



SYST 101: Intro to Systems

Lecture 7

Feb. 11, 2003

C. Wells, SEOR Dept.



Announcements

- Lab hours beginning Thursday
 - Class period
 - Hour after class period
 - We will load the computer for Mindstorm programming
 - But bring your CD, cable, and infrared transmitter



Agenda

- Project 1 status
- Issue formulation
- Flowcharts



Project Teams

- Who is not on a team now?
- Team member check-in
- Thoughts on peer evaluations



Old Project 1 Test Schedule

Feb 25	
12:15-12:30	Team 4
12:30-12:45	Team 2
12:45-1:00	Team 3
1:00-1:15	Team 8

Feb 27	
12:00-12:15	Team 1
12:15-12:30	Team 5
12:30-12:45	Team 7
12:45-1:00	Team 6
1:00-1:15	Team 9



New Project 1 Test Schedule

Feb 25	
Team 4	Test 12:15 Present 12:35
Team 2	Test 12:20 Present 12:45
Team 3	Test 12:25 Present 12:55
Team 8	Test 12:30 Present 1:05

Feb 27	
Team 1	Test 12:00 Present 12:25
Team 5	Test 12:05 Present 12:35
Team 7	Test 12:10 Present 12:45
Team 6	Test 12:15 Present 12:55
Team 9	Test 12:20 Present 1:05



Issue Formulation

- How to determine what's important and what's not?
 - How do you know when you've captured all the issues?



Defining Desired Behavior

- One starting point: Function flow diagrams
 - Flowcharts
 - Decision processes
- Structured analysis and object-oriented techniques addressed in SYST 301 & 520.

