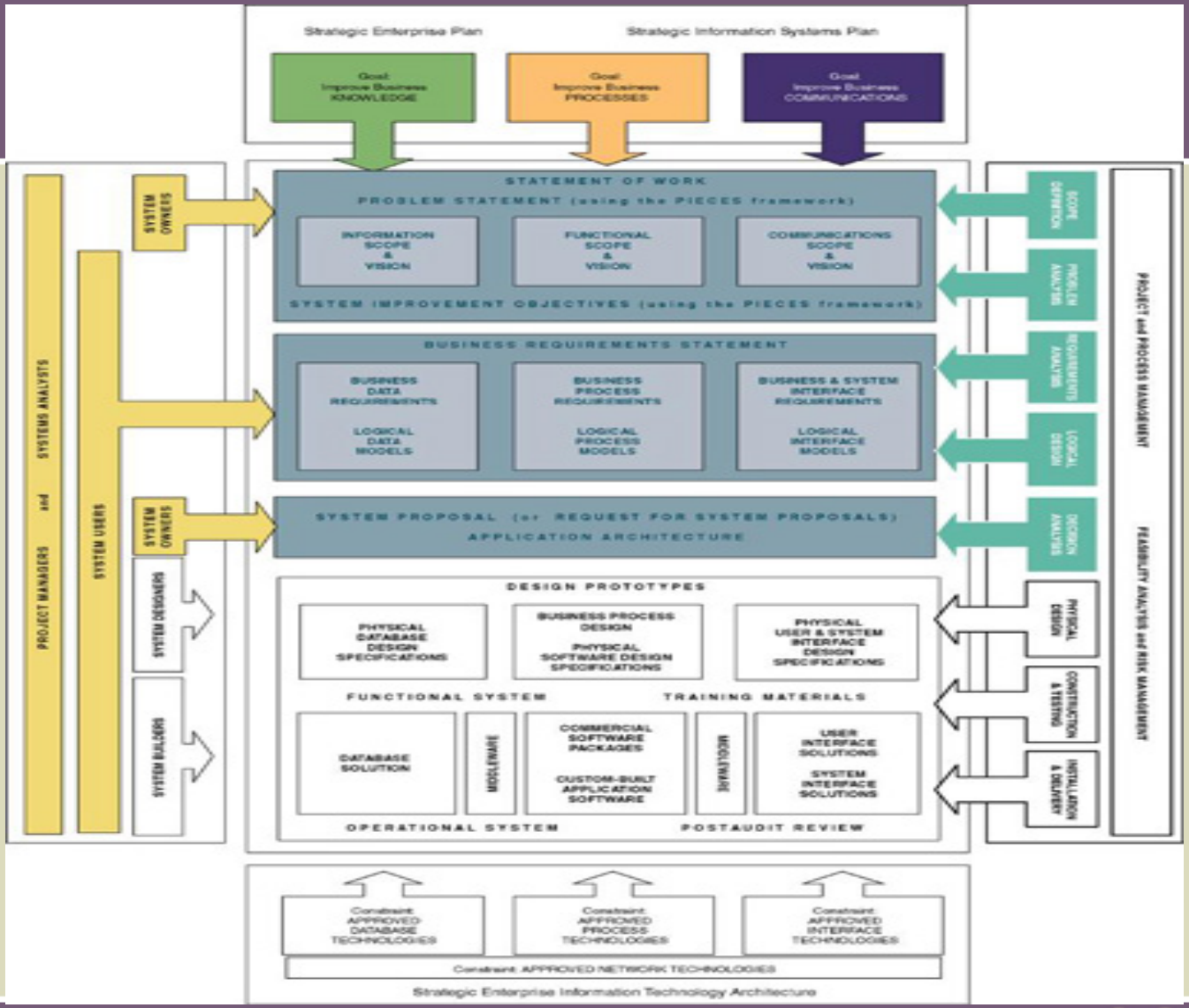
Chapter 5

Systems Analysis

Objectives

- Define systems analysis and relate it to the scope definition, problem analysis, requirements analysis, logical design, and decision analysis phases.
- Describe a number of systems analysis approaches for solving business system problems.
- Describe scope definition, problem analysis, requirements analysis, logical design, and decision analysis phases in terms of information system building blocks.
- Describe scope definition, problem analysis, requirements analysis, logical design, and decision analysis phases in terms of purpose, participants, inputs, outputs, techniques, and steps.
- Identify those chapters in this textbook that can help you learn specific systems analysis tools and techniques.


Strategic Enterprise Plan
Strategic Information Systems Plan
Goal: Improve Business KNOWLEDGE
Goal: Improve Business PROCESSES
Goal: Improve Business COMMUNICATIONS
STATEMENT OF WORK
PROBLEM STATEMENT (using the PIECES framework)
INFORMATION SCOPE & VISION
FUNCTIONAL SCOPE & VISION
COMMUNICATIONS SCOPE & VISION
SYSTEM IMPROVEMENT OBJECTIVES (using the PIECES framework)
BUSINESS REQUIREMENTS STATEMENT
BUSINESS DATA REQUIREMENTS
LOGICAL DATA MODELS
BUSINESS PROCESS REQUIREMENTS
LOGICAL PROCESS MODELS
BUSINESS & SYSTEM INTERFACE REQUIREMENTS
LOGICAL INTERFACE MODELS
SYSTEM PROPOSAL (or REQUEST FOR SYSTEM PROPOSALS)
APPLICATION ARCHITECTURE
DESIGN PROTOTYPES
PHYSICAL DATABASE DESIGN SPECIFICATIONS
BUSINESS PROCESS DESIGN
PHYSICAL SOFTWARE DESIGN SPECIFICATIONS
PHYSICAL USER & SYSTEM INTERFACE DESIGN SPECIFICATIONS
FUNCTIONAL SYSTEM
TRAINING MATERIALS
DATABASE SOLUTION
MIDDLEWARE
COMMERCIAL SOFTWARE PACKAGES
CUSTOM-BUILT APPLICATION SOFTWARE
MIDDLEWARE
USER INTERFACE SOLUTIONS
SYSTEM INTERFACE SOLUTIONS
OPERATIONAL SYSTEM
POSTAUDIT REVIEW
Constraint: APPROVED DATABASE TECHNOLOGIES
Constraint: APPROVED PROCESS TECHNOLOGIES
Constraint: APPROVED INTERFACE TECHNOLOGIES
Constraint: APPROVED NETWORK TECHNOLOGIES
Strategic Enterprise Information Technology Architecture
SYSTEM OWNERS
SYSTEM USERS
SYSTEM DESIGNERS
SYSTEM BUILDERS
SYSTEMS ANALYSTS and PROJECT MANAGERS
SCOPE DEFINITION
PROBLEM ANALYSIS
REQUIREMENTS ANALYSIS
LOGICAL DESIGN
DESIGN ANALYSIS
PHYSICAL DESIGN
CONSTRUCTION & TESTING
INSTALLATION & DELIVERY
PROJECT and PROCESS MANAGEMENT
FEASIBILITY ANALYSIS and RISK MANAGEMENT

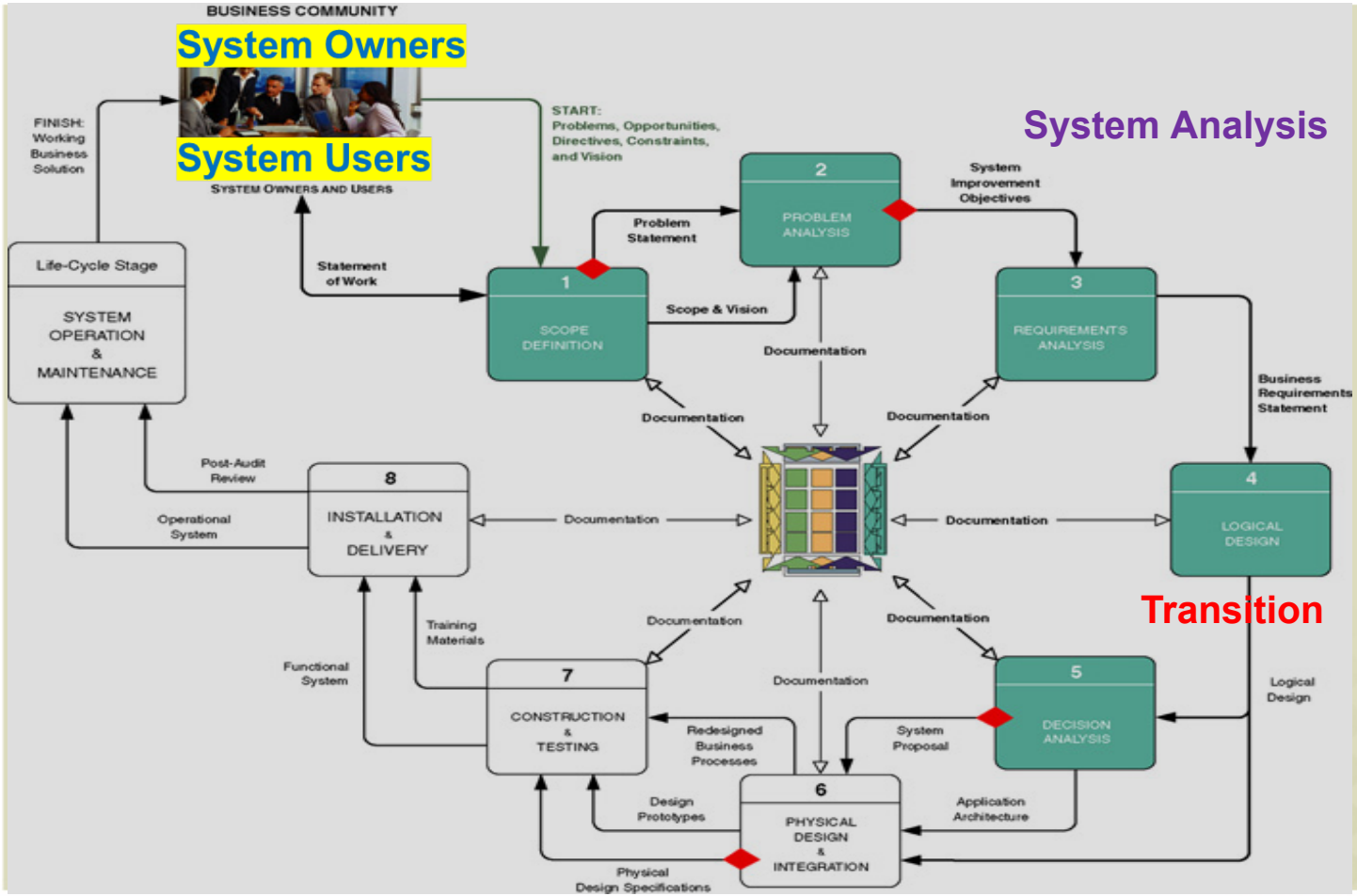
What is Systems Analysis ?

