Describing Current State and Specifications

11 November 2009
Lecture 4

Topics for Today

- Return to the roadmap
- Describing Existing State and Needs
  - Documentation and its parts
  - Documentation techniques
- Initial Design for Information Systems
  - Initial Design Documentation
  - Defining the Goals of a New Information System and Organizational Constraints
  - Business Processes and Computerized Processes
  - Describing a Computerized Process
  - Systemic Qualities and Size Approximation
  - Using a Prototype for the Initial Design
- Source: PS98 2.4-3.1

Where are we?

- Development Guides
  - Peretz Shoval (ADISSA)
  - MethodA (מפת"ח)
- Steps in Software Development Life Cycle (SDLC)
  - The development guides give direction for the planning, evaluation, and preparation for the system
  - The SDLC give guidance for the development and testing of the software
- We could use any number of SDLC with ADISSA or MethodA

Step	Stage	Sub-Stage	Whom
1	Initialization		Customer
2	Preliminary specification	Existing State Preliminary Specification	Customer
3	Feasibility Study	Feasibility Study Request for Proposals Receiving Proposals from suppliers Decision about supplier Contract and development plan	Customer
4	System Analysis	Data Dictionaries Dataflow Diagrams	Developer
5	System Design	Database planning User Interface design Input/Output specifications	Develop
6	Construction	Coding Code Documentation Acceptance tests Quality Assurance Testing documents	Developer
7	Deployment	User training Support training Data import from old system Pilot Turning it on	Developer and Customer
8	Fixing and Support		Customer/Developer

So far

- Return to the roadmap
- Describing Existing State and Needs
  - Documentation and its parts
  - Documentation techniques
- Initial Characterization for Information Systems
  - Initial Characterization Documentation
  - Defining the Goals of a New Information System and Organizational Constraints
  - Business Processes and Computerized Processes
  - Describing a Computerized Process
  - Systemic Qualities and Size Approximation
  - Using a Prototype for the Initial Characterization

Documenting Existing State

- Based on what we gathered, create summary documentation
  - Will be iterative, made by committees
  - Will be a document describing the output of the חקר מצב הקיים
  - Will be used to help in the אפיון ראשוני
- Major parts of the documentation should be:
  - Executive Summary
  - Organizational description
  - Description of current work processes (short)
  - Description of the current system (hand/computer)
  - List of problems (organized by previous part)
  - List of new requirements and information requested
  - Appendices with detailed information
    - Inputs, outputs, sample forms/documents, questionnaires, interview summaries
- Should be specific, but not too technical
  - Level of detail will vary
  - Use standard documentation techniques

Documentation Techniques

- Natural Language
- Structured Language
- Program Flowchart
- System Flowchart
- HIPO
- Process Chart
- Data Flow Diagrams (DFD)

- שפה טבעית
- שפה מובנית
- תרשים זרימה תכנית
- תרשים זרימה מערכת
- תרשימיHIPO
- תרשים תהליך
- תרשים זרימת מידע

Natural Language

- Just write down things freeform
  - Could be in speaking voice or formal writing
- Advantages:
  - Easiest to write
  - Good for questionnaires, interviews
  - Good for describing current state informally
- Disadvantages:
  - Much longer to read
  - Sometimes hard to understand (if badly written)
  - Not always clear (too easy to be vague)

Structured Language

- Use reserved words for describing processes
  - Can be Hebrew or English
- Like C or Java:
  - Order
  - Conditional
  - Loop
- Order:
  - Ordered steps
- Conditional:
  - If <condition> Then <action> Else <action>
- Switch Case:
  - Do Case:
    - Case 1: If <condition> Then <action>
    - Case 2: If <condition> Then <action>
    - …
    - Otherwise: <action>
  - End Case.
- Loop:
  - Do While <condition> <action>, <action>… End While
  - Repeat <action>, <action>… Until <condition>

