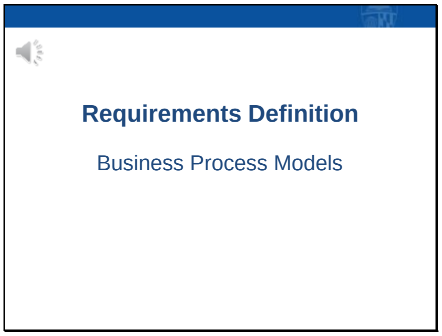
1

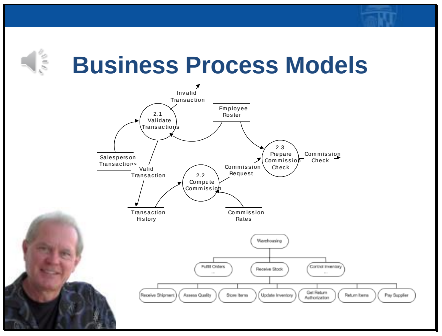


## Requirements Definition

Business Process Models

In this lecture we'll discuss how business process modeling fits into the requirements definition process.

2



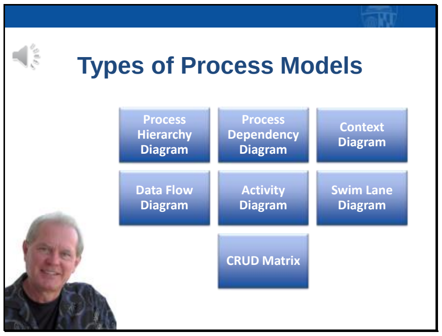
## Business Process Models

Business processes make up the heart of how an organization works. Business processes often evolve in an unstructured way and can become complex, hard to manage, and inefficient. When we develop software products to support an organization's processes, it's important to understand how those processes work. Development of new products often change an organization's processes and can also impact organizational entities in unexpected ways.

As part of the requirements definition activity in a software engineering project, it's often necessary to study existing business processes and evaluate the impact a new or modified software product will have on those processes. An excellent way to study these processes is by building process models.

Process models can help us discover how certain areas of a business currently work, how a proposed software product will impact existing operations, and can also help us to discover additional stakeholders, problem areas, and business rules.

3



## Types of Process Models

Process Hierarchy Diagram

Process Dependency Diagram

Context Diagram

Data Flow Diagram

Activity Diagram

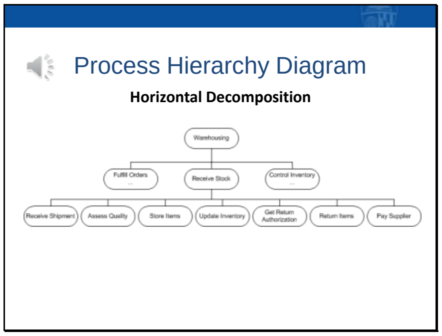
Swim Lane Diagram

CRUD Matrix

There are many types of process models used in practice today. I'm going to discuss a few of them…those that are listed here. My purpose in this lecture is to expose you to commonly used modeling tools…not to teach you the details of how to use these modeling tools…though most of them are pretty simple…so this is more of a "what" lecture than a "how to" lecture.

There are more business process models, and more modeling notations, than I will cover here.

4



## Process Hierarchy Diagram

**Horizontal Decomposition**

Warehousing

Fulfill Orders …

Receive Stock

Control Inventory …

Receive Shipment

Assess Quality

Store Items

Update Inventory

Get Return Authorization

Return Items

Pay Supplier

The first type of business process model that I'll talk about is the process hierarchy diagram. This is sometimes called a process decomposition diagram.

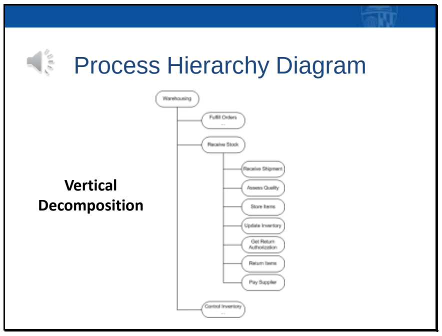
At the top of the hierarchy is one function of a business…in this case a warehousing function. A business function is defined as a group of business activities which together completely support one aspect of furthering the mission of an enterprise. The names of business functions are usually nouns…such as marketing, financial planning, or engineering. During the requirements phase of a project we would identify business functions that may be impacted by our project. Sometimes it is clear which business functions will be impacted…and sometimes it's not so clear…which is why it's important to model.

The hierarchy diagram starts with a business function and decomposes it into lower-level processes until the decomposition results in what's called elementary or primitive processes. An elementary process is the smallest unit of activity that would have meaning to an end user or stakeholder…basically it means that we can't break down a process any further.

In this diagram, I'm showing three processes associated with the warehousing function…fulfill orders, receive stock, and control inventory. There could be many other processes as well. The ellipsis…three dots…in the fulfill orders and control inventory portions of the diagram indicate that these could be further decomposed. Names of processes typically consist of an active verb and object…for example…allocate payment, calculate taxes, and so forth.