Systems analysis – a problem-solving technique that decomposes a system into its component pieces for the purpose of studying how well those component parts work and interact to accomplish their purpose.

Systems design – a complementary problem-solving technique (to systems analysis) that reassembles a system's component pieces back into a complete system—hopefully, an improved system. This may involves adding, deleting, and changing pieces relative to the original system.

Information systems analysis – those development phases in an information systems development project the primarily focus on the business problem and requirements, independent of any technology that can or will be used to implement a solution to that problem.

Context of Systems Analysis



Repository

Repository – a location (or set of locations) where systems analysts, systems designers, and system builders keep all of the documentation associated with one or more systems or projects.

- Network directory of computer-generated files that contain project correspondence, reports, and data
- CASE tool dictionary or encyclopedia (Chapter 3)
- Printed documentation (binders and system libraries)
- Intranet website interface to the above components

Model-Driven Analysis Methods

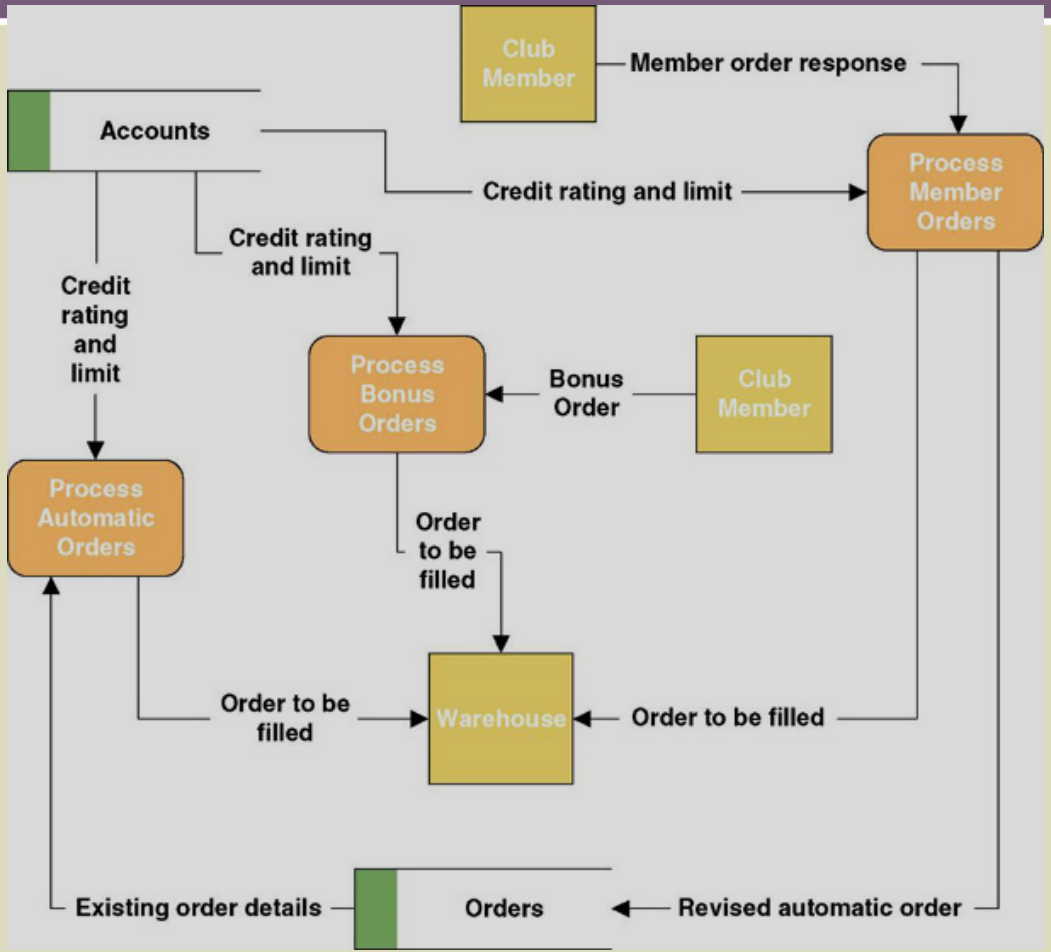
Model-driven analysis – a problem-solving approach that emphasizes the drawing of pictorial system models to document and validate both existing and/or proposed systems. Ultimately, the system model becomes the blueprint for designing and constructing an improved system.

Model – a representation of either reality or vision. Since "a picture is worth a thousand words," most models use pictures to represent the reality or vision.

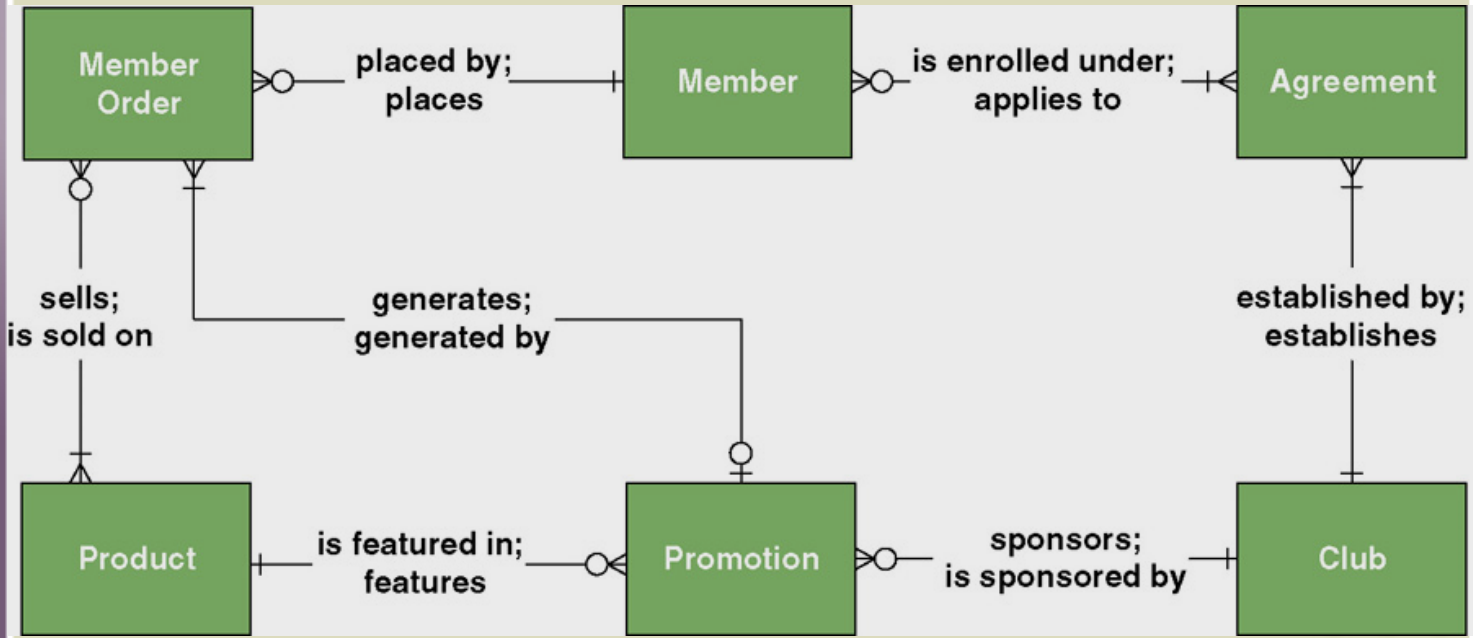
Model-Driven Approaches

- **Traditional Approaches**
  - **Structured Analysis**
    - Focuses on the flow of data through processes
    - Key model: data flow diagram (DFD)
  - **Information Engineering**
    - Focuses on structure of stored data
    - Key model: entity relationship diagram (ERD)
- **Object-Oriented Approach**
  - integrates data and process concerns into objects
    - Object – the encapsulation of the data (called properties) that describes a discrete person, object, place, event, or thing, with all the processes (called methods) that are allowed to use or update the data and properties. The only way to access or update the object's data is to use the object's predefined processes.
  - **Unified Modeling Language (UML)**