Structured Language: Conditional

- Natural Language: Read a list of names and grades. If the grade is greater than or equal to 95, write “Excellent.” If the grade is grade is greater or equal to 85, write “Good.” If the grade is less than 56, write “Fail.” Otherwise write nothing.

```
Begin
Display "Enter the name of the student, course code, and grade"
Read the data into variables s_name, course_code, grade
If grade >= 95 Then print s_name, course_code, grade, "Excellent"
   Else if grade >= 85 Then print s_name, course_code, grade, "Good"
        Else if grade <= 56 Then print s_name, course_code, grade, "Failed"
             Else [Note: For all other grades, nothing is printed]
End
```


Structured Language: Do Case

- Natural Language: Read a list of names and grades. If the grade is greater than or equal to 95, write “Excellent.” If the grade is grade is greater or equal to 85, write “Good.” If the grade is less than 56, write “Fail.” Otherwise write nothing.

```
Begin
Display "Enter the name of the student, course code, and grade"
Read the data into variables s_name, course_code, grade
For cases:
   Case 1: If grade >= 95 Then print s_name, course_code, grade, "Excellent"
   Case 2: If grade >= 85 The print s_name, course_code, grade, "Good"
   Case 3: If grade <= 56 Then print s_name, course_code, grade, "Failed"
   Otherwise: [Note: In this example it's also possible to skip the Otherwise
   clause, since there is nothing to do]
End Case
End
```


Structured Language: Loops

Natural Language: Extending the previous examples, consider that we wish to allow multiple student data entries and to print out at the end of the procedure the number of students read and their average grade.

```
Begin
n_students = 0, totalgrade = 0
more = "y" [This variable will check the loop condition]
While more = "y" Do:
      Display "Enter the name of the student, course code, and grade"
      Read the data into variables s_name, course_code, grade
      For cases:
            Case 1: If grade >= 95 Then print s_name, course_code, grade, "Excellent"
            Case 2: If grade >= 85 The print s_name, course_code, grade, "Good"
            Case 3: If grade <= 56 Then print s_name, course_code, grade, "Failed"
            Otherwise:
      End Case
      Increment n_students
      Add grade to totalgrade
      Print "Are there more grades? Press y or n"
      Read variable more
End While
Print "The number of students = ", n_students
Print "Average Grade = ", totalgrade/n_students
End
```

Structured Language: Procedures

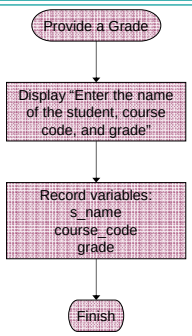
- If you have complicated procedures, name them
  - "Call" them by writing:
    - Do `procedure name'
    - Begin 'function name'

Program Flow Chart

- Graphical representation of structured language
  - Can often be automatically generated
  - Can be clearer than structured language, but don't add information
- Rules:
  - For each step, draw a box
  - For each branch, draw arrows to show the program flow
  - Procedures are drawn in separate flow charts

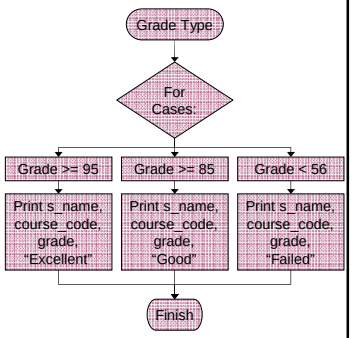
Program Flow Chart: Examples

Begin "Provide a Grade"
Display "Enter the name of the student, course code, and grade"
Read the data into variables s_name, course_code, grade
End



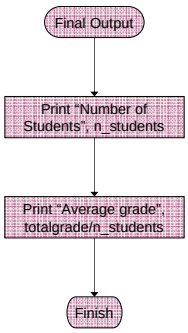
Program Flow Chart: Examples

Begin "Grade Type"
For cases:
Case 1: If grade >= 95 Then print s_name, course_code, grade, "Excellent"
Case 2: If grade >= 85 Then print s_name, course_code, grade, "Good"
Case 3: If grade <= 56 Then print s_name, course_code, grade, "Failed"
Otherwise:
End Case
End