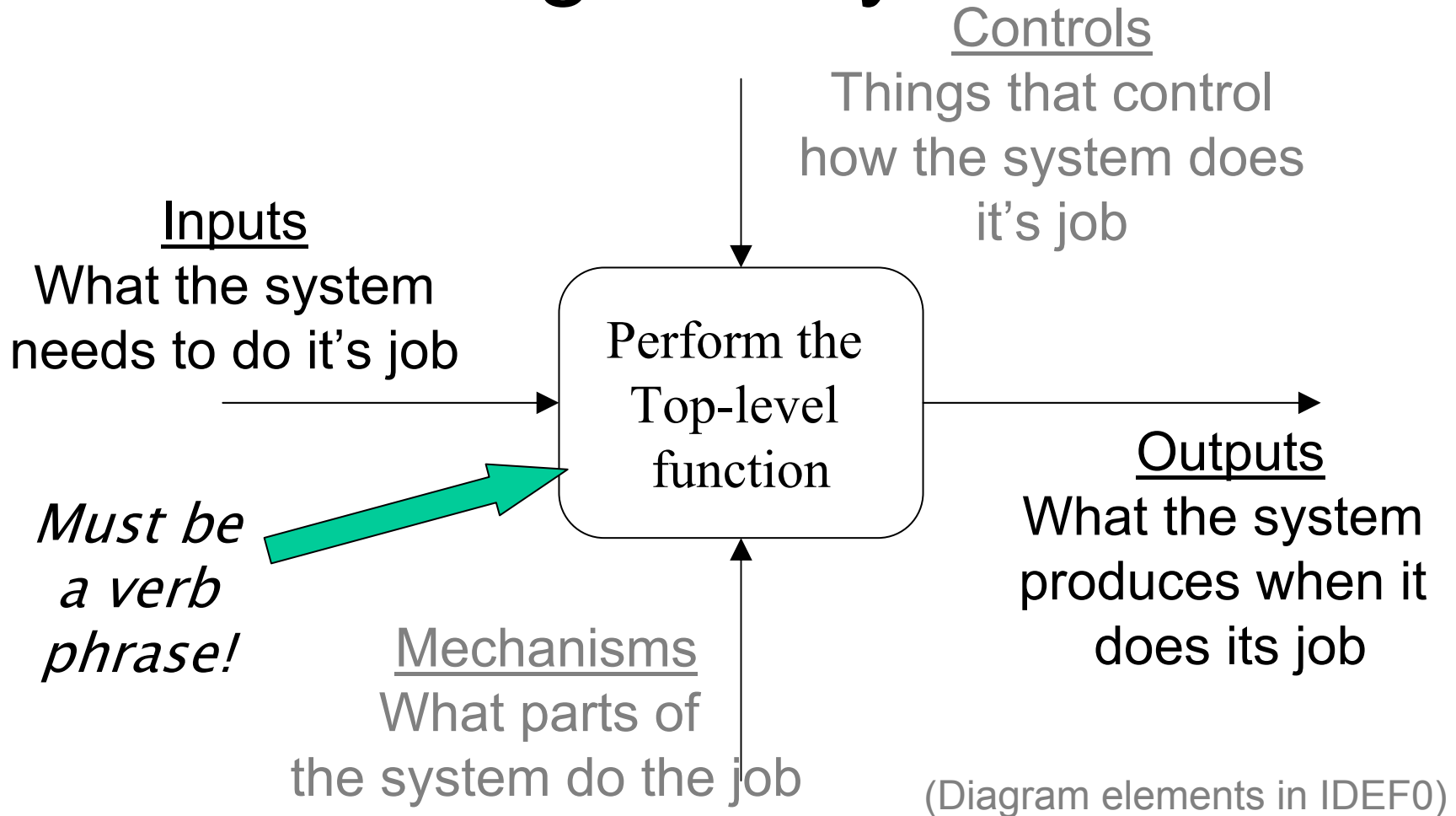


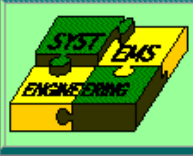
Process Description

- Just knowing the functions is not sufficient
 - What's the order? What makes me decide to this OR that? When can I start doing a function?
- Dynamic descriptions are also important.



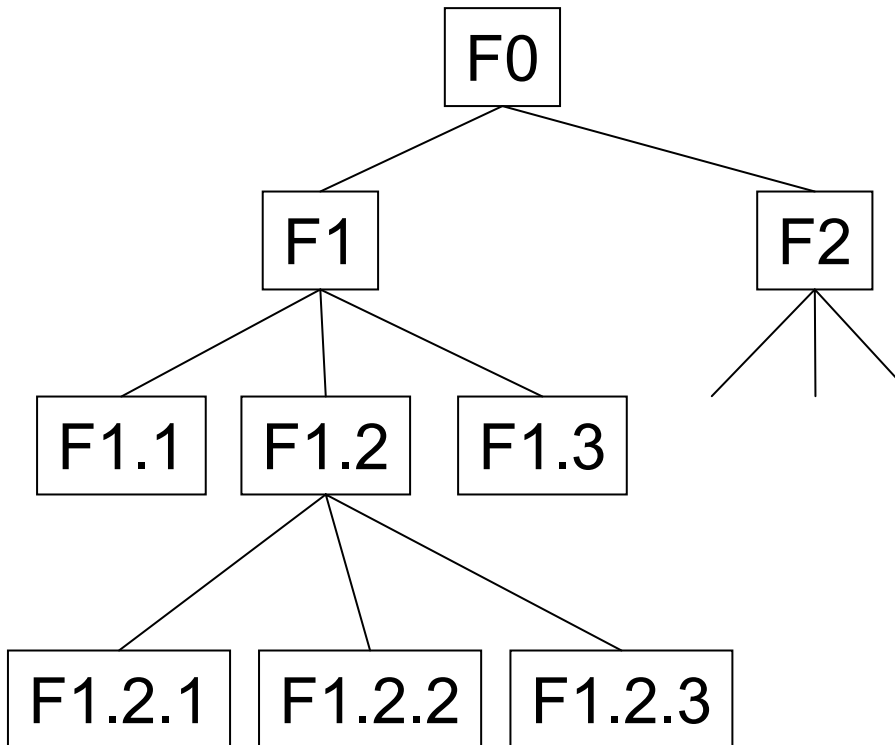
Diagram Syntax





Functional Decomposition – Representation Techniques

Graphical Representation



Outline Form

- F0
 - F1
 - F1.1
 - F1.2
 - F1.2.1
 - F1.2.2
 - F1.2.3
 - F1.3
 - F2



Deciding on the Subfunctions

- Does involve some degree of experience, practice, creativity
 - There are no instructions on how to come up with the “right” subfunctions
 - “Bad” or poor decompositions will have certain “symptoms” – return to this later
- Can be iterative with the following analysis steps (dataflows, entity-relationships)



