

King Saud University

College of Engineering

IE – 462: “Industrial Information Systems”

Spring – 2019 (2nd Sem. 1439-40H)

[Chapter 4:](#)

Structured Analysis and Functional

Architecture Design – p1 – IDEF0 – i - Fundamentals

Prepared by: Ahmed M. El-Sherbeeney, PhD

Lesson Overview

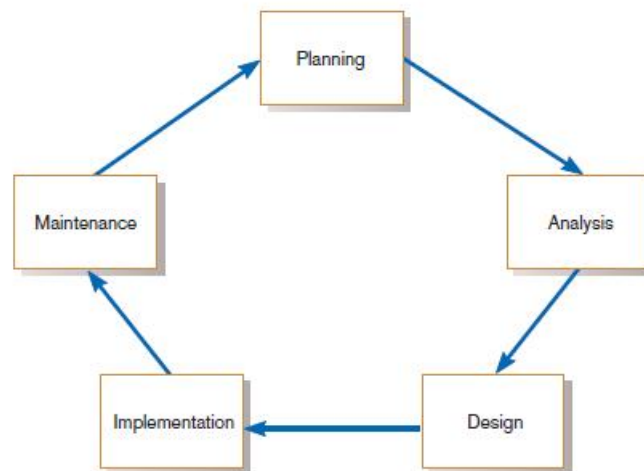
- [Modeling IIS](#) – (p1)
- Integrated Computer-Aided Manufacturing Definition 0 (IDEF0) – (p1)
 - i. [IDEF0 Methodology – Modeling Primitives](#)
 - ii. Case Study
- Data Flow Diagram (DFD) – (p2)

Modeling Industrial Information Systems



Modeling IIS

- **REMEMBER: Design** – a description of the recommended solution is converted into *logical* and then *physical* system specifications
 - **Logical design:** all *functional features* of the system chosen for development in analysis are described *independently* of any computer platform
 - **Physical design:** transforming the logical specifications of the system into *technology-specific details*



Modeling IIS – cont.

- This is *first step* in design of IIS for an industrial enterprise
- The design proceeds from a **definition of a business model** of the enterprise
- This business model (**IS model**) is a description of the:
 - **functions of the business** (or manufacturing operations)
 - **data requirements**, and
 - **interactions** between the functions and data requirements

Modeling IIS – cont.

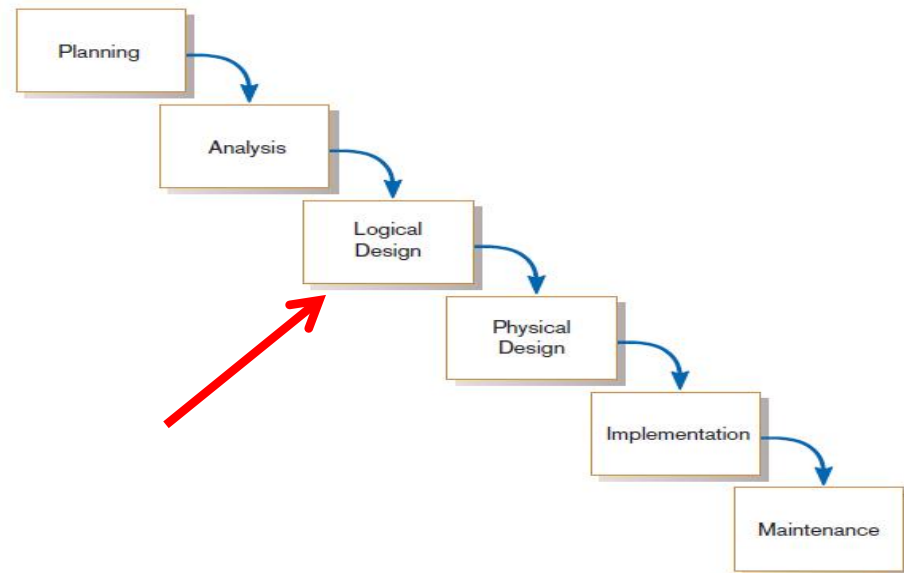
- **Functional or Activity architecture** describes:
 - a conceptual model of the activities that operate the business (or manufacturing operations) *and*
 - the relationships between those activities
- The word **architecture** denotes the fact that the model has a *layered structure*
- A related conceptual model of the firm is the **Informational or Data architecture**:
 - this is a model of the *information requirements* needed to perform the functions of the business

Modeling IIS – cont.

- Functional and informational architectures, when taken together:
 - form a *high-level blueprint* for the implementation of computer integration in the enterprise

Modeling IIS – cont.

- There are **three layers** of system design to consider in an information system project:
 - Conceptual
 - Implementation and
 - Execution
- Conceptual layer consists of **logical** design of functional and data requirements



Layers of IS Design Process

Conceptual Layer

Functional
Architecture

Informational
Architecture

Implementation Layer

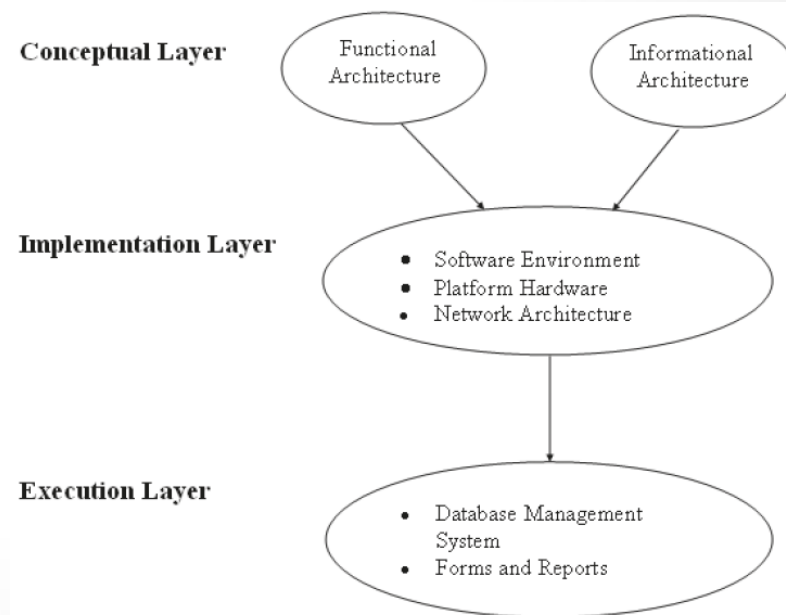
- Software Environment
- Platform Hardware
- Network Architecture