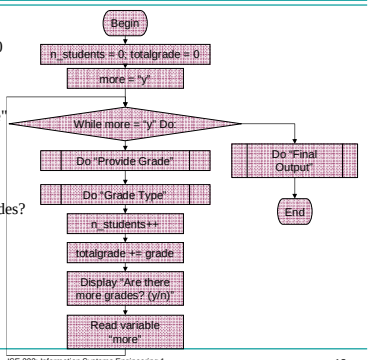
Program Flow Chart: Examples

Begin "Final Output"
Print "The number of students = ", n_students
Print "Average Grade = ", totalgrade/n_students
End



Program Flow Chart: Examples

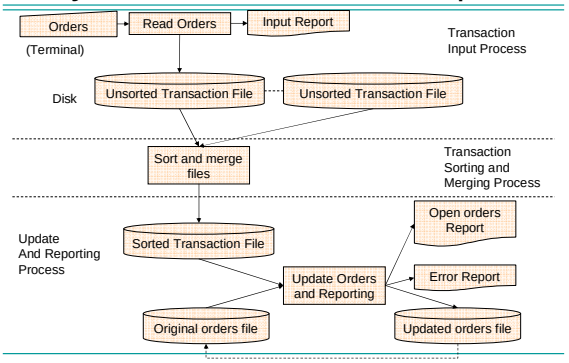
Begin "Main"
n_students = 0, totalgrade = 0
more = "y"
While more = "y" Do:
  Perform "Provide a Grade"
  Perform "Grade Type"
  Increment n_students
  Add grade to totalgrade
  Print "Are there more grades? Press y or n"
  Read variable more
End While
Perform "Final Output"
End



System Flow Chart

- Summarizes the interactions between parts of an information system using arrows and lines
  - Do not show information at the level of logic and procedures
- Rules:
  - Each computer is a square
  - Each disk is a cylinder
  - Each file is a box with a curvy bottom
  - Processes are shown with arrows to show the flow of work or information
  - Arrows indicate input/output relations between computers, documents, disks

System Flow Chart: Example



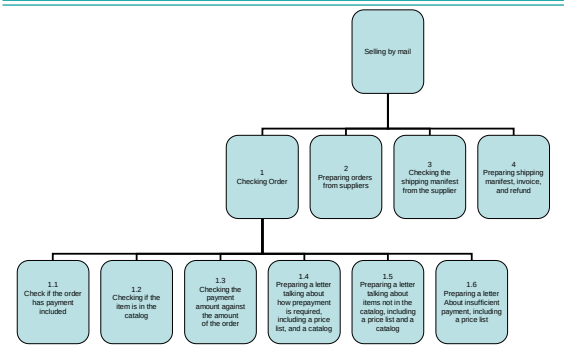
HIPO

- Hierarchy Plus Input-Process-Output have two types of diagrams
- Vertical Table of Contents (VTOC)
  - At the top are high level processes (in boxes)
  - Each box has a line leading to any child or sub-process
  - Boxes are numbered to show hierarchy
- Input-Process-Output Diagrams:
  - Show each process in three columns:
    - Inputs, process name, outputs
  - Graph can show multiple processes stacked
  - Graphs can be at different levels of detail

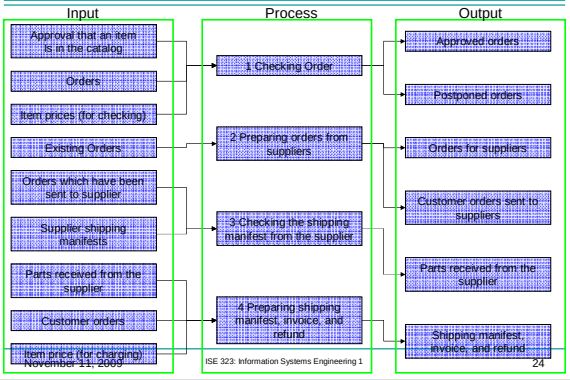
HIPO: Example

- Natural Language: A supply company receives orders by mail for items in its catalog.
- It returns merchandise by mail as well.
- The company does not keep any stock in house.
- When it receives an order, it checks the items listed in its catalog and the amount of payment included with the order (all orders must be prepaid).
- Approved orders are then sorted by the manufacturers of the items ordered.
- The company then sends out orders from the manufacturers according to the orders it received.
- Customer orders are sent out when the company receives the items from the manufacturers.