Rules on Functional Decomposition

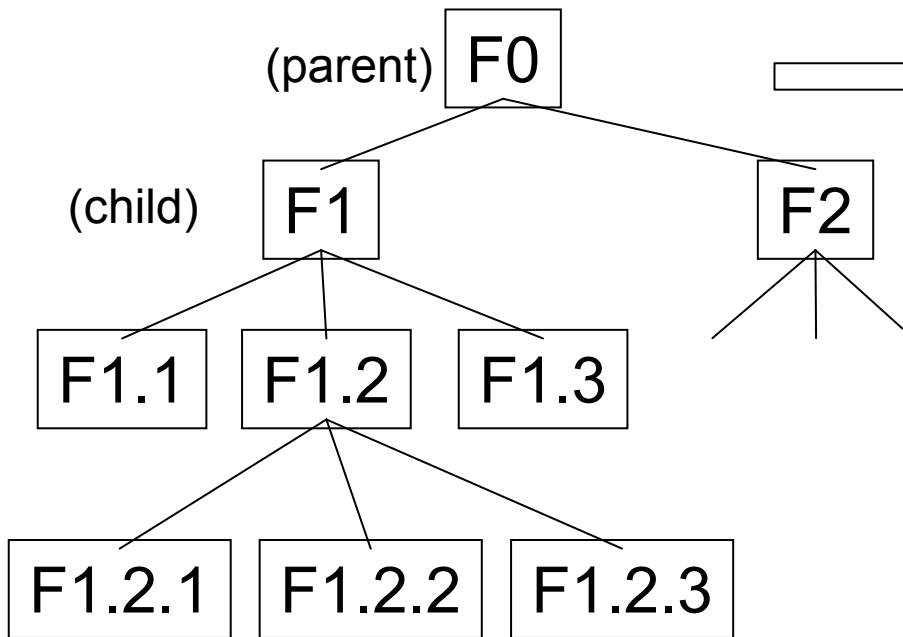
- No function can be repeated.
- Function names must be verb phrases – connecting flows must be entities (data, things)
- Functions should have clear boundaries between them
 - This will be reinforced when you define the input-output entities.
- How many levels of decomposition are necessary?
 - No single answer – depends on the scope and intended audience of your design project
- How can one tell what's a “good” decomposition?
 - Clear subfunctions, easy connections between them, don't violate the above rules



