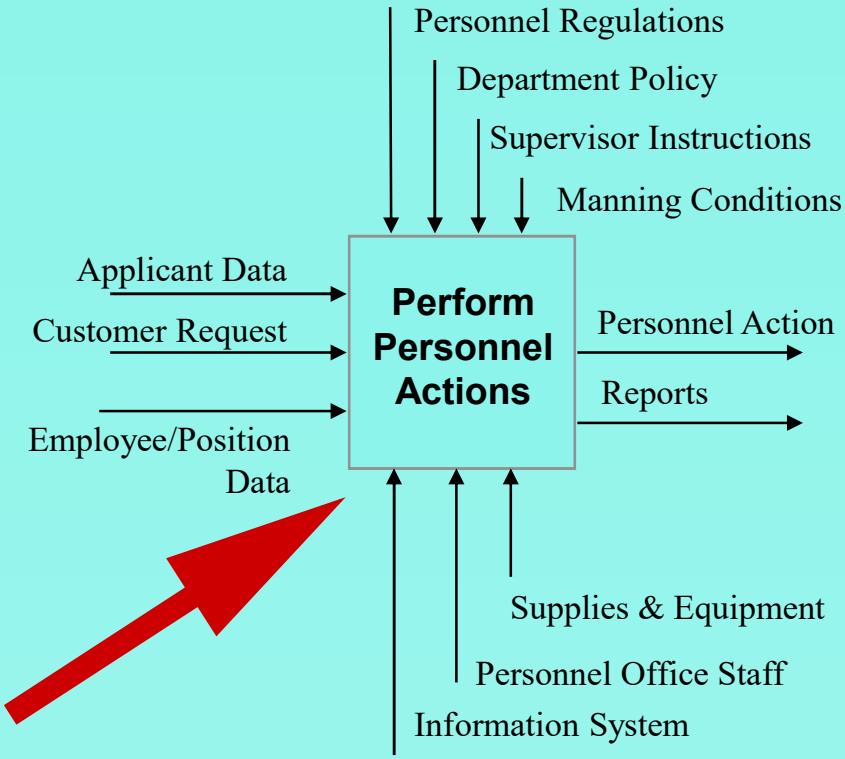
We define purpose as the reason to develop this particular activity model.

Purpose: To document the activities associated with managing Personnel Actions and identify non-value-added activities that might be eliminated.

Context, Purpose, and Viewpoint:

VIEWPOINT

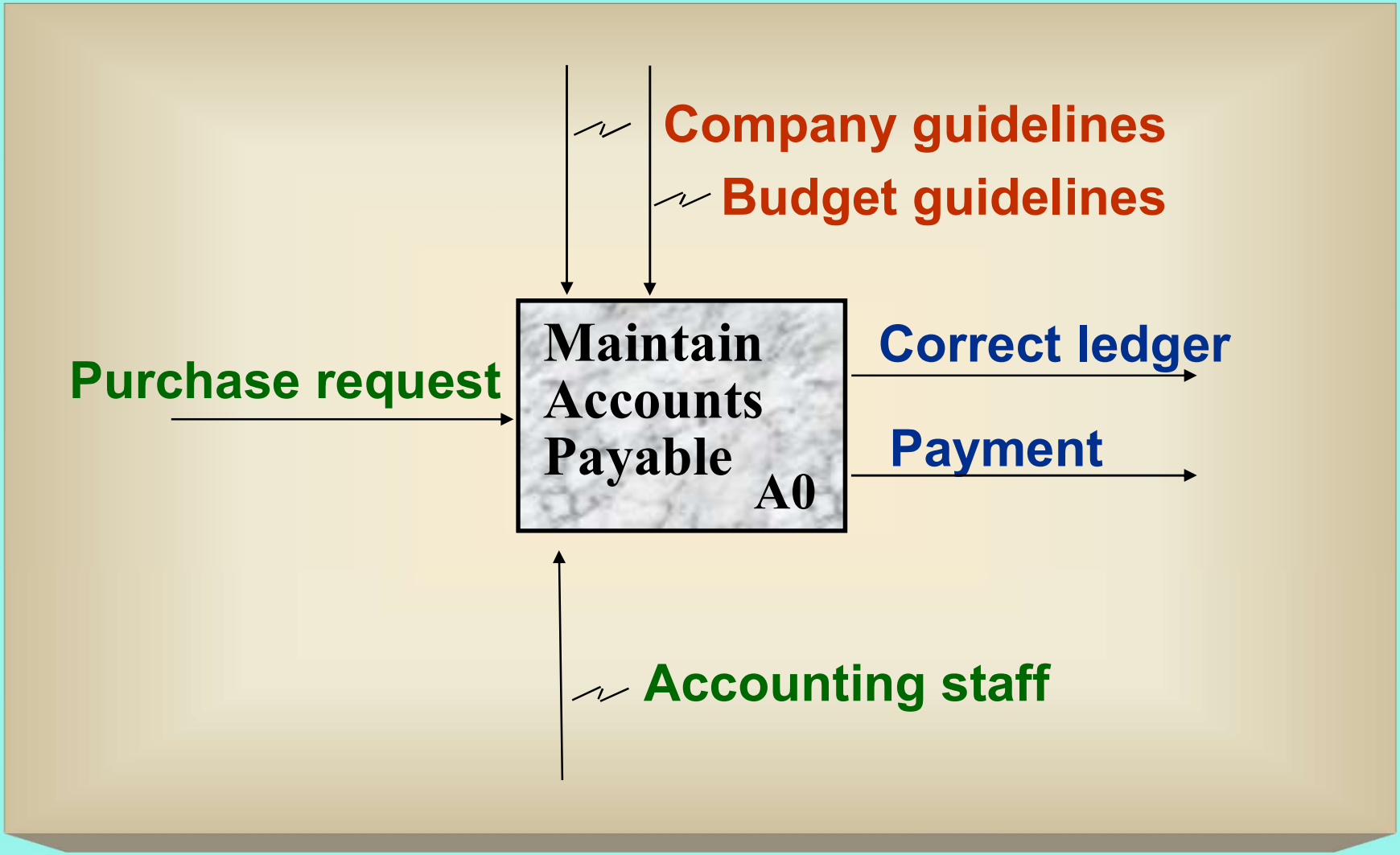
Viewpoint can be thought of as the perspective of the person/group developing the model.