VTOC: Example



IPO: Example



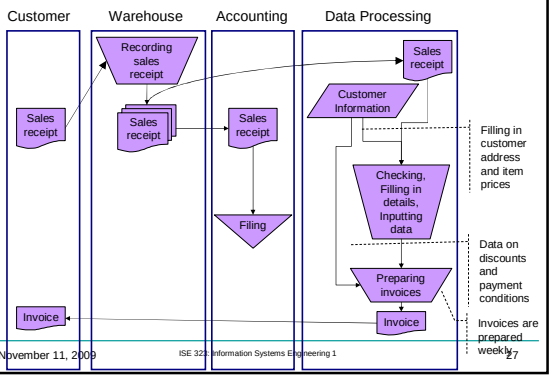
Process Chart

- Shows a work flow diagram which is not necessarily computerized
  - Can be useful for showing existing state when discussing non-computerized (or half-computerized) processes
- Rules:
  - Graph divided into columns to show the people or workers who are involved
  - Each item represents an action, input, or document which is processed
  - Arrows between the items show the process and flow of work

Process Chart: Example

- Natural Language: The process starts with a customer filling out a sales receipt. It is then passed to the warehouse, copied for filing by the accounting staff, processed in a few ways by the data processing staff, and then returned as an invoice for the customer.

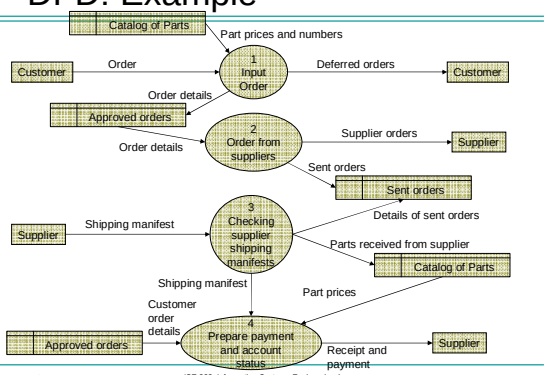
Process Chart: Example



Data Flow Diagram (DFD)

- Similar to HIPO, just it shows some of the cause and effect relationships better
  - Doesn't show internal process logic, just inputs/outputs
  - Shows data sources and user interactions
  - Can be more specific by including lower level processes
- Rules:
  - Processes are circles with names and numbers
  - Players are squares
  - Data sources are rectangles
  - Arrows between parts show flow or data or information
- We're going to talk a lot more about DFDs in the course of the semester

DFD: Example



DFD: Example

1. Function 1 "Input Order" records customer order details. It compares certain values (such as prices) which are found in the "Catalog of Parts" data sources, and then either updates (adds a record to) the "Approved orders" data source or sends a message to the customer if the ordered is defective.
2. Function 2 "Order from Suppliers" prepares and sends orders to the appropriate suppliers and records the orders in the "Sent orders" data source.
3. Function 3 "Checking supplier shipping manifests" is performed when the merchandise arrives from the suppliers. Along with the shipping manifest, it checks the contents of the delivery received, and updates the item catalog (with what's now in stock). It then sends data to Function 4.
4. Function 4 "Prepare payment and account status" prepares a receipt and payment for the supplier. It checks the shipping manifest against the orders from the customers as well.

So far

- Return to the roadmap
- Describing Existing State and Needs
  - Documentation and its parts
  - Documentation techniques
- Initial Design for Information Systems
  - Initial Design Documentation
  - Defining the Goals of a New Information System and Organizational Constraints
  - Business Processes and Computerized Processes
  - Describing a Computerized Process
  - Systemic Qualities and Size Approximation
  - Using a Prototype for the Initial Design

Initial Design – אפיון הראשוני

- The initial design is a document based on:
  - The existing state of the system
  - Requested fixes and additional functionality
  - Business requirements of the organization
  - Organizational constraints
- It's the basis for the Request for Proposals (RFP)
  - Who is it sent to?