I’ve illustrated further decomposing the receive stock process into lower-level processes.

5



## Process Hierarchy Diagram

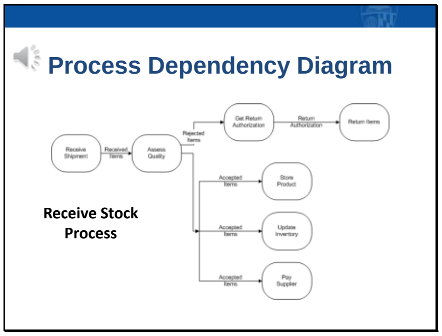
**Vertical Decomposition**

Here’s the same information, but in a slightly different visual presentation. Now, as you can see, the process hierarchy diagram shows the business processes associated with a business function. I’m only showing a few levels of decomposition here. In practice, there could be many levels. It depends on each individual function.

The purpose of a process hierarchy diagram is to visually communicate some or all of an organization’s functions. It helps analysts to identify and communicate the scope of a project, and it also helps analysts to identify additional project stakeholders.

What this diagram doesn’t show, however, are the dependencies between the various processes. For that, we need another diagram.

6



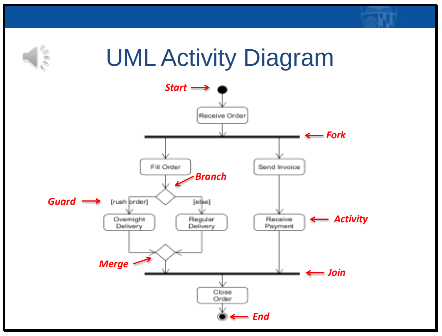
Process Dependency Diagram

The process dependency diagram illustrates the dependencies between various processes. Dependencies mean that certain processes must precede other processes, and some must be completed in a particular order.

In this process dependency diagram I'm illustrating the dependencies associated with the Receive Stock process. Arrows are used to depict the dependencies. The diagram is very simple to follow. The receive shipment process involves receiving a delivery of items from a supplier. The output of the process represents the items that are received. The items are then inspected for damage. Some items may be rejected, in which case a return authorization is requested and the items are subsequently returned to the supplier. The accepted items are stored in inventory, inventory control information is updated, and the supplier is paid.

The purpose of a process dependency diagram is to help understand which steps in a process are dependent on others. It can help to identify processes that may need to be changed or are otherwise impacted by a project. It can also be used to identify problems and bottlenecks.

7



UML Activity Diagram

Another common tool for modeling business processes is the UML Activity Diagram. This activity diagram shows the dependencies between activities in a business process.

The activities are represented by rectangles. Activities can represent an entire business process or a step in a business process. Activities should have arrows going in and out, unless the activity is a start or end activity. The arrows are called transitions.

At the top of the diagram, a special symbol is used to represent the start of a process. The first step in the process is to receive an order. The first horizontal bar represents what is called a fork. A fork indicates that several activities will occur in parallel.  Forks have one

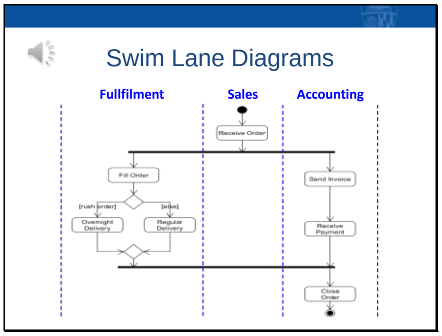
entry transition and multiple exit transitions.
A fill order activity takes place along the left-hand side. The order is sent via overnight delivery or regular delivery. At the same time, the customer is invoiced, and payment is ultimately received.

When all the parallel activities are completed, the order can be closed. The second horizontal bar represents a join. A join has multiple input transitions and one exit transition. All the activities above the join must be completed before the exit transition can take place.

A special symbol is used to designate the end of the process. A process will have one start node, but it may have several end nodes.

Let’s take a look at the left-hand activities. Branching occurs based on the type of shipping the customer has chosen for the order. The first diamond symbol is called a branch symbol. Nothing gets annotated inside the diamond. There is something called a guard…or guard condition…associated with the left-hand transition from the branch. Guard conditions are boolean expressions. If a guard condition is true, its transition will be taken. If guard conditions are used, they must form a complete set…meaning each outgoing transition must have a guard. It’s okay to use [else] or [otherwise] to indicate fall-through logic. Each branch must have a merge, which is also indicated with a diamond.

8

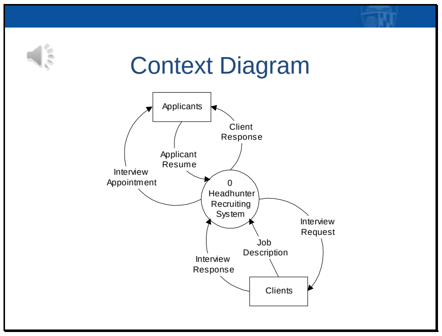


## Swim Lane Diagrams

A swim lane diagram is another common tool to model business processes. A swim lane diagram is used to illustrate which business area is responsible for the activities associated with a process.