Execution Layer

- Database Management System
- Forms and Reports

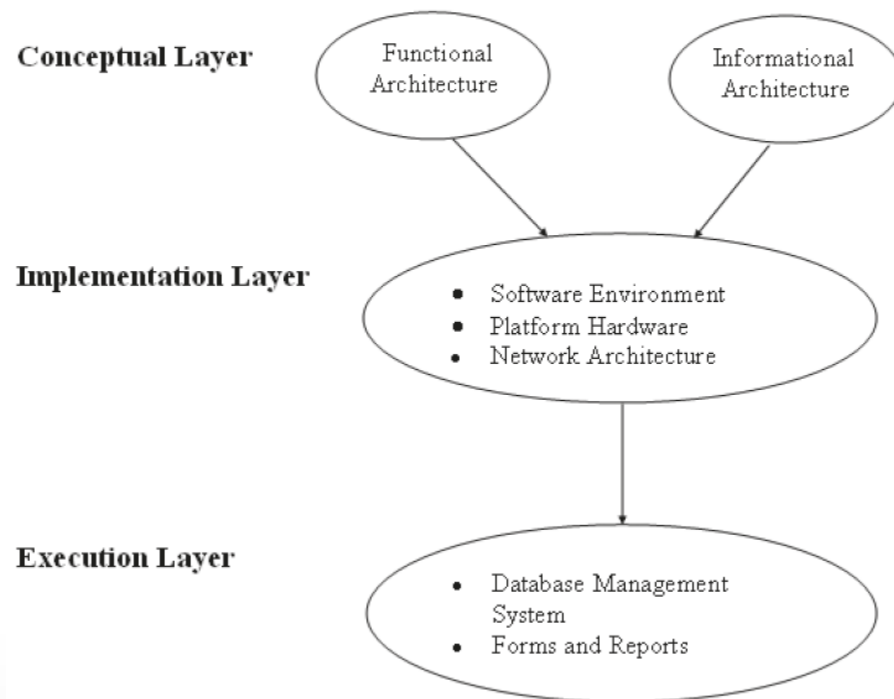
Modeling IIS – cont.

- When a conceptual design is complete, the next step is to **implement** the blueprint as hardware and software
- This is the *implementation layer* that requires the selection of:
 - database management system (DBMS)
 - hardware platforms, and
 - communication medium



Modeling IIS – cont.

- At the **execution** layer, the conceptual model and implementation techniques are coded in **software** in terms of forms and reports
- *Forms and reports* are used to interface with the individuals performing the functions defined in the functional architecture



Modeling IIS – cont.

- Two methodologies for designing a functional architecture, known as “structured analysis” techniques:
 - ***data flow diagrams*** (1979), widely used by information system professionals in all industries
 - ***structured analysis and design technique*** (SADT) (1988), adapted for manufacturing enterprises under the name ***integrated computer-aided manufacturing definition 0*** (IDEF0)
 - Both methodologies are based on *graphical notations* used to describe information flows among processes of the enterprise being documented (described next)