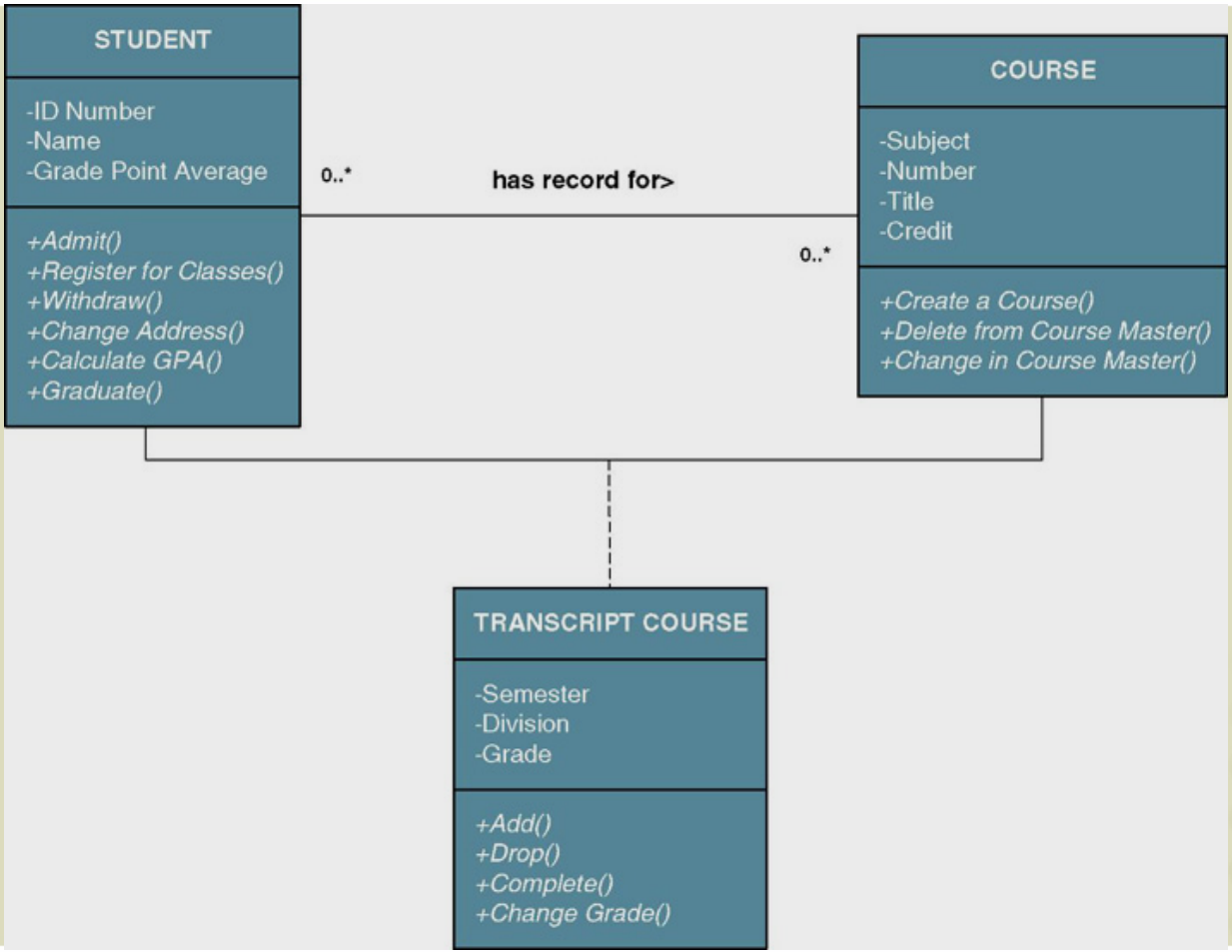
A Simple Process Model



A Simple Data Model



A Simple Object Model



Accelerated Systems Analysis

Accelerated systems analysis approaches emphasize the construction of prototypes to more rapidly identify business and user requirements for a new system.

prototype – a small-scale, incomplete, but working sample of a desired system.

- Accelerated systems analysis approaches
  - Discovery Prototyping
  - Rapid Architected Analysis

Discovery Prototyping

Discovery prototyping – a technique used to identify the users' business requirements by having them react to a quick-and-dirty implementation of those requirements.

- Advantages
  - Prototypes cater to the "I'll know what I want when I see it" way of thinking that is characteristic of many users and managers.
- Disadvantages
  - Can become preoccupied with final "look and feel" prematurely
  - Can encourage a premature focus on, and commitment to, design
  - Users can be misled to believe that the completed system can be built rapidly using prototyping tools

Rapid Architected Analysis

Rapid architected analysis – an approach that attempts to derive system models (as described earlier in this section) from existing systems or discovery prototypes.

- **Reverse engineering** – the use of technology that reads the program code for an existing database, application program, and/or user interface and automatically generates the equivalent system model.

Requirements Discovery

Requirements discovery – the process, used by systems analysts of identifying or extracting (Elicit) system problems and solution requirements from the user community.

Elicit= Collect = Extract

Requirements Discovery Methods

- **Fact-finding** – the process of collecting information about system problems, opportunities, solution requirements, and priorities.
  - Sampling existing documentation, reports, forms, databases, etc
  - Research of relevant literature
  - Observation of the current system
  - Questionnaires and surveys
  - Interviews
- **Joint requirements planning** (JRP) –use of facilitated workshops to bring together all of the system owners, users, and analysts, and some systems designer and builders to jointly perform systems analysis.
  - Considered a part of a larger method called joint application development (JAD), a more comprehensive application of the JRP techniques to the entire systems development process.

Business Process Redesign

Business process redesign (BPR) – the application of systems analysis methods to the goal of dramatically changing and improving the fundamental business processes of an organization, independent of information technology.

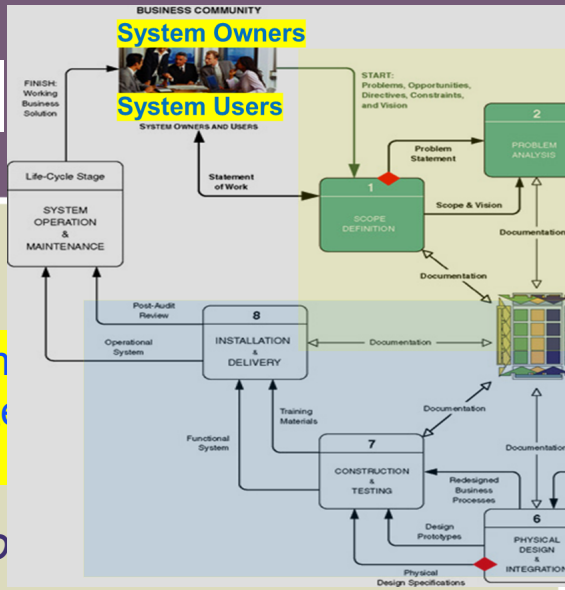
Agile Methods

Agile method – integration of various approaches of systems analysis and design for applications as deemed appropriate to problem being solved and the system being developed.

SDLC= System Development Life Cycle

- Most commercial methodologies do not impose a single approach (structured analysis, IE, OOA) on systems analysts.
- Instead, they integrate all popular approaches into a collection of agile methods.
- System developers are given the flexibility to select from a variety of tools and techniques to best accomplish the tasks at hand,
- Hypothetical *FAST* methodology operates this way.

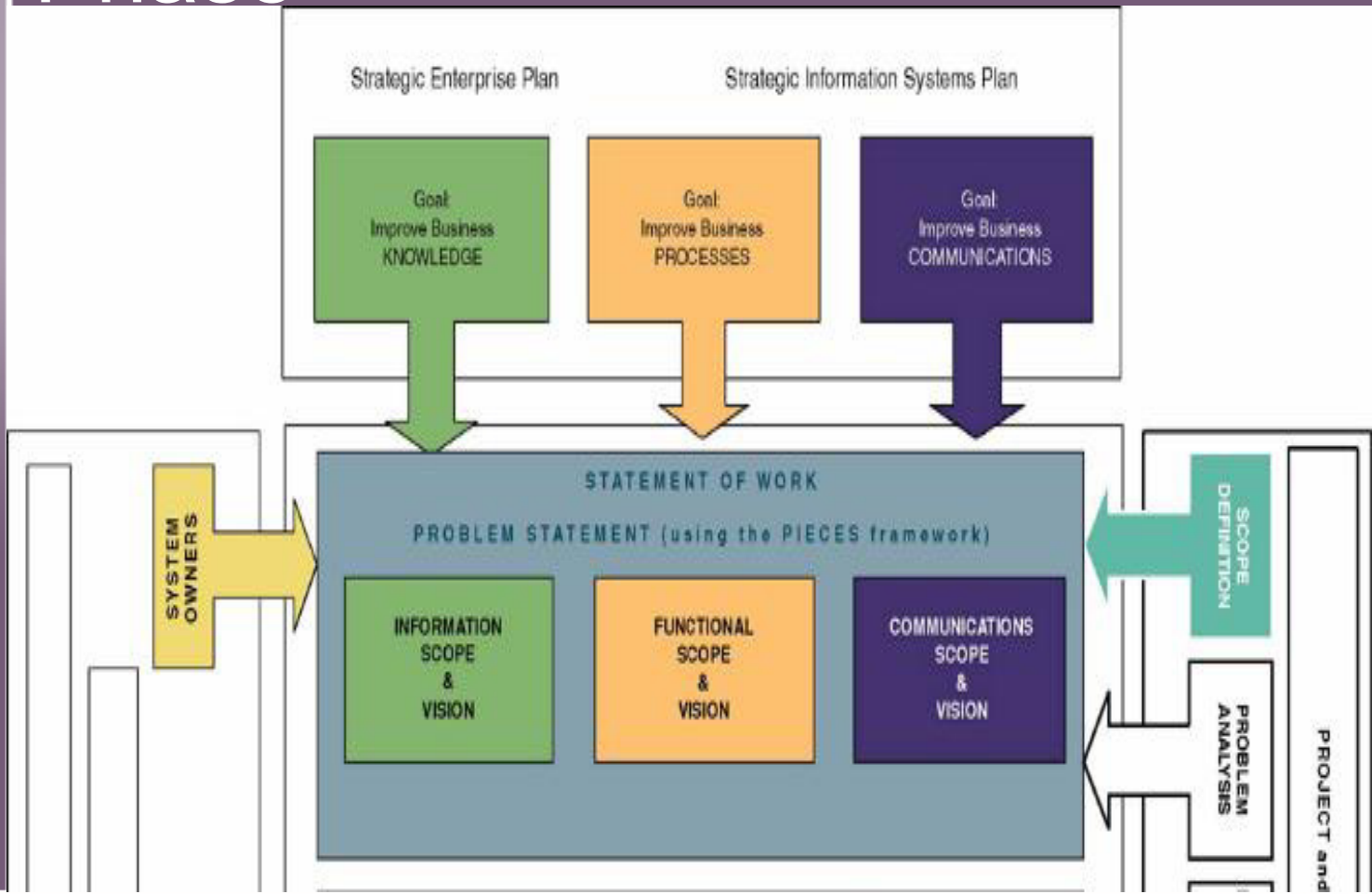
FAST Systems Anal



- Scope Definition Phase
  - *Is the project worth looking at?*
- Problem Analysis Phase
  - *Is a new system worth building?*
- Requirements Analysis Phase
  - *What do the users need and want fro*
- Logical Design Phase
  - *<u>What</u> must the new system do?*
- Decision Analysis Phase
  - *What is the best solution?*

Fran
syste

Context of Scope Definition Phase



Ta… P…

Project Request or Assignment
SYSTEM OWNERS AND USERS (OR STEERING COMMITTEE)
PROJECT CHARTER
1.1
Identify baseline problems and opportunities
Preliminary Problem Statement
problem statements (PIECES)
1.2
Negotiate baseline scope
statements of project scope
1.5
Communicate the project plan
problem statements and scope
Repository
Preliminary Problem Statement with Scope
1.3
Assess baseline worthiness
statement of work project schedule and resource assignments
problem statements with scope
1.4
Develop baseline schedule & budget
Baseline Project Plan and Schedule
(project is worthy)

Key Terms for Scope Definition Phase

Steering body – a committee of executive business and system managers that studies and prioritizes competing project proposals to determine which projects will return the most value to the organization and thus should be approved for continues systems development.

- Also called a *steering committee*.