The diagram illustrated here represents the same process as in the last example, but the "swim lanes" make it very clear which business area is responsible for performing each activity. In this example, I've used the activity diagram notation, but virtually any flowcharting notation could be used.

9

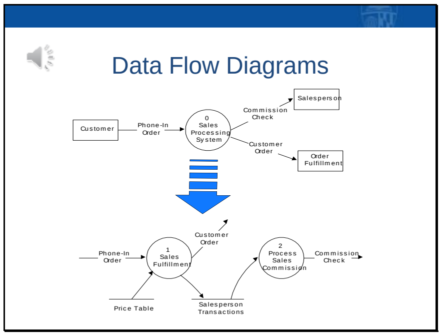


## Context Diagram

A context diagram is a tool for visually describing the scope of a system. The circle, or bubble, represents the entire system under study. The rectangles are called external entities. They can represent system users, other systems, and other organizations that are external to the system under study. The arrows indicate the key data that flows across the system boundary.

As I already mentioned, a context diagram provides a visual description of the system scope. It can also help to identify additional stakeholders.

10



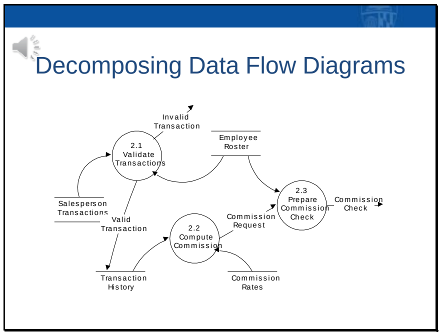
## Data Flow Diagrams

Yet another type of process model is a data flow diagram. It shows how information flows through a system and the process dependencies as well. Data flow diagrams originated as part of the structured systems analysis methodologies…and there are several notational styles.

Data flow diagrams usually start with a context diagram, and are then decomposed until they reach the level of elementary processes…so they can show process hierarchy as well as process dependency. There are also a set of rules associated with data flow diagrams that are used to ensure consistency and correctness.

In this example, I'm starting with a context diagram for a sales processing system. The entire system is shown as a single bubble. I then decomposed the context diagram into a first-level data flow diagram. Note that there are two processes shown at this level…a sales fulfillment process and a sales commission process. Note also that the three data flows from the context diagrams…phone in order, customer order, and commission check…appear on the first-level diagram. This is one of the rules…and it's called data flow "balancing" or "levelling". Net data flows for a child data flow diagram and its parents must be the same, except for file accesses and trivial error handling.

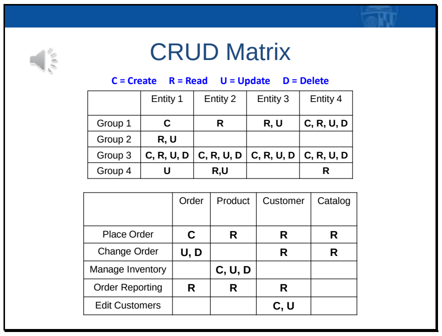
11



## Decomposing Data Flow Diagrams

In this example, I further decomposed the sales commission process into sub-processes. Note that the net input and output data flows are the same as in the parent process. The additional data flows are inter-process data flows for the processes shown at this level, file accesses, and an error-handling flow.

12



## CRUD Matrix

C = Create R = Read U = Update D = Delete

| | Entity 1 | Entity 2 | Entity 3 | Entity 4 |
|---|---|---|---|---|
| Group 1 | **C** | **R** | **R, U** | **C, R, U, D** |
| Group 2 | **R, U** | | | |
| Group 3 | **C, R, U, D** | **C, R, U, D** | **C, R, U, D** | **C, R, U, D** |
| Group 4 | **U** | **R,U** | | **R** |

| | Order | Product | Customer | Catalog |
|---|---|---|---|---|
| Place Order | **C** | **R** | **R** | **R** |
| Change Order | **U, D** | | **R** | **R** |
| Manage Inventory | | **C, U, D** | | |
| Order Reporting | **R** | **R** | **R** | |
| Edit Customers | | | **C, U** | |

A type of diagram that is often used in conjunction with both process modeling and data modeling is the CRUD matrix. CRUD stands for create, read, update, and delete…and it can be used to map data permissions between data entities and other things.

In this example, I'm indicating the types of access permissions different stakeholder groups have to four different entity types.

In this example, I'm showing how entity types are accessed with respect to five different use cases. This will be very helpful to the system designers, and can also be used to help verify correctness and completeness of the use cases. For example, note that there is no customer delete capability in any of the use cases indicated in the matrix. This could indicate several things: one…a delete capability is not required, two…we've missed a requirement, or three…the edit customers use case may not be complete.