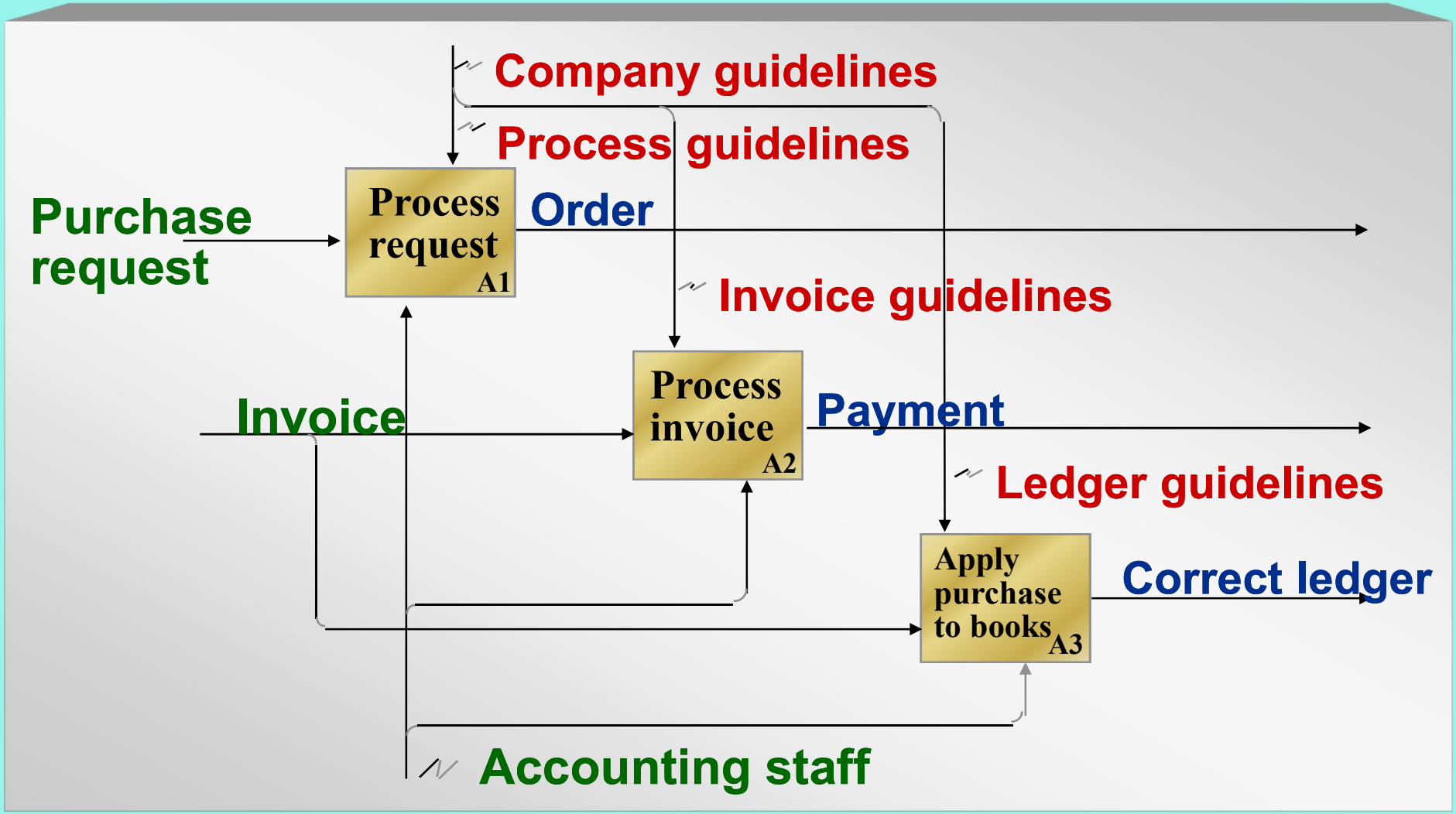
Viewpoint:

Personnel Officer

Decomposition: An Example



Decomposition: An Example



Function Model for Planning and Implementing a Feat Ext module

- **Purpose: To obtain a better understanding of the various tasks involved in planning and implementation of a feature extraction module**
- **Context: We will assume CAD model formats, process planning requirements and resources available (people and computers) are known. The FE module will be built using available existing resources (no new tools or software will be purchased).**
- **Viewpoint: that of an industrial / mfg engineer who has a background in designing / building software systems**