Initial Design and Requirements

- The line between current state and requirements is blurry
- If the current state phase was done in depth, the initial design is easy
  - If not, the work must be done now to organize what's needed
- The initial design must be well done (since it's the heart of the RFP)
  - Suppliers who get an insufficiently detailed RFP will either:
    - Not respond
    - Make lots of assumptions about what is needed
- In Mafteach מפתח:
  - Initial Design is part of "Design" phase
  - After finishing "Design", you go right to "RFP"

Consultants for the Design

- An alternative approach: Bring in outside help to write the Initial Design
  - Write a high level characterization document first
- Can be helpful for
  - Small companies who don't have the resources to do a proper initial design
  - Large companies who want a big system too complex for them to design themselves
- Can help later on with system analysis as well
  - We must examine the existing state and derive a design from it
- We'll talk about a middle path – the organization writes the initial design, but the developers do the analysis

Defining Goals of a New Info Sys

- Defining goals for a new information system includes a prioritization of needs
- Must define system scope
- Goals and scope are defined in terms of the existing state results
- When defining goals, keep in mind organizational constraints
  - Should be quantified as much as possible and concretized

Defining the Goals

- Goals for the information system should be quantitative and specific
  - "It should be easy to use"
  - "It should work fast"
  - "The website's catalog response time should be 50 milliseconds or less during peak usage"
- Each goal must have a reason – so we know what we can relax
  - Why 50 milliseconds?
- Example: For a bank's credit department, "Improve customer loan collection"
  - Produce a monthly report of all accounts more than 45 days past due
  - For all accounts in minus, apply an interest rate according to the type of customer and the size of the debt

Defining Priorities

- Define priorities for each part and requirement identified
  - Based on practicality, use, and organizational need
- Know where to work first in case:
  - We run over time
  - We run over budget

Defining Scope and Boundaries

- Define the boundaries and scope at the beginning
  - Know what the system will do and what it will not do
- Example limitations:
  - Limitations to do only part of an organization's needs
  - Restriction to use in a geographic area
  - Reduction in the types of services provided
  - Reduction in the amount of history maintained
  - Limitations to types of customers
- Can help in maintaining physical, budgetary, or strategic goals
  - We may adjust them later during the feasibility study

Organizational Constraints

- Time Constraints
  - System needs to be finished by a given date.
  - If it's not feasible or missed, the project may need to be canceled.
- Technical Constraints
  - Hardware, software, and technical limitations of the organization's equipment.
  - If new equipment or products must be acquired, management must make the investment in the new materials.
  - There may be a bias against changes to commonly used systems in the organization.
  - Technical requirements approved in this phase must be approved and not subject to later review during the "feasibility study"
- Cost Constraints
  - System has a budget which must be kept to.
  - If the costs get out of control and beyond the budget, the system might need to be scaled down or canceled.
- Manpower Constraints
  - The number of people needed to design, develop, maintain, and use the new system.
  - Even if the design and development will take place outside, support and use may take place in house.
  - Often development or design groups have specific ways of working which are not easily changed.

Organizational Constraints

- Organizational/Industrial Constraints
  - The organization's processes and culture must be understood well. If the new system's design will work only if the organization's structure or work patterns change, there will be a problem.
  - When will be system be running? When can it go down for maintenance?
- Fundamental Assumptions
  - Special constraints which need to be acknowledged
  - Quantify any a fundamental assumption about the quantity of inputs, outputs, processes, or transactions that the system will perform
  - Consider expectation for growth in the organization
  - Consider any assumptions about the business climate, technology availability, or political situation

Critical Conditions

- Some conditions will be judged as "make or break" critical conditions
- Define them correctly
  - Don't incorrectly label too many (or too few) as critical
- Critical conditions will be used to judge feasibility of the new system

Business and Computerized Processes

- The functional description should take business processes into consideration
- A **business process** is an action (or set of actions) which the organization performs as part of its duty or behavior
  - Can be computerized, manual
  - Can be performed by one worker or by many together

Business Processes: Examples

- Prepare a quarterly budget.
- Order supplies from a supplier.
- Produce an item for sale.
- Prepare receipts for customers.
- Calculate and pay monthly salaries.
- Calculate and pay taxes.
- Sell items to a customer.
- Decide yearly sales targets.