Project charter – the final deliverable for the preliminary investigation phase. A project charter defines the project scope, plan, methodology, standards, and so on.

- Preliminary master plan includes preliminary schedule and resource assignments (also called a *baseline plan*).
- Detailed plan and schedule for completing the next phase of the project.

Sample Request for System Services

SoundStage Entertainment Club
Information System Services
Phone: 494-0666 Fax: 494-0999
Internet: http://www.soundstage.com
Intranet: http://www.soundstage.com/iss

REQUEST FOR INFORMATION SYSTEM SERVICES

DATE OF REQUEST	SERVICE REQUESTED FOR DEPARTMENT(S)
January 9, 2003	Member Services, Warehouse, Shipping

SUBMITTED BY (key user contact)
Name Sarah Hartman
Title Business Analyst, Member Services
Office B035
Phone 494-0867

EXECUTIVE SPONSOR (funding authority)
Name Galen Kirkhoff
Title Vice President, Member Services
Office G242
Phone 494-1242

TYPE OF SERVICE REQUESTED:
- ☐ Information Strategy Planning
- ☒ Business Process Analysis and Redesign
- ☒ New Application Development
- ☐ Other (please specify) _____
- ☐ Existing Application Enhancement
- ☐ Existing Application Maintenance (problem fix)
- ☐ Not Sure

BRIEF STATEMENT OF PROBLEM, OPPORTUNITY, OR DIRECTIVE (attach additional documentation as necessary)
The information strategy planning group has targeted member services, marketing, and order fulfillment (inclusive of shipping) for business process redesign and integrated application development. Currently serviced by separate information systems, these areas are not well integrated to maximize efficient order services to our members. The current systems are not adaptable to our rapidly changing products and services. In some cases, separate systems exist for similar products and services. Some of these systems were inherited through mergers that expanded our products and services. There also exist several marketing opportunities to increase our presence to our members. One example includes Internet commerce services. Finally, the automatic identification system being developed for the warehouse must fully interoperate with member services.

BRIEF STATEMENT OF EXPECTED SOLUTION
We envision completely new and streamlined business processes that minimize the response time to member orders for products and services. An order shall not be considered fulfilled until it has been received by the member. The new system should provide for expanded club and member flexibility and adaptability of basic business products and services.

We envision a system that extends to the desktop computers of both employees and members, with appropriate shared services provided across the network, consistent with the ISS distributed architecture. This is consistent with strategic plans to retire the AS/400 central computer and replace it with servers.

ACTION (ISS Office Use Only)

- ☐ **Feasibility assessment approved**
- ☒ **Feasibility assessment waived**
- ☐ **Request delayed**
- ☐ **Request rejected**

Assigned to Sandra Shepherd
Approved Budget $ 450,000
Start Date ASAP **Deadline** ASAP
Backlogged until date: _____
Reason: _____

Authorized Signatures:
Rebecca J. Todd
Chair, ISS Executive Steering Body

Galen Kirkhoff
Project Executive Sponsor

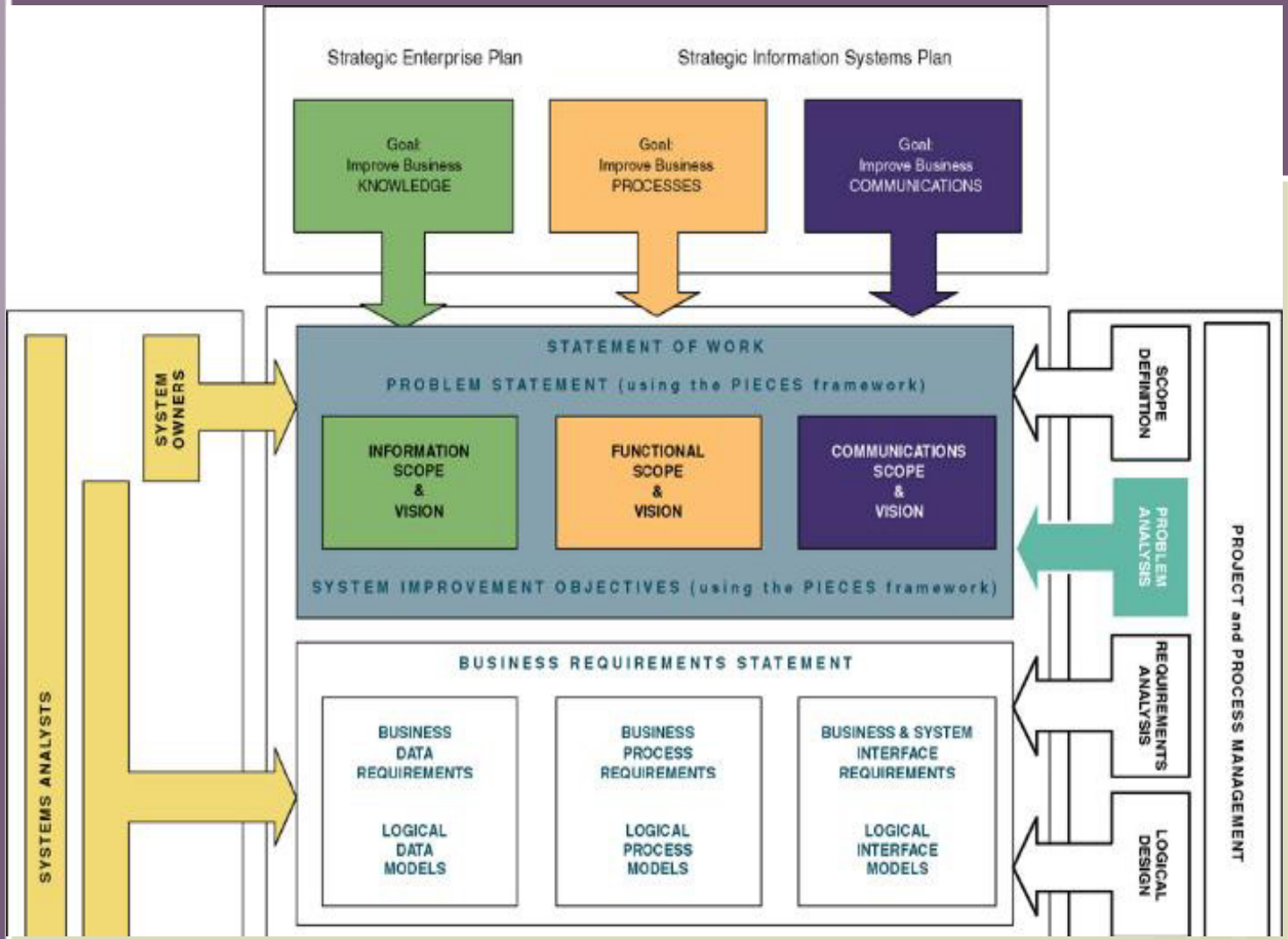
FORM ISS-100.RPSS (Last revised December, 1999)

Sample Problem Statements

Problem Statements

Project: Member services information system	Project manager: Sandra Shepherd
Created by: Sandra Shepherd	Last updated by: Robert Martinez
Date created: January 9, 2003	Date last updated: January 15, 2003

Brief Statements of Problem, Opportunity, or Directive	Urgency	Visibility	Annual Benefits	Priority or Rank	Proposed Solution
1. Order response time as measured from time of order receipt to time of customer delivery has increased to an average of 15 days.	ASAP	High	$175,000	2	New development
2. The recent acquisitions of Private Screenings Video Club and GameScreen will further stress the throughput requirements for the current system.	6 months	Med	75,000	2	New development
3. Currently, three different order entry systems service the audio, video, and game divisions. Each system is designed to interface with a different warehousing system; therefore, the intent to merge inventory into a single warehouse has been delayed.	6 months	Med	515,000	2	New development
4. There is a general lack of access to management and decision-making information. This will become exasperated by the acquisition of two additional order processing systems (from Private Screenings and GameScreen)	12 months	Low	15,000	3	After new system is developed, provide users with easy-to-learn and -use reporting tools.


Strategic Enterprise Plan
Strategic Information Systems Plan
Goal: Improve Business KNOWLEDGE
Goal: Improve Business PROCESSES
Goal: Improve Business COMMUNICATIONS
SYSTEM OWNERS
SYSTEMS ANALYSTS
STATEMENT OF WORK
PROBLEM STATEMENT (using the PIECES framework)
INFORMATION SCOPE & VISION
FUNCTIONAL SCOPE & VISION
COMMUNICATIONS SCOPE & VISION
SYSTEM IMPROVEMENT OBJECTIVES (using the PIECES framework)
BUSINESS REQUIREMENTS STATEMENT
BUSINESS DATA REQUIREMENTS
LOGICAL DATA MODELS
BUSINESS PROCESS REQUIREMENTS
LOGICAL PROCESS MODELS
BUSINESS & SYSTEM INTERFACE REQUIREMENTS
LOGICAL INTERFACE MODELS
SCOPE DEFINITION
PROBLEM ANALYSIS
REQUIREMENTS ANALYSIS
LOGICAL DESIGN
PROJECT and PROCESS MANAGEMENT


THE BUSINESS COMMUNITY
(approval to continue project - from preliminary investigation)
Project Charter
SYSTEM OWNERS AND USERS (OR STEERING COMMITTEE)
SYSTEM IMPROVEMENT OBJECTIVES
2.1
Understand the problem domain
Problem Domain and Business Vocabulary
2.2
Analyze problems and opportunities
current system documentation, system models
problem statements, cause/effect analyses
2.6
Communicate findings and recommenda-tions
problem analyses, system models, and system improvement objectives
Repository
problem domain, process models, process analysis
2.3 (opt)
Analyze business processes
project plan
problem analyses, system improvement objectives, and constraints
Updated Project Plan
2.5
Update or refine the project plan
System Improvement Objectives
2.4
Establish system improvement objectives

Key Terms of the Problem Analysis Phase

Cause-and-effect analysis – a technique in which problems are studied to determine their causes and effects.

> In practice, effects can be symptomatic of more deeply rooted problems which, in turn, must be analyzed for causes and effects until the causes and effects do not yield symptoms of other problems.

Context Diagram – a pictorial model that shows how the system interacts with the world around it and specifies in general terms the system inputs and outputs.