Functional/Process Modeling

1. Integrated Computer-Aided Manufacturing Definition 0 (IDEF0)

IDEF0 Methodology – Modeling Primitives

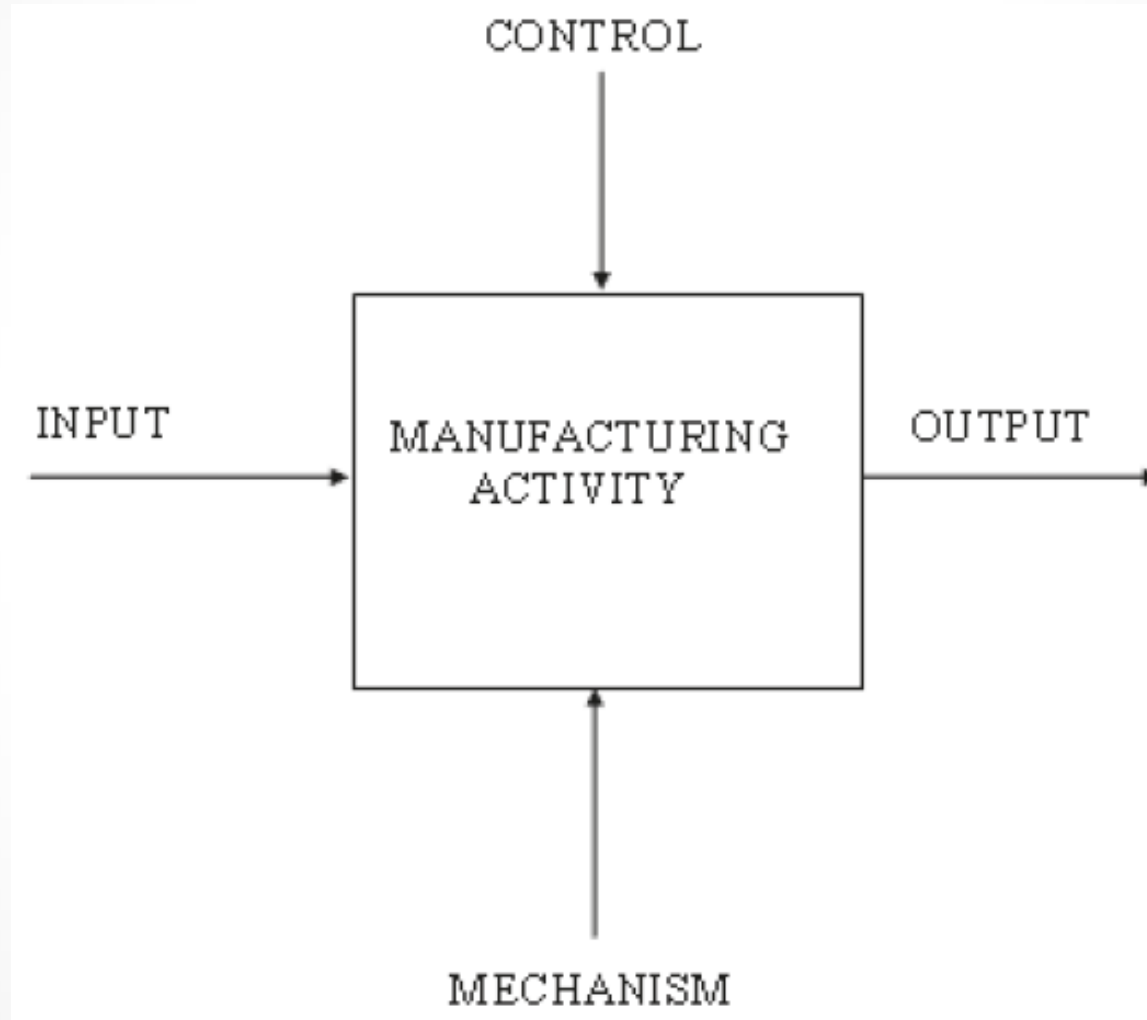


IDEF0

Methodology Modeling Primitives

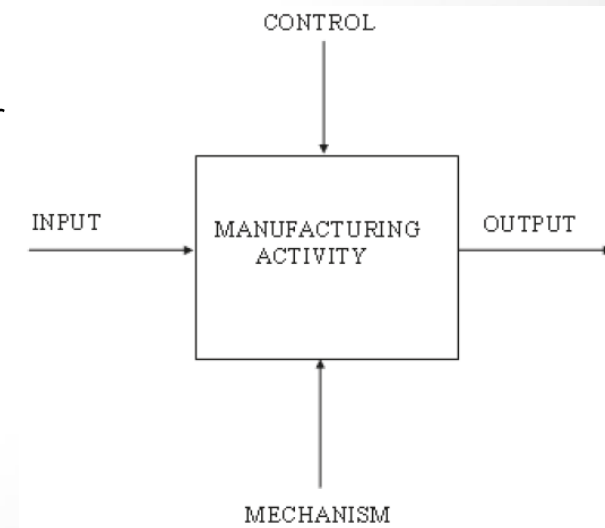
- IDEF0 (integrated computer-aided manufacturing definition 0):
 - modeling methodology
 - used for designing and documenting **hierarchic, layered, modular systems**
- The **activity box**:
 - used to describe a *function being performed* in the enterprise
- Function can be either a
 - *material conversion* function (machining a part), or
 - *information conversion* function (processing a requisition for ordering materials)

IDEF0 Activity Box and Connecting Arrows



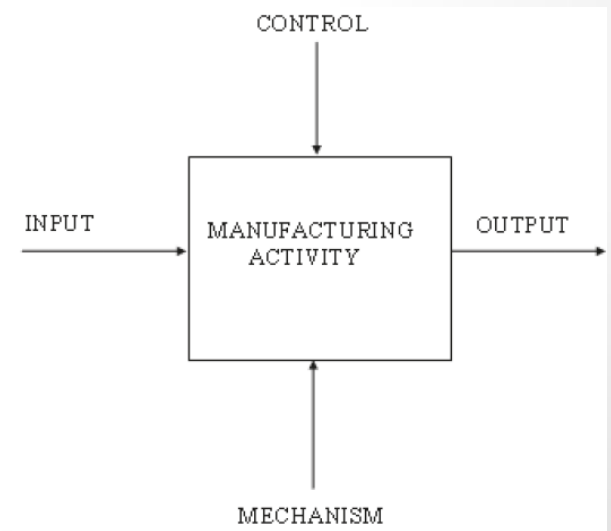
IDEF0 Activity

- **Inputs:** items that are transformed by the function
 - a workpiece to be machined, or
 - a requisition information to be transformed into a purchase order
- **Outputs:** result of the transformation process provided by the activity
 - finished component after machining, or
 - a purchase order



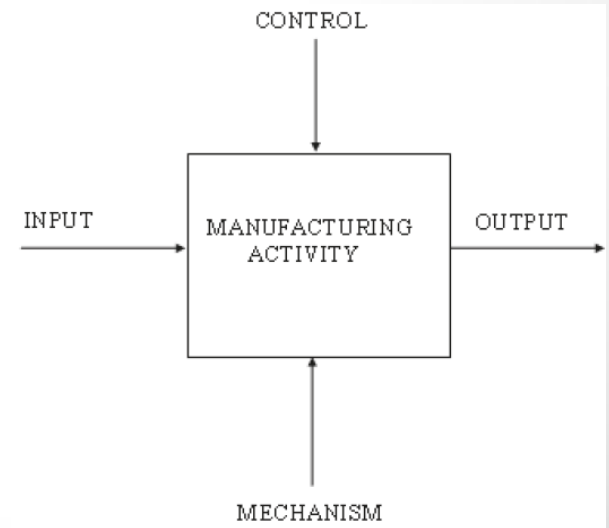
IDEF0 Activity

- **Mechanisms:** means by which a function is realized
- Mechanism in material conversion of a workpiece to a finished component might require (as mechanisms):
 - a *lathe* and
 - *lathe operator*
- Information conversion to process a requisition into a purchase order:
 - Could involve a *purchasing agent* as the mechanism



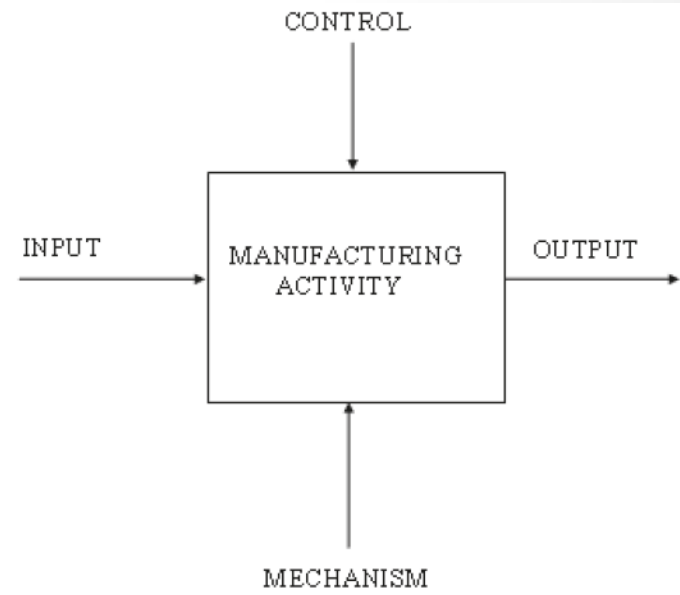
IDEF0 Activity

- A **control** is a condition or set of conditions that *guide* or *constrain* the performance of the activity
- *Machining activity* example:
 - parts may require a *numerical control parts program*
- *Information conversion* example:
 - requisition processing function may require adherence to *set of company rules or purchasing policy*
 - e.g. purchasing only from approved vendors



IDEF0 Activity

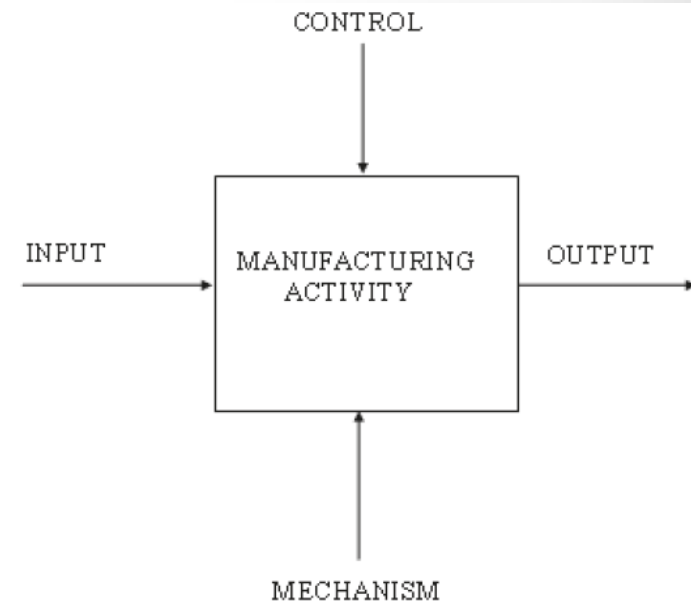
- **Activity box** and 4 **arcs** provide a concise expression:
 - an **input** is transformed ...
 - into an **output** ...
 - by an **activity (function)** ...
 - performed by a **mechanism** ...
 - and governed by a **control**