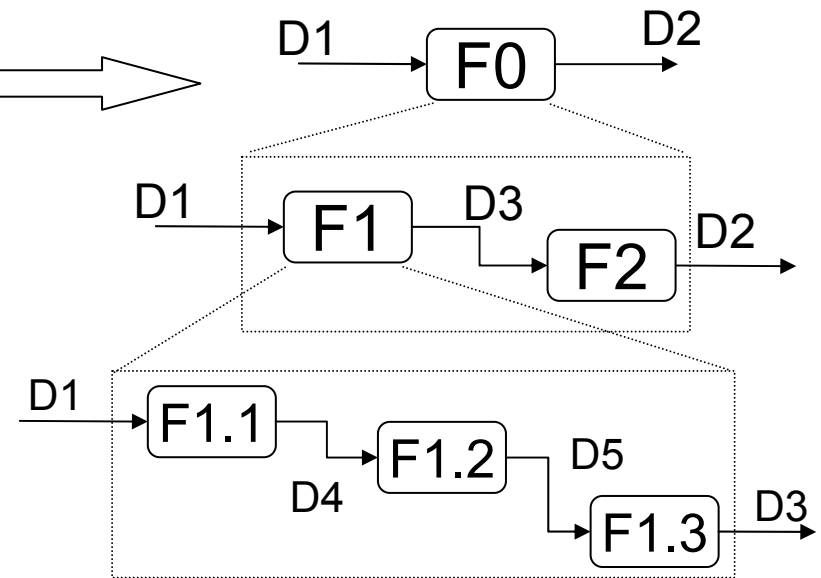
Functional Decompositions

- Every function that is decomposed gets it's own diagram

Hierarchical Tree



Hierarchy of diagram pages



D = "data"
Slide 14



FFBDs

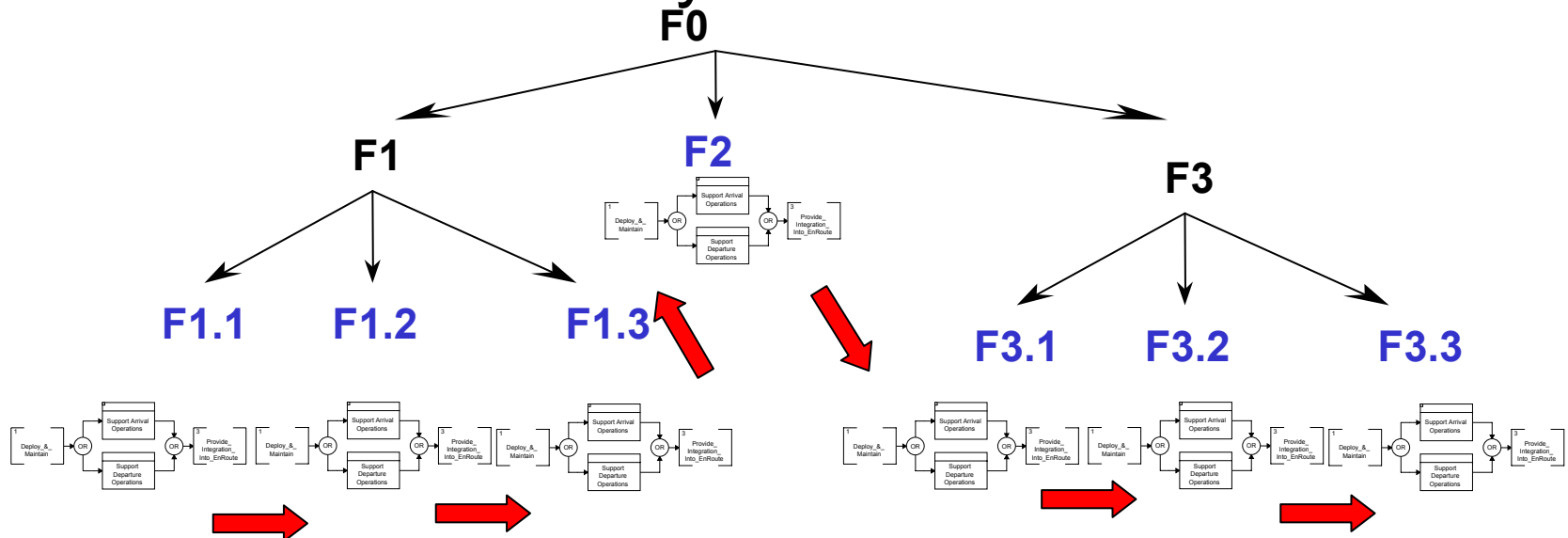
(Functional Flow Block Diagrams)

- FFBDs are one method of describing system behavior
- Behavior modeling is a key component of a complete system description
 - What the system does -> Functional View
 - What the system is made up of -> Entities & Mechanisms
 - How the system behaves -> Behavioral View
- Goal: Achieve an Understanding of the Diagram Syntax and Construction Techniques



FFBDs Form a Coherent Chain

- FFBDs are built for each **leaf node** in the functional hierarchy



★ Leaf node: Nodes in the functional hierarchy which do not have any further decomposition.



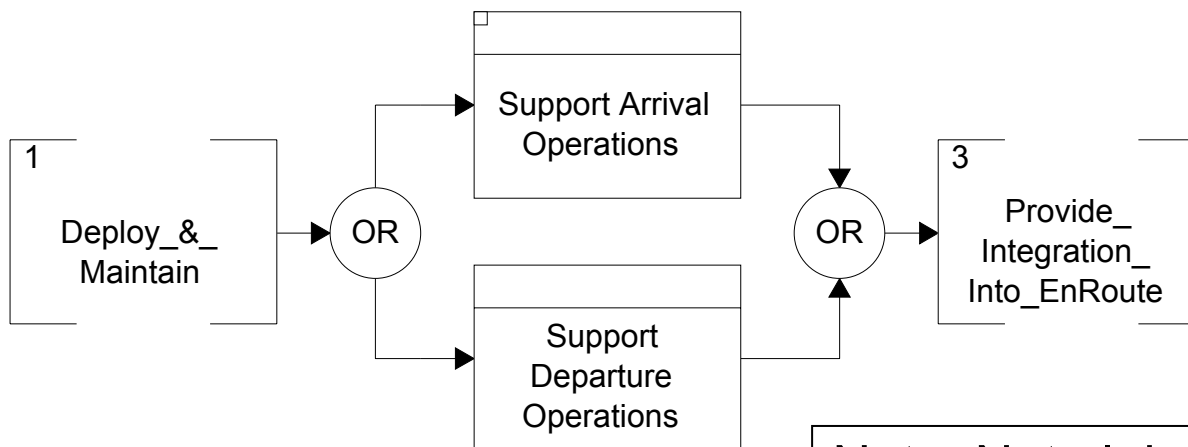
FFBD Constructs

- Basic - Input and Output boxes at each end correspond to preceding and following activities
- Logic connecting nodes control the action
 - Functions need not be sequential



FFBD Diagram Constructs

- **Desired Behavior:** In this case, we want each activity to be able to be performed concurrently and asynchronously of the other activity. Therefore, use a parallel structure.