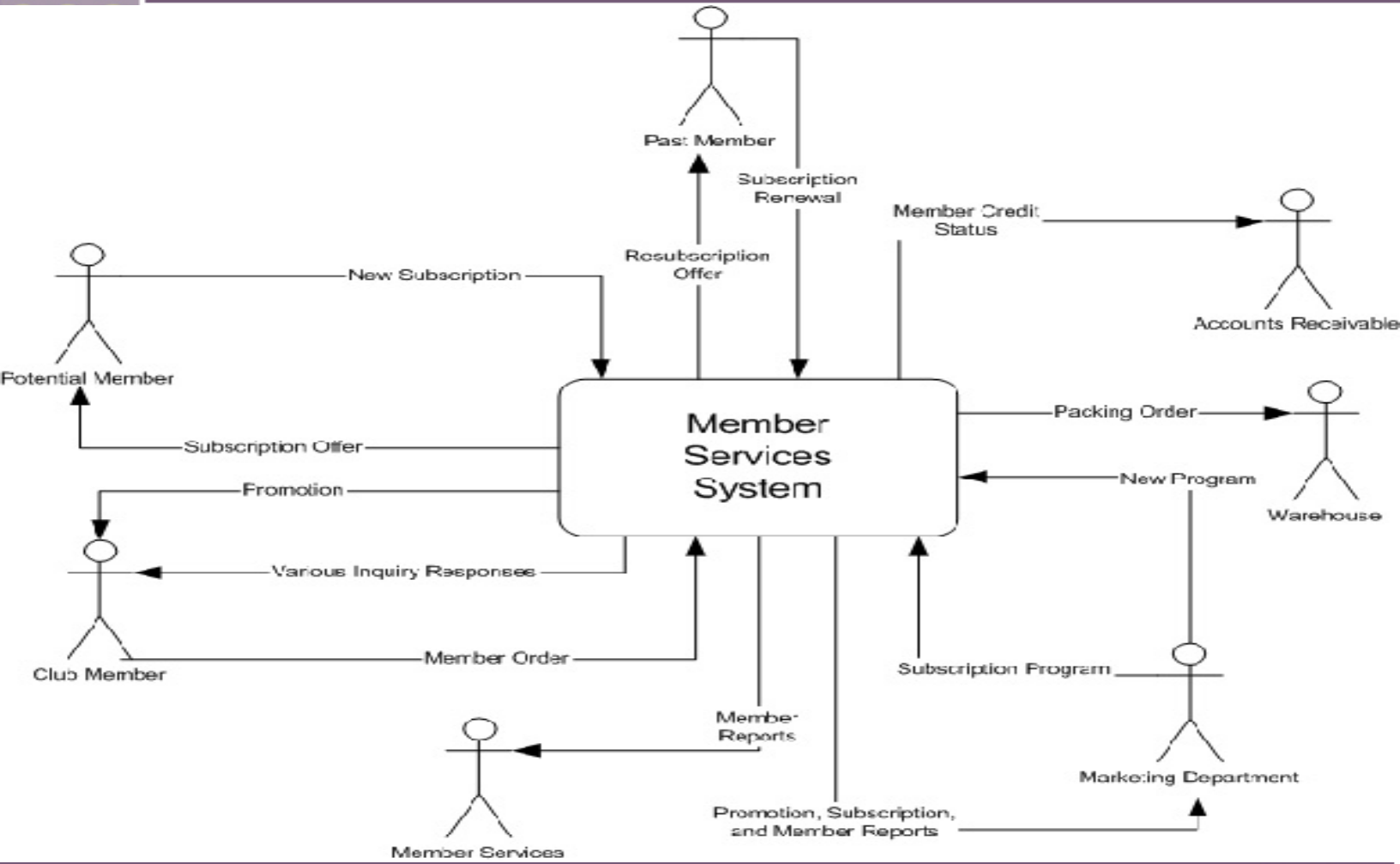
Sample Cause-and-Effect Analysis

PROBLEMS, OPPORTUNITIES, OBJECTIVES, AND CONSTRAINTS MATRIX

Project:	Member Services Information System	Project Manager:	Sandra Shepherd
Created by:	Robert Martinez	Last Updated by:	Robert Martinez
Date Created:	January 21, 2003	Date Last Updated:	January 31, 2003

CAUSE-AND-EFFECT ANALYSIS		SYSTEM IMPROVEMENT OBJECTIVES	
Problem or Opportunity	**Causes and Effects**	**System Objective**	**System Constraint**
1. Order response time is unacceptable.	1. Throughput has increased while number of order clerks was downsized. Time to process a single order has remained relatively constant. 2. System is too keyboard-dependent. Many of the same values are keyed for most orders. Net result is (with the current system) each order takes longer to process than is ideal. 3. Data editing is performed by the AS/400. As that computer has approached its capacity, order edit responses have slowed. Because order clerks are trying to work faster to keep up with the volume, the number of errors has increased. 4. Warehouse picking tickets for orders were never designed to maximize the efficiency of order fillers. As warehouse operations grew, order filling delays were inevitable.	1. Decrease the time required to process a single order by 30%. 2. Eliminate keyboard data entry for as much as 50% of all orders. 3. For remaining orders, reduce as many key-strokes as possible by replacing keystrokes with point-and-click objects on the computer display screen. 4. Move data editing from a shared computer to the desktop. 5. Replace existing picking tickets with a paperless communication system between member services and the warehouse.	1. There will be no increase in the order processing workforce. 2. Any system developed must be compatible with the existing Windows 95 desktop standard. 3. New system must be compatible with the already approved automatic identification system (for bar coding).

Sample Context Diagram



Key Terms of the Problem Analysis Phase (cont.)

Objective – a measure of success. It is something that you expect to achieve, if given sufficient resources.

- *Reduce the number of uncollectible customer accounts by 50 percent within the next year.*
- *Increase by 25 percent the number of loan applications that can be processed during an eight-hour shift.*
- *Decrease by 50 percent the time required to reschedule a production lot when a workstation malfunctions.*

Constraint – something that will limit your flexibility in defining a solution to your objectives. Essentially, constraints cannot be changed.

- *The new system must be operational by April 15.*
- *The new system cannot cost more than $350,000.*
- *The new system must be web-enabled.*
- *The new system must bill customers every 15 days.*

System Improvement Report Outline

I. Executive summary (approximately 2 pages)
- A. Summary of recommendation
- B. Summary of problems, opportunities, and directives
- C. Brief statement of system improvement objectives
- D. Brief explanation of report contents

II. Background information (approximately 2 pages)
- A. List of interviews and facilitated group meetings conducted
- B. List of other sources of information that were exploited
- C. Description of analytical techniques used

III. Overview of current system (approximately 5 pages)
- A. Strategic implications (if project is part of or impacts existing IS strategic plan)
- B. Models of the current system
  1. Interface model (showing project scope)
  2. Data model (showing project scope)
  3. Geographical models (showing project scope)
  4. Process model (showing functional decomposition only)

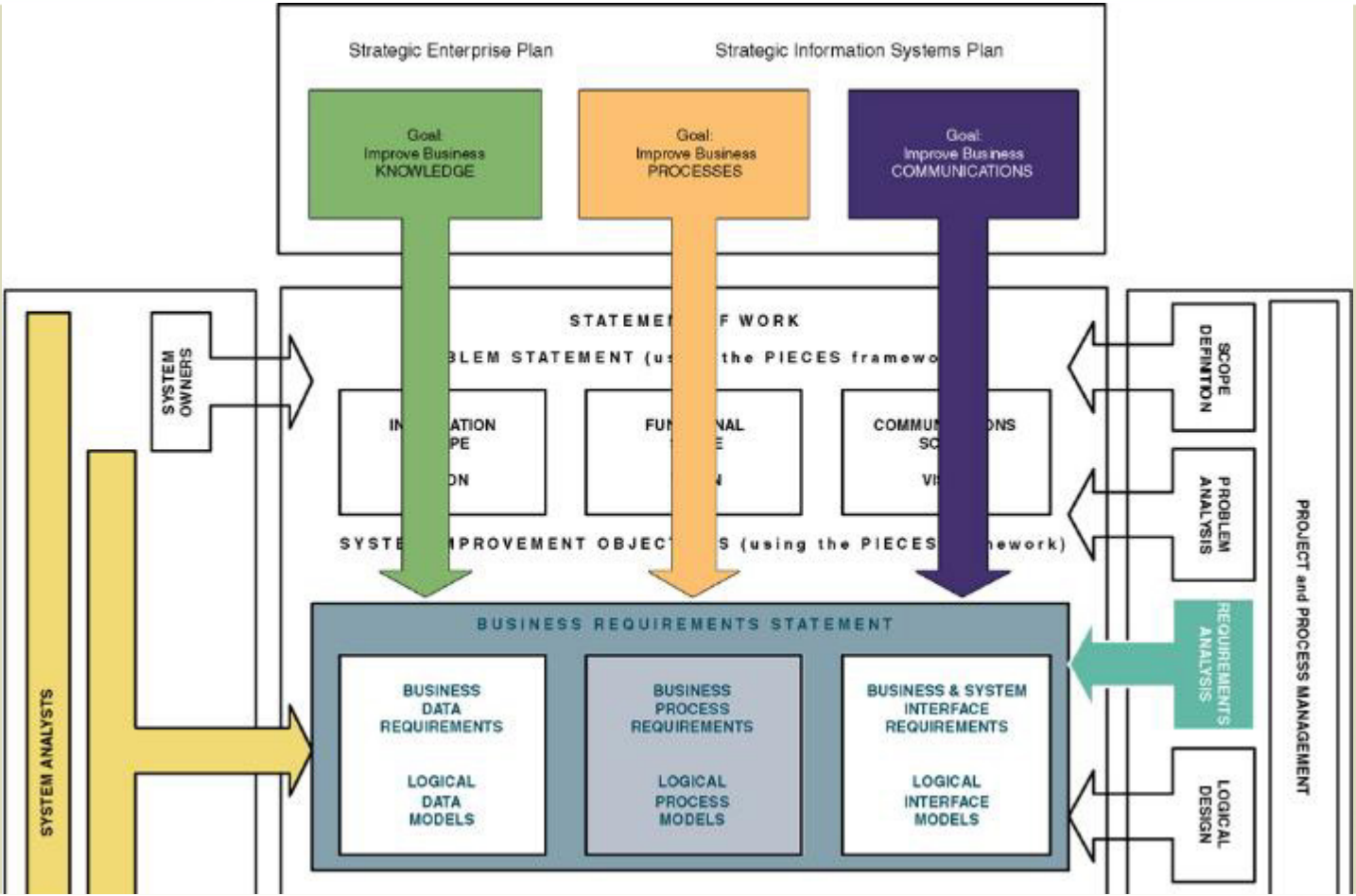
System Improvement Report Outline (cont.)