- The specific activity, inputs, outputs, mechanisms, and controls are defined by the *situation* being modeled

IDEF0 Activity

- **Grammatical convention** used in naming activities and arcs:
 - **Activities** represent *actions* being performed and are labeled with *verb phrases*
 - **Inputs, outputs, mechanisms, and controls** represent things and are labeled with *noun phrases*

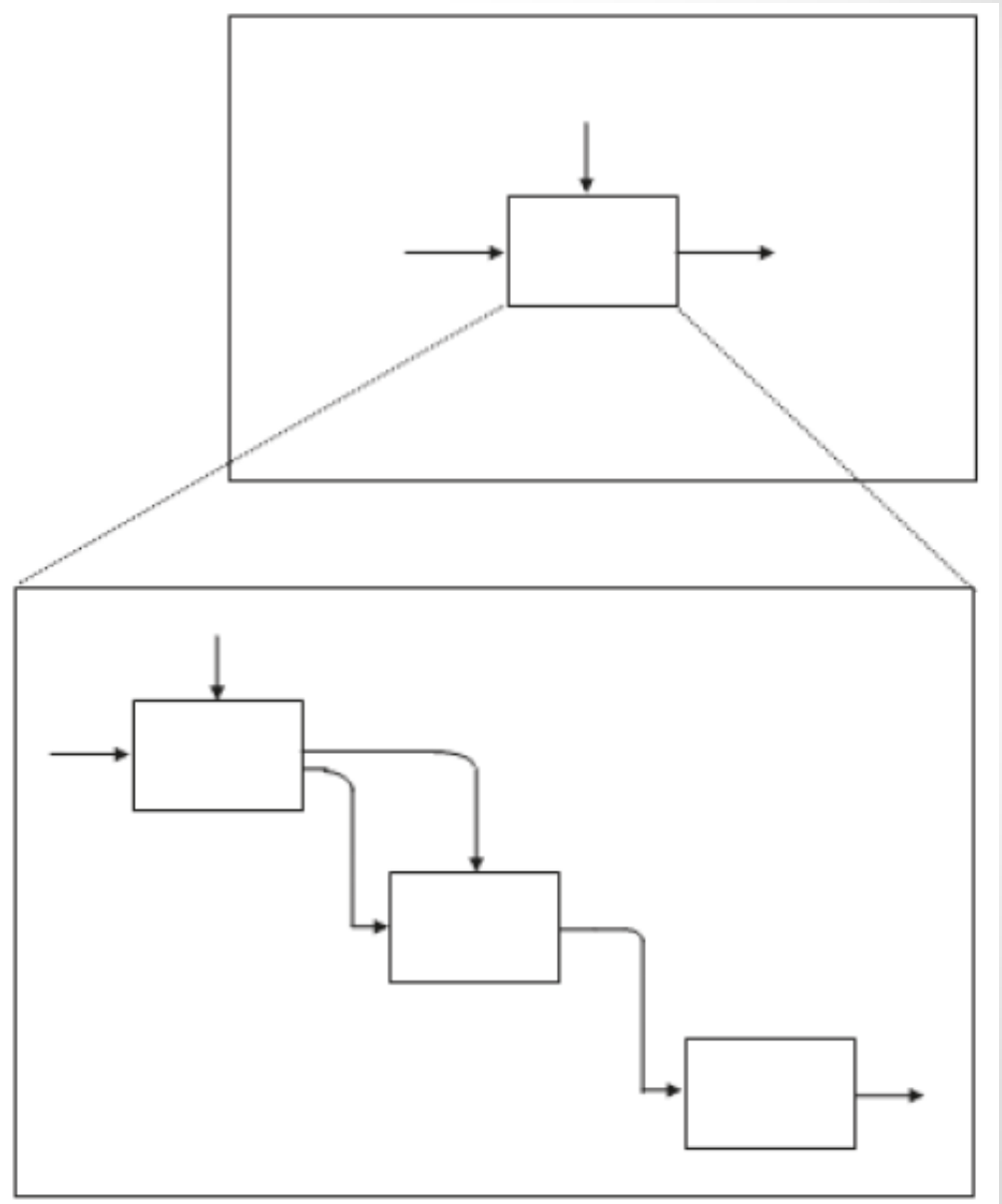


IDEF0 Hierarchic decomposition

- IDEF0 is a top-down modeling approach
- *First layer:*
 - a **single activity box**
 - describes the *overall function* of the enterprise, organization, or process within the enterprise (i.e. the subject of the model)
- This overall activity is then decomposed into its *major sub-activities* at the *second layer*
- Functions are related to each other by their **material flows** and **information flows**
 - e.g. the *output* material or information of one activity may provide the *input* to another activity