- Business processes are divided up by the organization's structure and parts

Business Processes and Initial Design

- The purpose of identifying business processes is to consider which can be supported or performed by the information system
  - These are *computerized processes*
- We identified business processes in the existing state
  - List can help identify business processes now done and which can be computerized
  - If the list was not made yet, it must be assembled now
- When we are done, we must have a list of all business processes which are being computerized
  - Idea of how they run now
  - Plan for how they will run in the future

Scope of a Computerized Process

- The new IS may replace, perform, or help some business processes
  - IS meant to aid users
  - May automate or just alerting
- Some processes can be done only one way
- Some have multiple ways to be done
  - Now is normally not the time to decide this
  - Steering committee may intercede and decide
  - If no decision, write down all options and they will be evaluated later
- Some computerized processes have multiple smaller steps:
  - Inputs, outputs, processing, updating data sources

Describing a Computerized Process

Users
- List of the users for the computerized process.
- Could be individuals, classes of users, members of a group or division, or roles within a group.

Reasons why the process is performed
- Business processes may be performed based on some outside trigger or event.
- Triggers must be identified.
- Often more than one trigger exists for a process.
- Some triggers lead to multiple processes
- Triggers can be outside events, scheduled events, or initiative taken by an internal actor.

Periodicity of the event or trigger
- Some triggers are events which occur on a fixed or predictable schedule.
- Others are occasional events which have no predictable pattern.
- The system triggers and their periodicity aid in designing the kind of response time expected and its potential load.

Process Inputs
- Not all processes involve the collection of new data
- Those that do need to have a detailed description of the types of input that they accept.
- Some potential ways to describe inputs:
  - Source of the inputs and where they arrive from
  - Method of input
  - Details of data to be input
  - Size of the data should be considered
  - Display of the input

Process Outputs
- Many computerized processes have outputs which can be characterized in different ways.
- Some ways to describe and characterize outputs:
  - Target
  - Output Means and Method of Transmission
  - Detail of the Output Information
  - Size of the data
  - Sample Output

Describing a Computerized Process

Data stored and retrieved from data sources
- Most processes include reading or writing from some data source.
- At this level we are not concerned with the particulars of the data source used, whether its a file, a database, or a physical medium.
- Most details of the data source, storage, and reading mechanisms can be elided for simplicity.
- For each read and write action to a data source, the following items should be clear:
  - What is the type of the action
  - Data to be written/read
  - Size

Describing the Process Logic
- For certain important computerized processes, it is important to describe their logic.
- The logic does not need to be at a deep level, just enough to be understood by a reader
- For processes which have complicated needs, a detailed description might be justified.
- The process logic can be shown in any of the ways described above
  - Natural Language
  - Structured Language and Flow Charts
  - Data Flow Diagram

Other Plans and Needs
- Special considerations and processes in the system
- Response times
- Online or batch mode
- Execution permissions and access restrictions

Systemic Qualities

- Some aspects of the system are not at the process level:
  - Interface requirements (GUI, web, thin client)
- Such requirements should be specified separately
  - Made clear similar to above
- Presence of expert users may push for more detailed or complicated interfaces

Size Approximation

- Must have some idea of size for inputs, outputs, and storage
- Based on this, the system designers can make decisions
- Size affects response time, architectural support
- Number of users affects interface design, concurrency support, and training

Using a Prototype

- A mockup prototype can be made now to display some aspects of the system
  - Can't be too detailed – we don't have them yet
- Can show mockups of special inputs and outputs to give them attention
- Can make mockup GUIs easily too
- Processes can be mocked up using special visualization software

Conclusion

- Return to the roadmap
- Describing Existing State and Needs
  - Documentation and its parts
  - Documentation techniques
- Initial Design for Information Systems
  - Initial Design Documentation
  - Defining the Goals of a New Information System and Organizational Constraints
  - Business Processes and Computerized Processes
  - Describing a Computerized Process
  - Systemic Qualities and Size Approximation
  - Using a Prototype for the Initial Design