I. Analysis of the current system (approx. 5-10 pages)
- A. Performance problems, opportunities, cause-effect analysis
- B. Information problems, opportunities, cause-effect analysis
- C. Economic problems, opportunities, cause-effect analysis
- D. Control problems, opportunities, cause-effect analysis
- E. Efficiency problems, opportunities, cause-effect analysis
- F. Service problems, opportunities, and cause-effect analysis

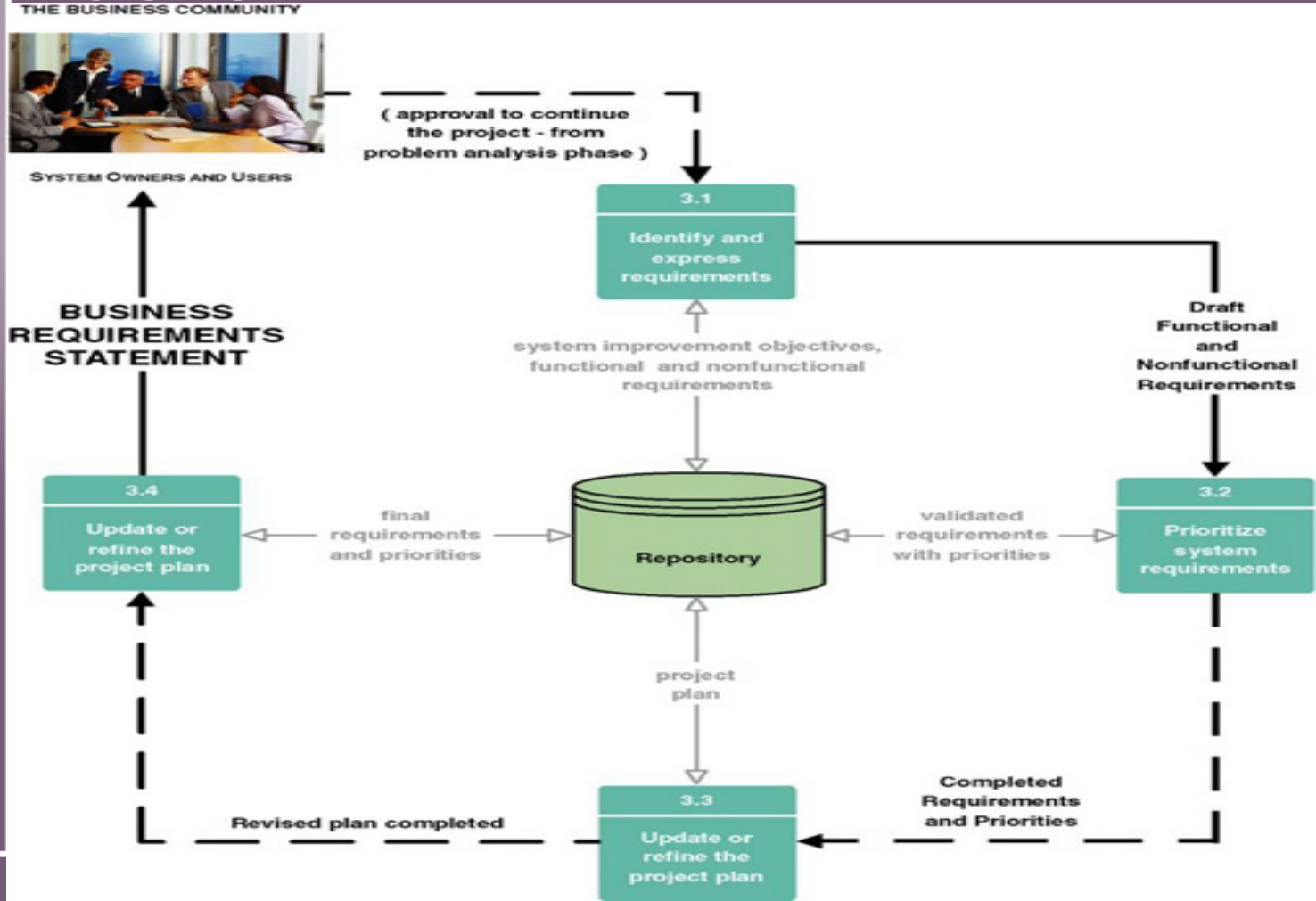
II. Detailed recommendations (approx. 5-10 pages)
- A. System improvement objectives and priorities
- B. Constraints
- C. Project Plan
  1. Scope reassessment and refinement
  2. Revised master plan
  3. Detailed plan for the definition phase

III. Appendixes
- A. Any detailed system models
- B. Other documents as appropriate

Context of Requirements Analysis Phase



Requirements Analysis Phase Tasks



Key Terms of Requirements Analysis Phase

Functional requirement – a description of activities and services a system must provide.

- inputs, outputs, processes, stored data.

Nonfunctional requirement – a description of other features, characteristics, and constraints that define a satisfactory system.

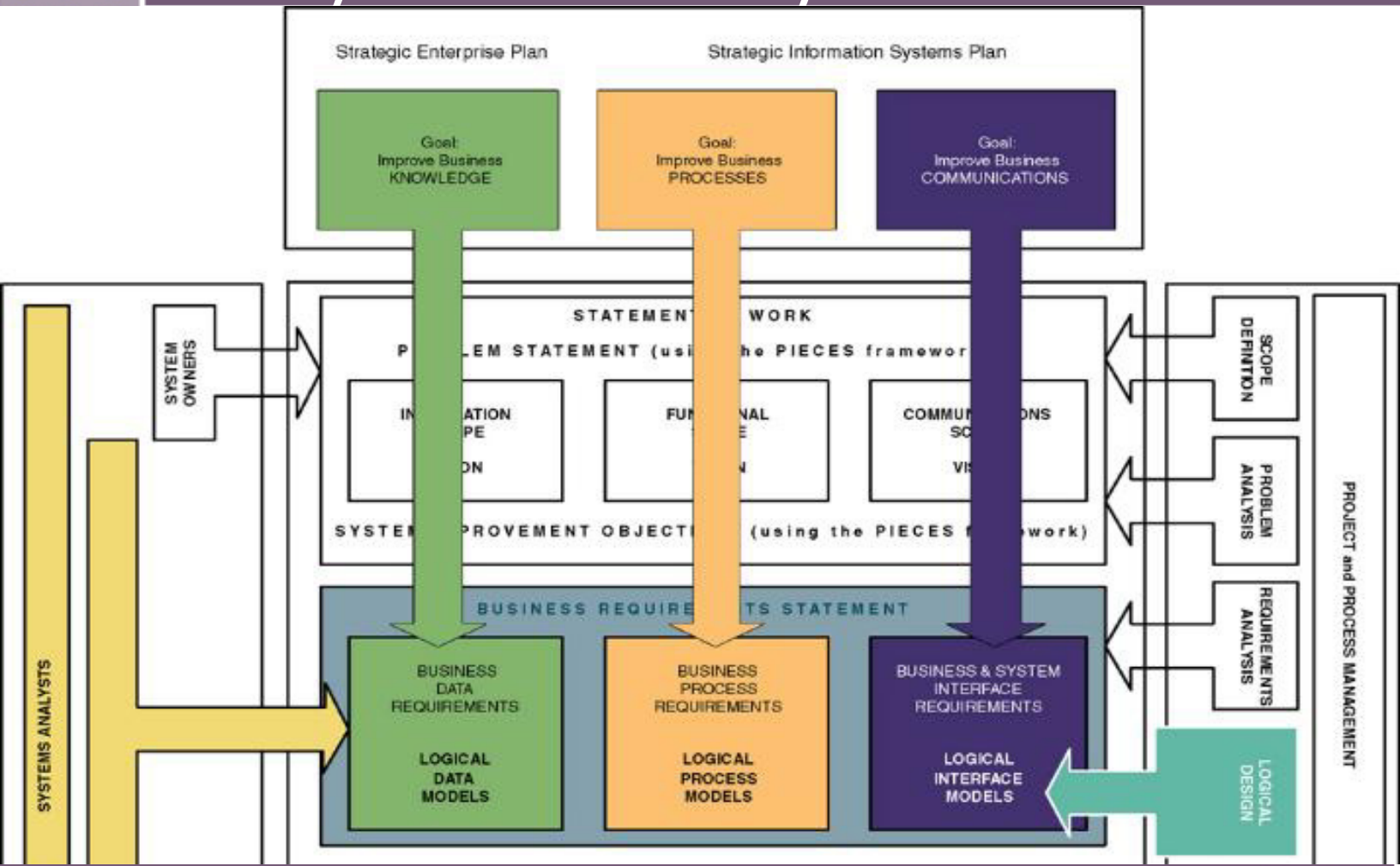
- Performance, ease of learning and use, budgets, deadlines, documentation, security, internal auditing controls.

Use case – a business scenario or event for which the system must provide a defined response. Use cases evolved out of object-oriented analysis; however, their use has become common in many other methodologies for systems analysis and design.