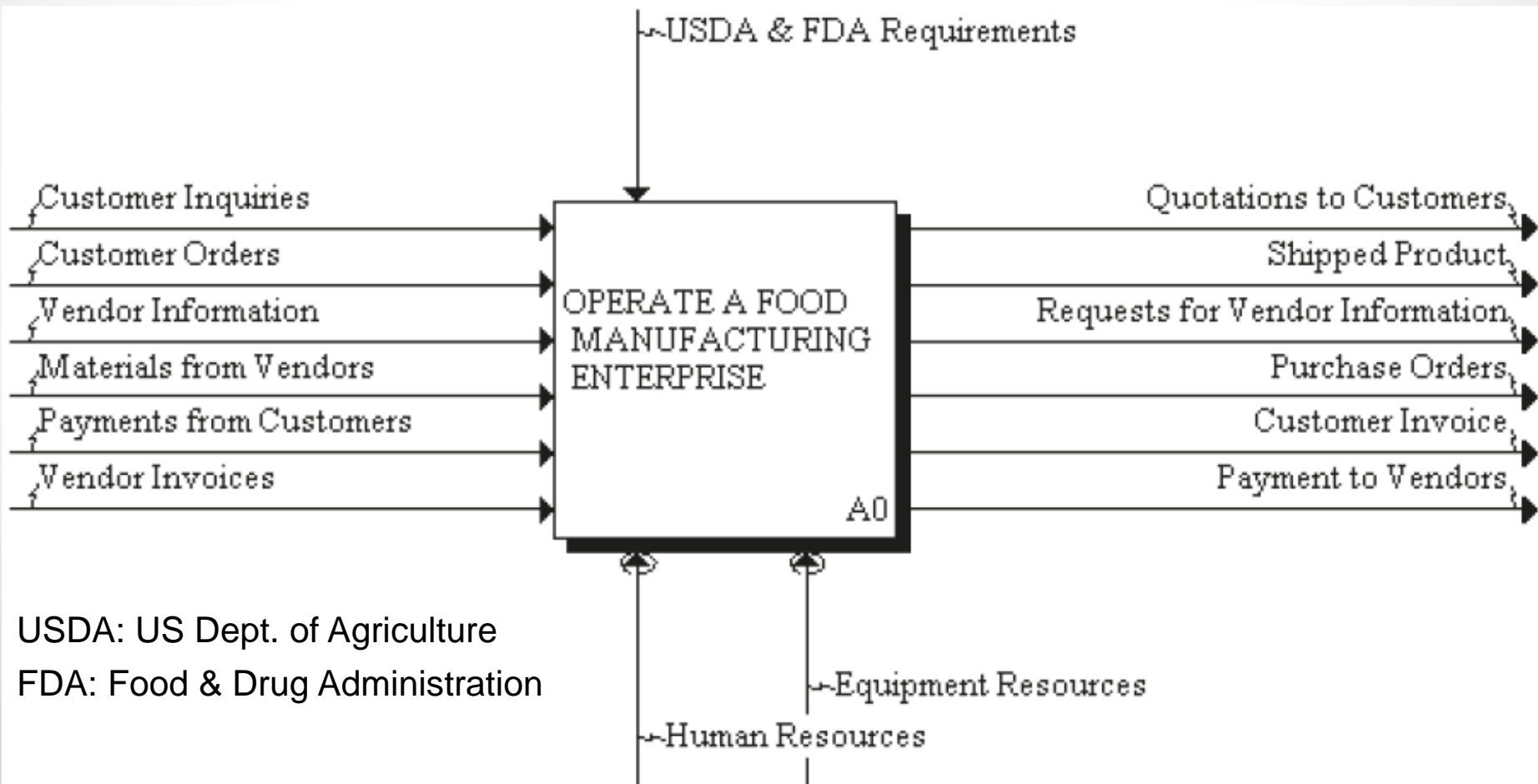
IDEF0 Activity

- Relationship among levels in IDEF0 methodology



An Integrated IDEF0 Model of an Entire Manufacturing Enterprise

- Top-level view of the enterprise: Node A0



Elements of the A0 Activity

- **A0**: high-level view of the enterprise and its interactions with the outside world
- Related Inputs (I) and Outputs (O):
 - Customer inquiries (I) \Rightarrow Quotations to customers (O)
 - Customer orders (I) \Rightarrow Shipped product (O)
 - Customer invoice (O) \Rightarrow Payments from customers (I)
 - Request for vendor information (O) \Rightarrow Vendor information (I)
 - Purchase orders (O) \Rightarrow Materials from vendors (I)
 - Vendor invoice (I) \Rightarrow Payments to vendors (O)

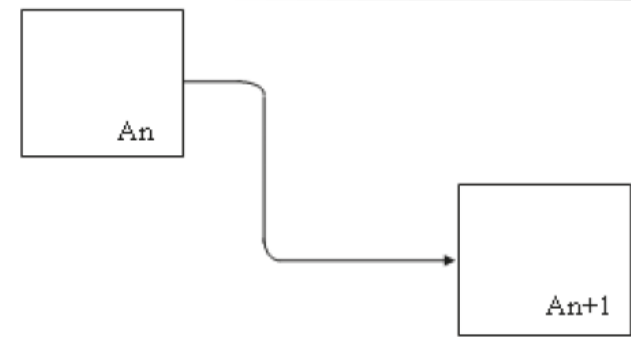
Elements of the A0 Activity

- In summary: operation of a manufacturing enterprise can be viewed as *3 interrelated processes*:
 1. **Physical flow of materials**; e.g. input materials from vendors is transformed (i.e. the manufacturing process) into the output shipped product
 2. **Information flow**; e.g. vendor information and quotation to customers
 3. **Financial flow**; e.g. payments from customers and payments to vendors
- Outside **controls**: USDA, FDA (important class of standards)
- **Mechanisms**: human resources and equipment
 - resources; these *convert* inputs to outputs

Decomposition of Node A0

- Decomposing node A0 identifies four major activities at the next level ([next slide](#)):

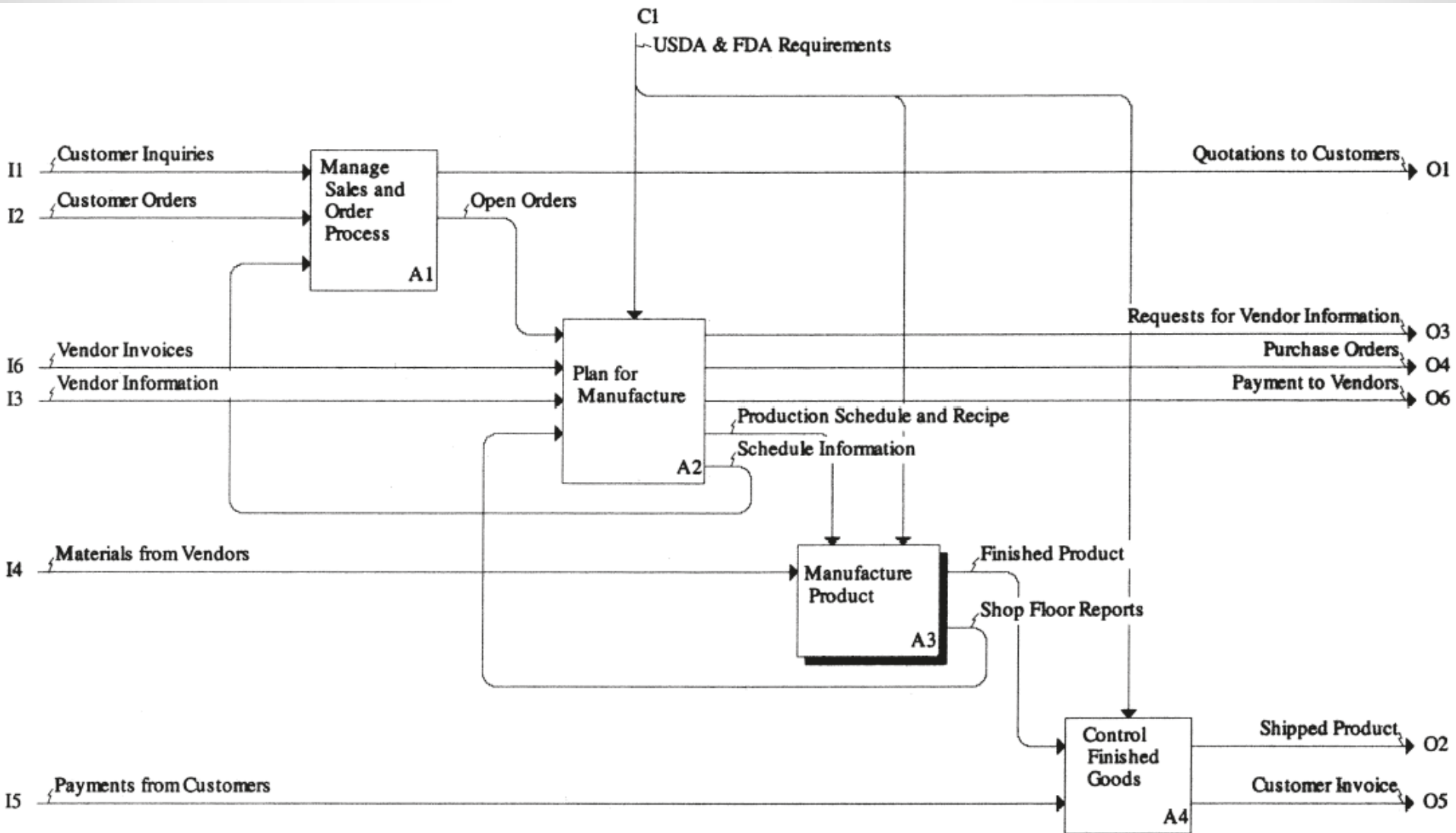
- o A1: Manage Sales and Orders Process,
- o A2: Plan for Manufacture,
- o A3: Manufacture Product, and
- o A4: Control Finished Goods