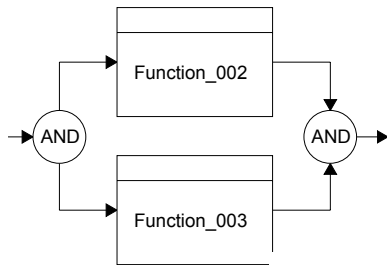


Note: Note labels on the input and output ref. boxes.

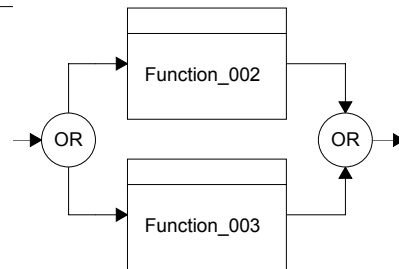


Other FFBD Diagram Constructs

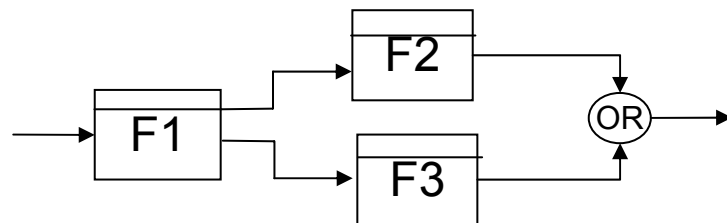
- Additional Constructs Available



Both functions can run concurrently, and both must finish before process continues



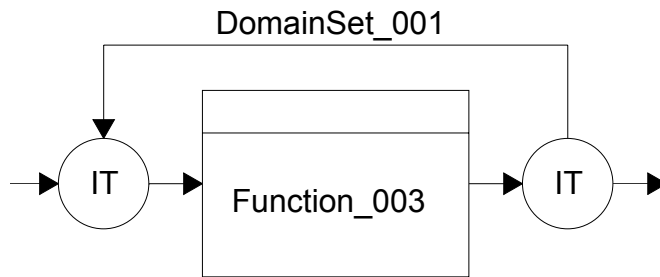
Both functions can run concurrently, but process can continue when either one finishes



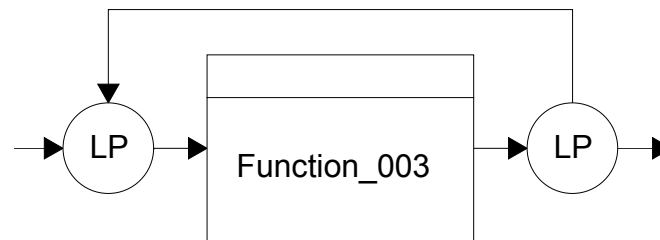
“If x then do F2, when y do F3”. Need to define 2 or more Completion_Criteria for F1



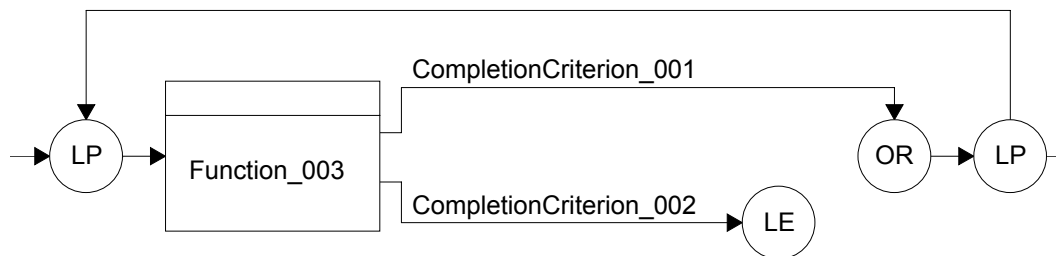
FFBD Diagram Constructs 2



Iterate: Function_003 performed for every element in some DomainSet_001.



Loop: Function_003 performed until some condition is satisfied.



Loop with Exit: Function_003 performed until some condition is satisfied. Loop Exit (LE) leaves the diagram.



Implications of FFBDs

- Does not address people issues
 - FFBD are like plans: worthless if nobody executes them
- Provides ordering not a schedule
- Only as good as the thought that goes into it
 - Does not ensure “goodness”



Assignments

- Reading
 - Petroski, Chapter 4, “Zippers and Development”
- Homework
 - Work on Project 1