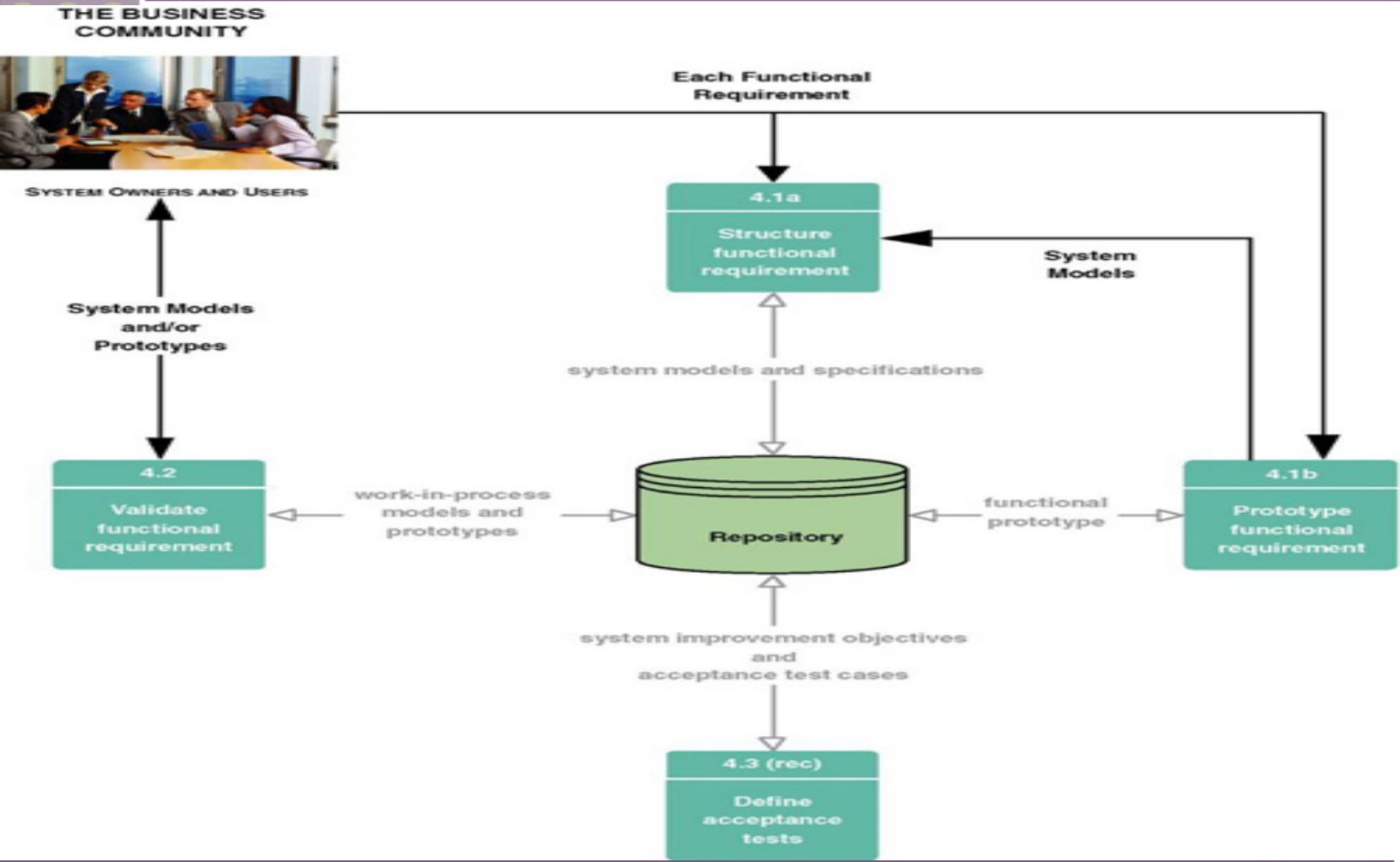
Timeboxing – a technique that delivers information systems functionality and requirements through versioning.

1. Development team selects the smallest subset of the system that, if fully implemented, will return a value to the systems owners and users.
2. That subset is developed, ideally with a time frame of 6-9 months or less.
3. Value-added versions of the system are developed in similar time frames.

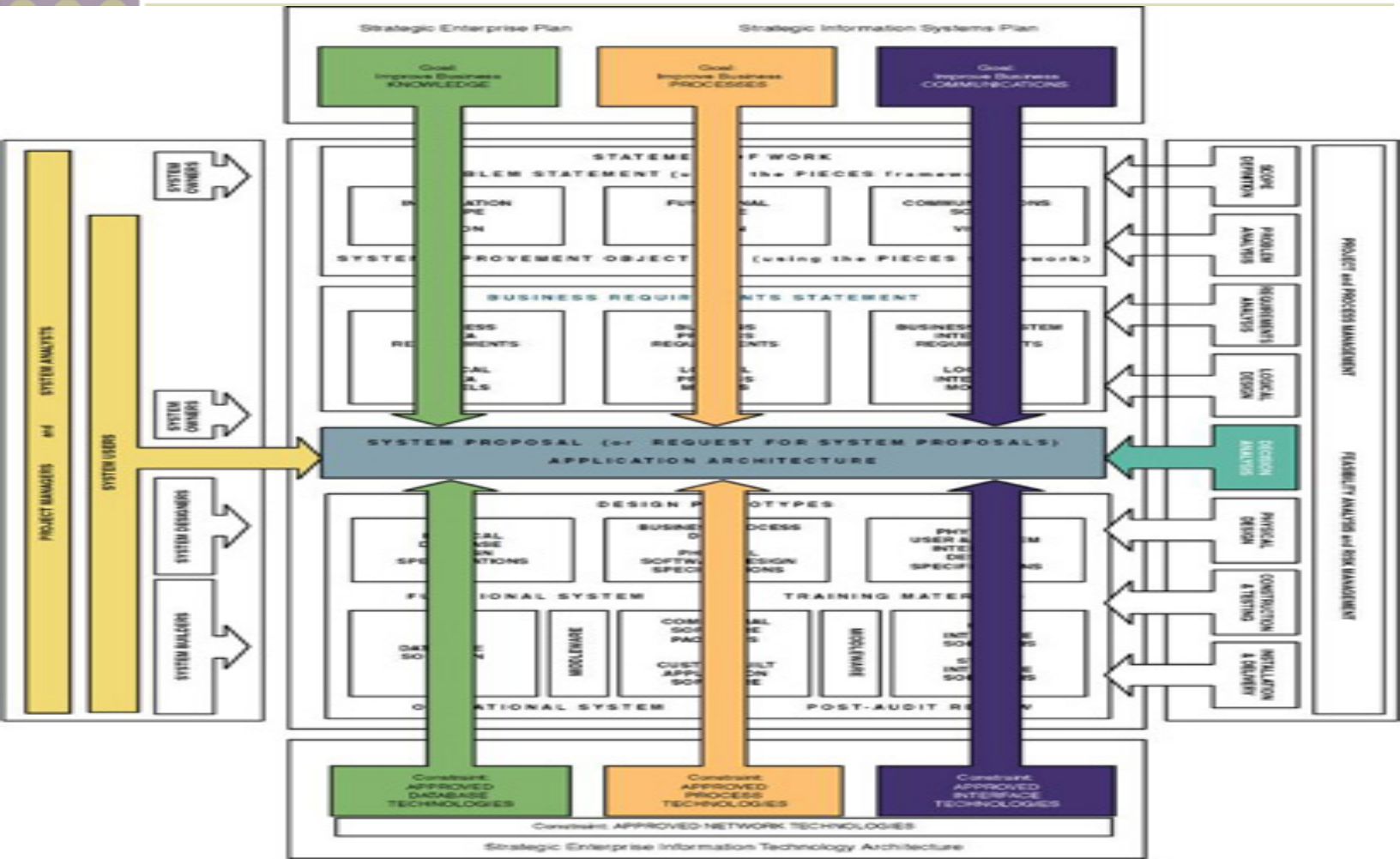
Context of Logical Design Phase of Systems Analysis



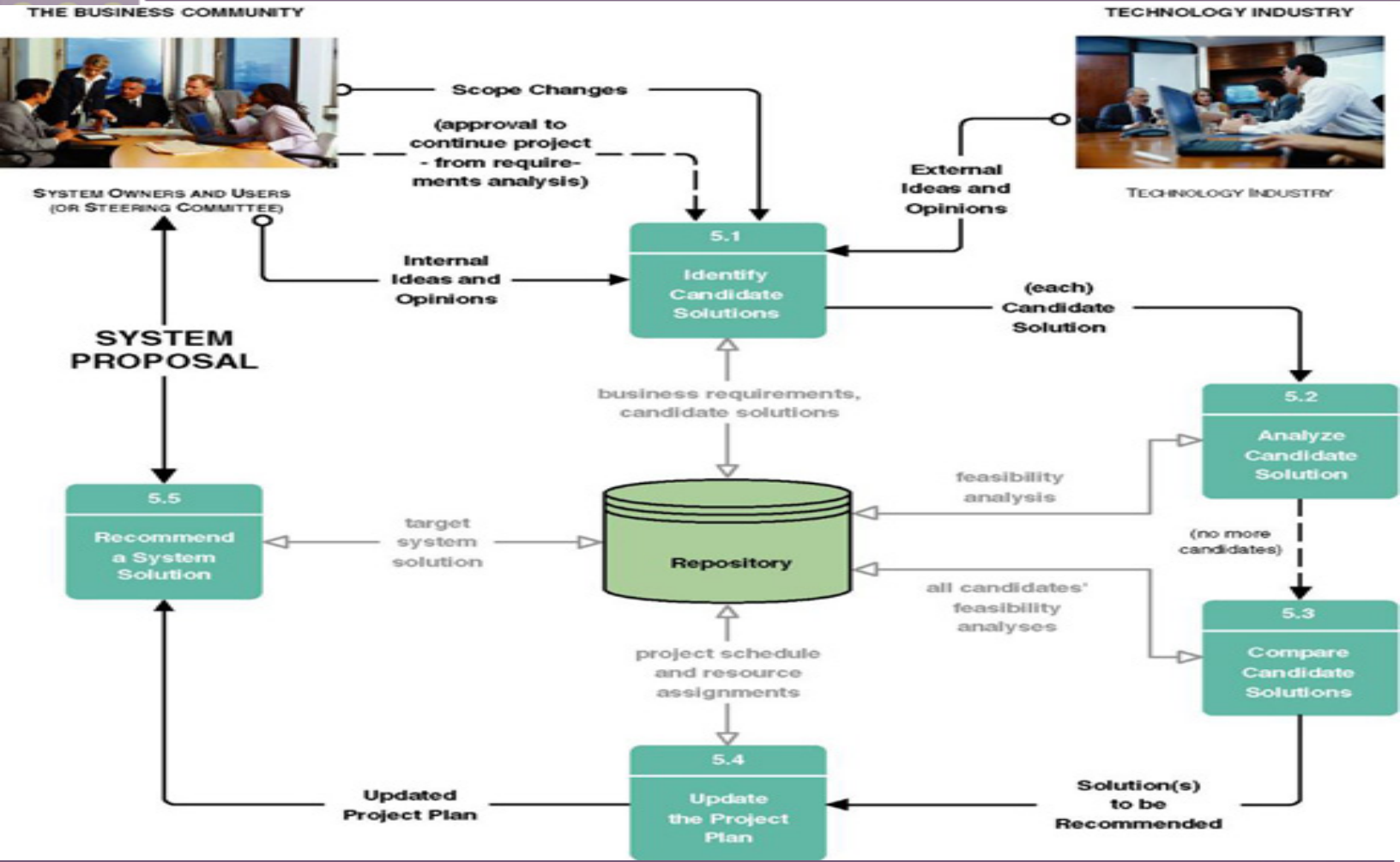
Tasks for Logical Design Phase



Context of Decision Analysis Phase



Tasks for Decision Analysis Phase



Key Terms of Decision Analysis Phase

- **Technical feasibility** – Is the solution technically practical? Does our staff have the technical expertise to design and build this solution?
- **Operational feasibility** – Will the solution fulfill the users' requirements? To what degree? How will the solution change the users' work environment? How do users feel about such a solution?
- **Economic feasibility** – Is the solution cost-effective?
- **Schedule feasibility** – Can the solution be designed and implemented within an acceptable time period?