Simple output/input relationship

- Inputs and outputs of the *parent* activity are **inherited** by the *child*)
 - o e.g. [customer inquiries](#) and [customer orders](#) are handled by the sales organization and are inputs to [node A1](#)
 - o e.g. [finished product](#) leaves the enterprise from finished goods inventory (shown as an output of activity [A4](#))

Decomposition of Node A0



Decomposition of Node A0

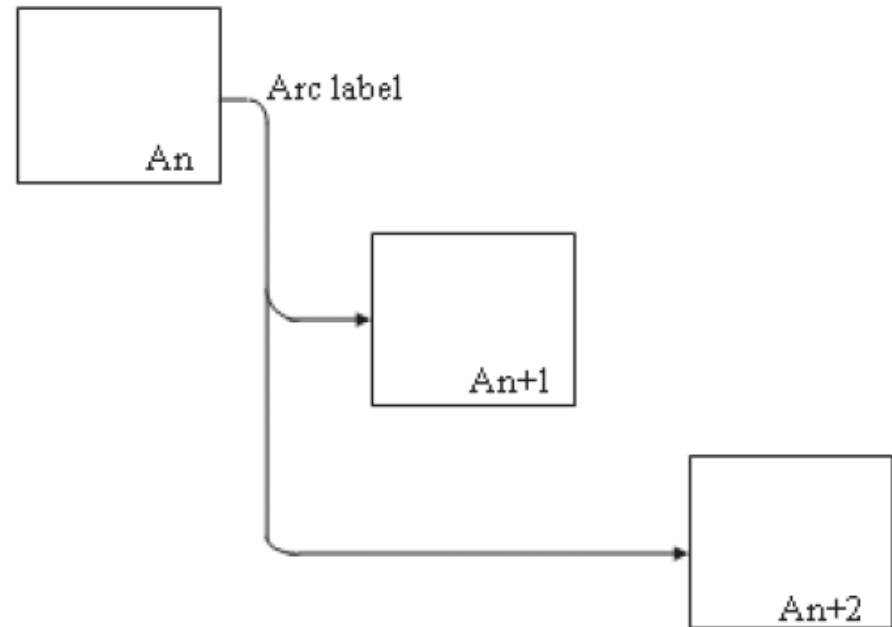
- Note the *activity flow relationship*:
 - When several activity boxes are drawn at the *same level* of decomposition, they are ordered by *sequential order/dominance*
 - e.g. “[Manage Sales and Order Processes](#)” activity precedes “Plan for Manufacture” activity, etc.
 - For ease of diagramming, IDEF0 methodology recommends using a *staircase pattern* whenever possible
 - Also recommended: at each level of decomposition, from *three to six child* activities be created from each *parent*

Decomposition of Node A0

- Decomposition of a parent activity is determined by the analyst in conjunction with enterprise personnel
- The *breakdown structure* of an activity into its main child activities is usually written in an *indented list*
- The following indented list applies at this point:
 - A0 — Operate a Food Manufacturing Enterprise
 - A1 — Manage Sales and Orders Process
 - A2 — Plan for Manufacture
 - A3 — Manufacture Product
 - A4 — Control Finished Goods

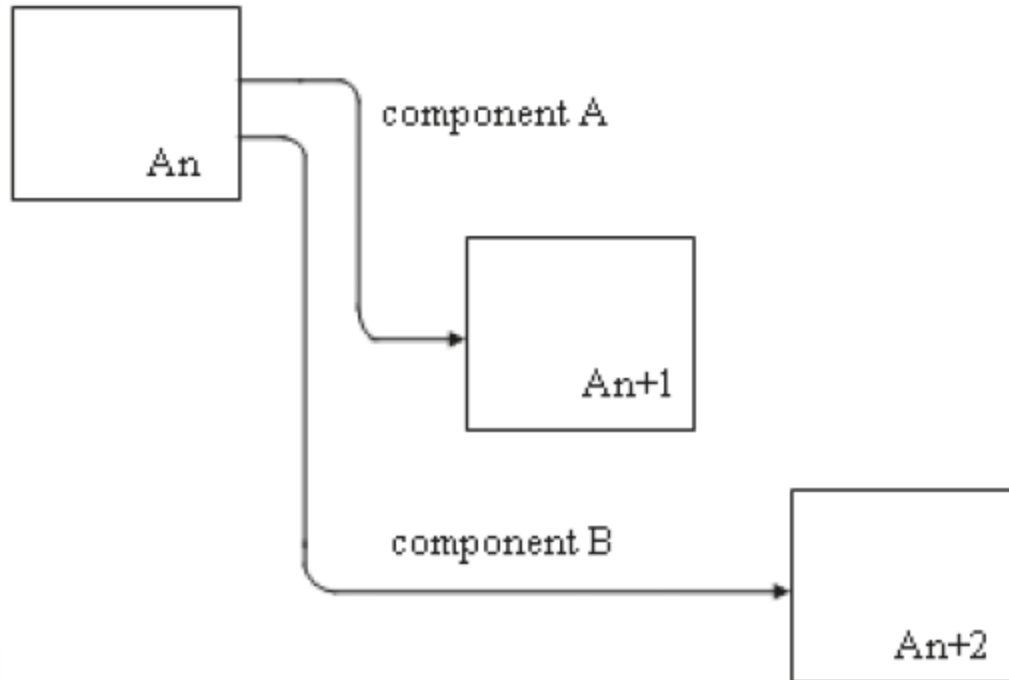
Connecting flows between activities

- **Parallelism:** simultaneous flow to more than one activity
- This is more common with a *flow of information* than with a *flow of physical entities*



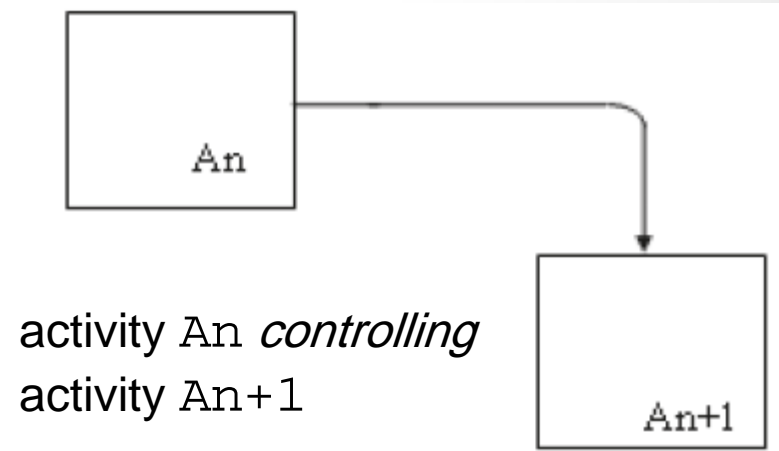
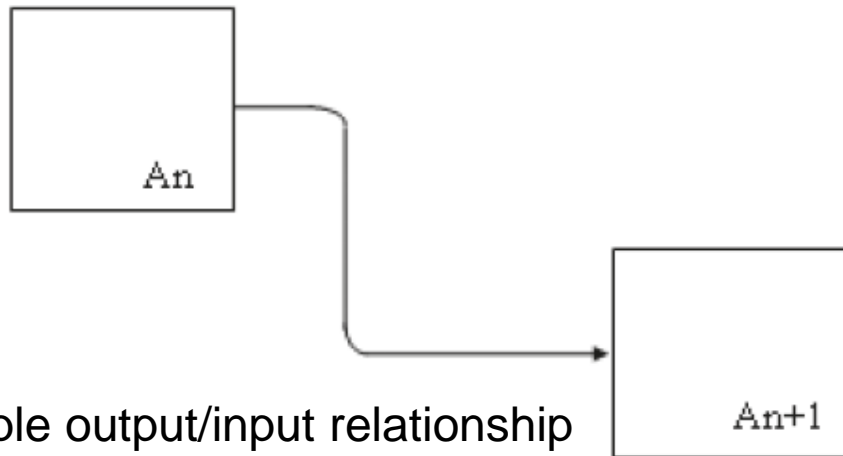
Connecting flows between activities

- **Distribution of flows** to more than one activity



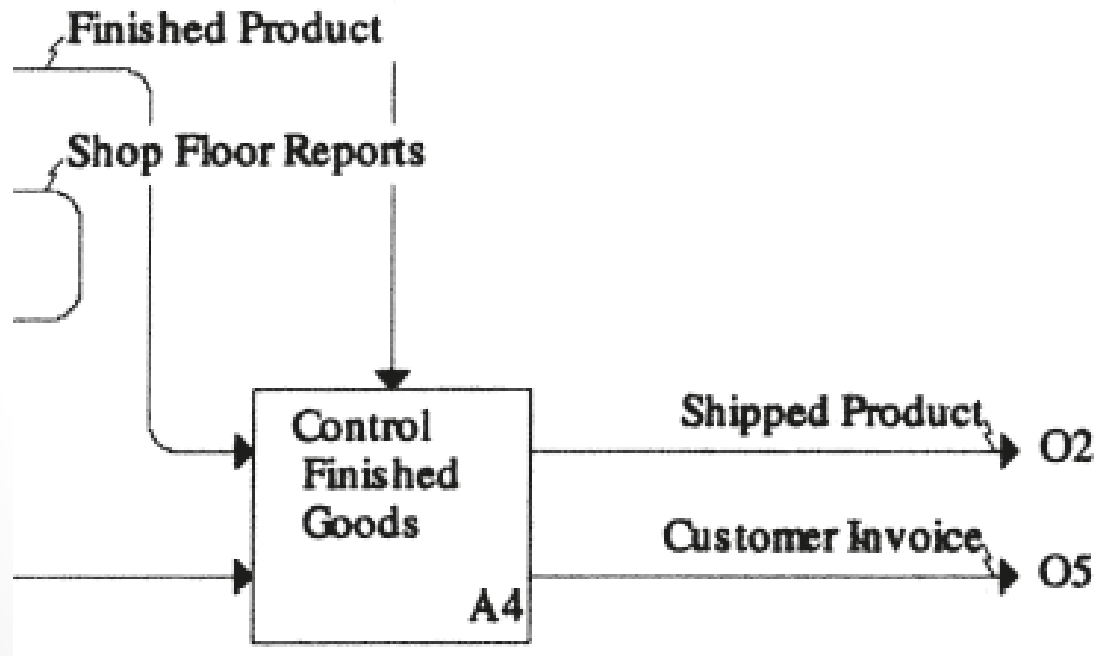
Connecting flows between activities

- One activity can provide inputs, controls, or both to other activities



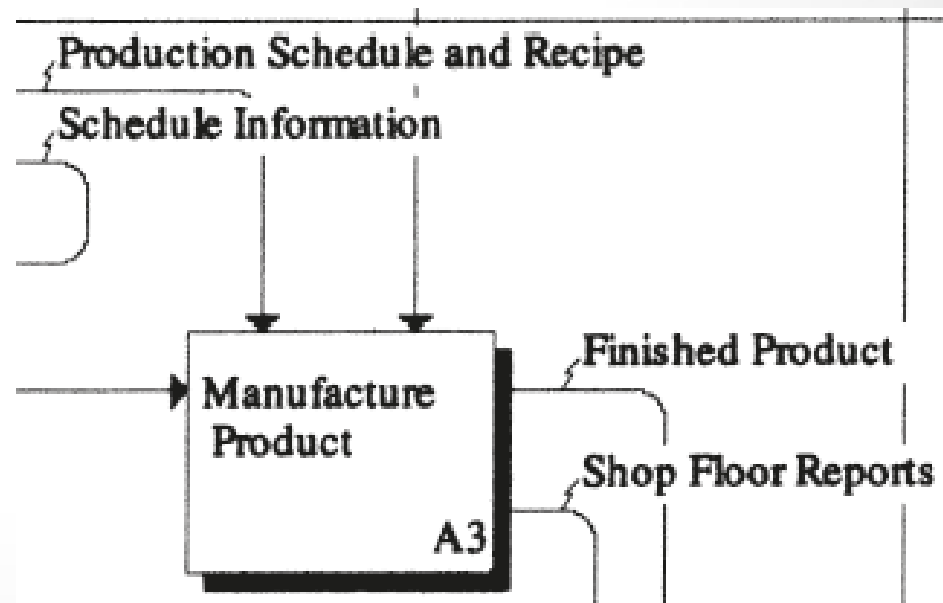
Connecting flows between activities

- Q: how to avoid confusion in deciding whether an arc is either an *input* to an activity or a *control*?
 - If entity represented by arc is **converted** into some other form by the activity, it is clearly an **input**
 - e.g. activity [A4](#) converts *finished product* into *shipped product*



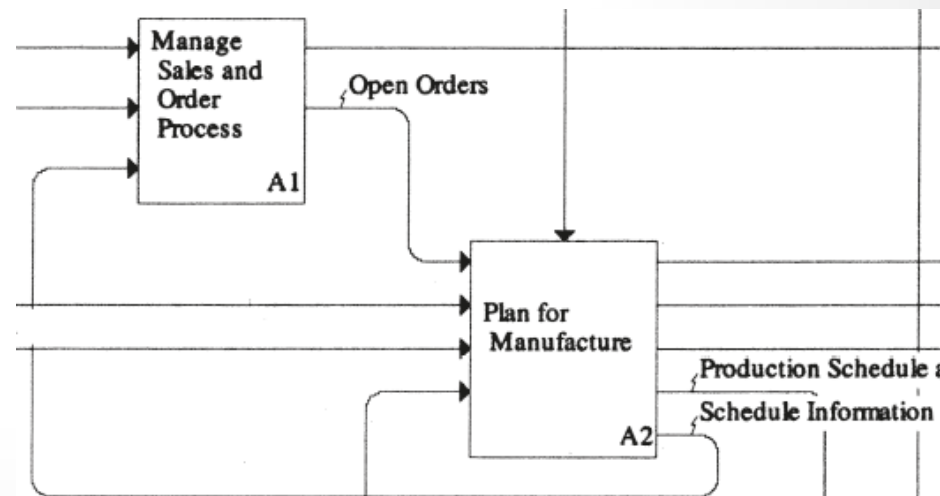
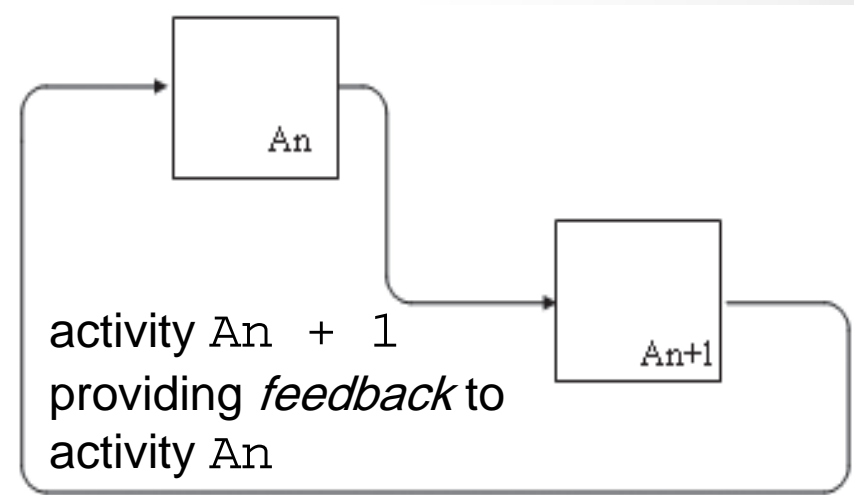
Connecting flows between activities

- Q: how to avoid confusion in deciding whether an arc is either an *input* to an activity or a *control*? (cont.)
 - If entity represented by arc **directs** the activity as to how it will perform its function, it is clearly a **control**
 - e.g. "*production schedule and recipe*" is information that *tells* activity [A3](#) what products will be produced on a specific day (production schedule) *and* how these products will be produced (recipe)



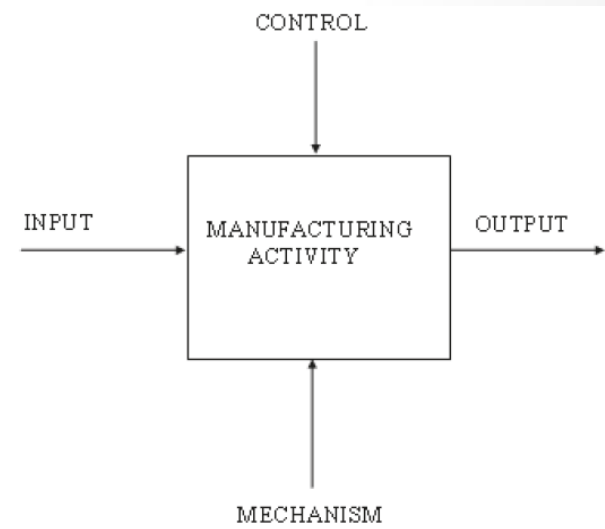
Connecting flows between activities

- **Feedback:** occurs when information generated in a subsequent activity is used by a prior activity in the activity diagram
 - e.g. relationship between [A1: Manage Sales and Order Process](#) and [A2: Plan for Manufacture](#) (i.e. to know delivery dates for customer quotation, you must know planned mfg. schedule)*



Connecting flows between activities

- Note, we have not shown *mechanisms* in the decomposition of node A0
- Mechanisms are only required at the **elemental level** of the modeling process (i.e. when a specific activity is identified at the *lowest level* of the hierarchy)



Sources

- [Design of Industrial Information Systems](#). Thomas Boucher, and Ali Yalcin. Academic Press. First Ed. 2006. [Chapter 4](#).
- Some useful videos:
 - Function modelling using IDEF0: The basics of functions, inputs, outputs, mechanisms and controls (<https://youtu.be/xyO5n6Ay-1I>)
 - AI0Win Tutorial – “Manage a Coffee Shop” (<https://youtu.be/kHDNIFclsiY>)