Candidate Systems Matrix

Characteristics	Candidate 1	Candidate 2	Candidate 3	Candidate ...
Portion of System Computerized Brief description of that portion of the system that would be computerized in this candidate.	COTS package Platinum Plus from Entertainment Software Solutions would be purchased and customized to satisfy Member Services required functionality.	Member Services and warehouse operations in relation to order fulfillment.	Same as candidate 2.	
Benefits Brief description of the business benefits that would be realized for this candidate.	This solution can be implemented quickly because it's a purchased solution.	Fully supports user required business processes for SoundStage Inc. Plus more efficient interaction with member accounts.	Same as candidate 2.	
Servers and Workstations A description of the servers and workstations needed to support this candidate.	Technically architecture dictates Pentium Pro, MS Windows NT class servers and Pentium, MS Windows NT 4.0 workstations (clients).	Same as candidate 1.	Same as candidate 1.	
Software Tools Needed Software tools needed to design and build the candidate (e.g., database management system, emulators, operating systems, languages, etc.). Not generally applicable if applications software packages are to be purchased.	MS Visual C++ and MS Access for customization of package to provide report writing and integration.	MS Visual Basic 5.0 System Architect 3.1 Internet Explorer	MS Visual Basic 5.0 System Architect 3.1 Internet Explorer	
Application Software A description of the software to be purchased, built, accessed, or some combination of these techniques.	Package Solution	Custom Solution	Same as candidate 2.	

Candidate Systems Matrix (cont.)

Method of Data Processing Generally some combination of: on-line, batch, deferred batch, remote batch, and real-time.	Client/Server	Same as candidate 1.	Same as candidate 1.	
Output Devices and Implications A description of output devices that would be used, special output requirements (e.g., network, preprinted forms, etc.), and output considerations (e.g., timing constraints).	(2) HP4MV department laser printers (2) HP5SI LAN laser printers	(2) HP4MV department laser printers (2) HP5SI LAN laser printers (1) PRINTRONIX bar-code printer (includes software & drivers) Web pages must be designed to VGA resolution. All internal screens will be designed for SVGA resolution.	Same as candidate 2.	
Input Devices and Implications A description of input methods to be used, input devices (e.g., keyboard, mouse, etc.), special input requirements (e.g., new or revised forms from which data would be input), and input considerations (e.g., timing of actual inputs).	Keyboard & mouse	Apple "Quick Take" digital camera and software (15) PSC Quickscan laser bar-code scanners (1) HP Scanjet 4C Flatbed Scanner Keyboard & mouse	Same as candidate 2.	
Storage Devices and Implications Brief description of what data would be stored, what data would be accessed from existing stores, what storage media would be used, how much storage capacity would be needed, and how data would be organized.	MS SQL Server DBMS with 100GB arrayed capability.	Same as candidate 1.	Same as candidate 1.	

Feasibility Matrix

Feasibility Criteria	Weight	Candidate 1	Candidate 2	Candidate 3	Candidate . . .
Operational Feasibility **Functionality**. A description of to what degree the candidate would benefit the organization and how well the system would work. **Political**. A description of how well received this solution would be from both user management, user, and organization perspective.	30%	Only supports Member Services requirements and current business processes would have to be modified to take advantage of software functionality **Score: 60**	Fully supports user required functionality. **Score: 100**	Same as candidate 2. **Score: 100**	
Technical Feasibility **Technology**. An assessment of the maturity, availability (or ability to acquire), and desirability of the computer technology needed to support this candidate. **Expertise**. An assessment of the technical expertise needed to develop, operate, and maintain the candidate system.	30%	Current production release of Platinum Plus package is version 1.0 and has only been on the market for 6 weeks. Maturity of product is a risk and company charges an additional monthly fee for technical support. Required to hire or train C++ expertise to perform modifications for integration requirements. **Score: 50**	Although current technical staff has only Powerbuilder experience, the senior analysts who saw the MS Visual Basic demonstration and presentation have agreed the transition will be simple and finding experienced VB programmers will be easier than finding Powerbuilder programmers and at a much cheaper cost. MS Visual Basic 5.0 is a mature technology based on version number. **Score: 95**	Although current technical staff is comfortable with Powerbuilder, management is concerned with recent acquisition of Powerbuilder by Sybase Inc. MS SQL Server is a current company standard and competes with SYBASE in the Client/Server DBMS market. Because of this we have no guarantee future versions of Powerbuilder will "play well" with our current version SQL Server. **Score: 60**	
Economic Feasibility **Cost to develop:** **Payback period (discounted):** **Net present value:** **Detailed calculations:**	30%	 Approximately $350,000. Approximately 4.5 years. Approximately $210,000. See Attachment A. **Score: 60**	 Approximately $418,040. Approximately 3.5 years. Approximately $306,748. See Attachment A. **Score: 85**	 Approximately $400,000. Approximately 3.3 years. Approximately $325,500. See Attachment A. **Score: 90**	
Schedule Feasibility An assessment of how long the solution will take to design and implement.	10%	Less than 3 months. **Score: 95**	9–12 months **Score: 80**	9 months **Score: 85**	
Ranking	100%	60.5	92	83.5	

Typical System Proposal Outline

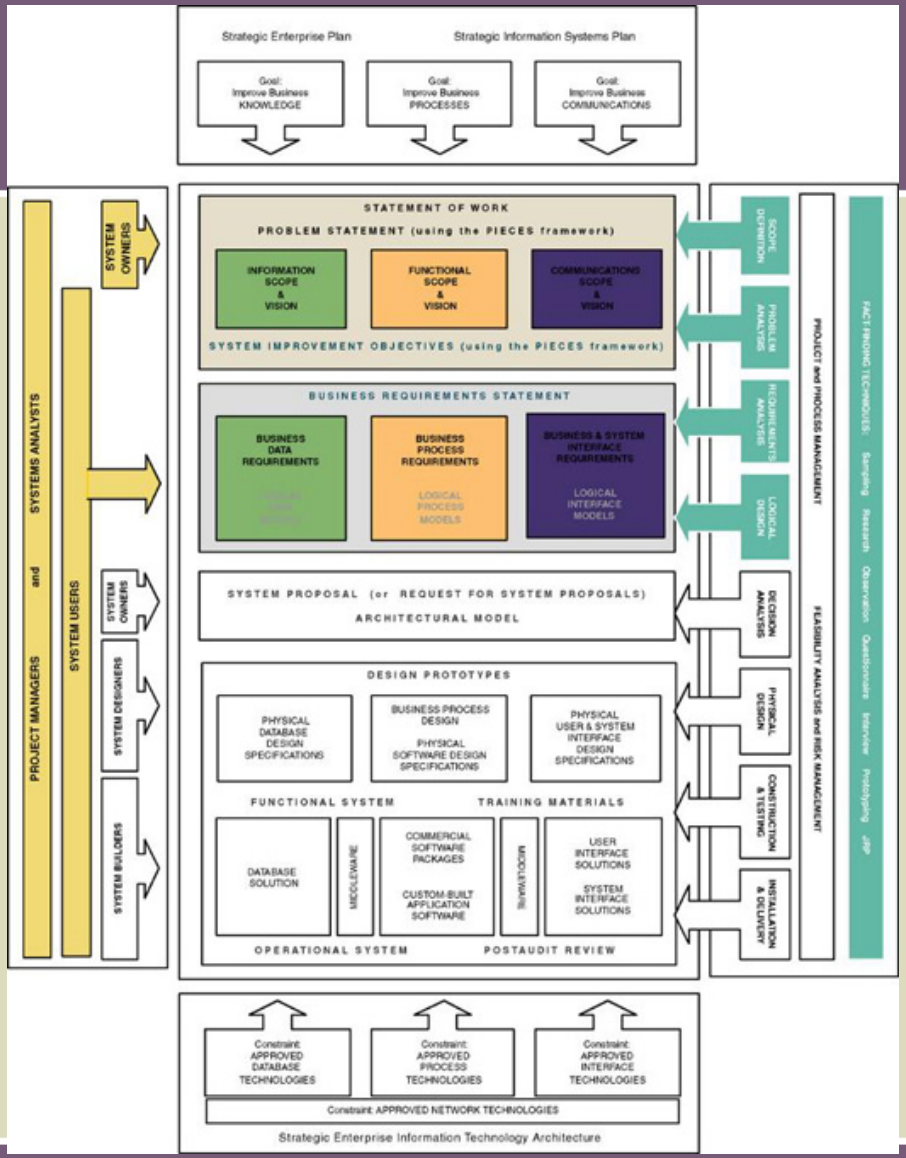
I. Introduction
A. Purpose of the report
B. Background of the project leading to this report
C. Scope of the report
D. Structure of the report
II. Tools and techniques used
A. Solution generated
B. Feasibility analysis (cost-benefit)
III. Information systems requirements
IV. Alternative solutions and feasibility analysis
V. Recommendations
VI. Appendices

Chapter 6

Fact-Finding Techniques for Requirements Discovery

Objectives

- Define system requirements and differentiate between functional and nonfunctional requirements.
- Understand the activity of problem analysis and be able to create an Ishikawa (fishbone) diagram.
- Understand the concept of requirements management.
- Identify and characterize seven fact-finding techniques.
- Understand six guidelines for effective listening.
- Understand body language and proxemics.
- Characterize the typical participants in a JRP session.
- Complete the planning process for a JRP session.
- Describe benefits of JRP as fact-finding technique.
- Describe a fact-finding strategy that will make the most of your time with end-users.


Strategic Enterprise Plan
Strategic Information Systems Plan
Goal: Improve Business KNOWLEDGE
Goal: Improve Business PROCESSES
Goal: Improve Business COMMUNICATIONS
PROJECT MANAGERS and SYSTEMS ANALYSTS
SYSTEM USERS
SYSTEM OWNERS
SYSTEM OWNERS
SYSTEM DESIGNERS
SYSTEM BUILDERS
STATEMENT OF WORK
PROBLEM STATEMENT (using the PIECES framework)
INFORMATION SCOPE & VISION
FUNCTIONAL SCOPE & VISION
COMMUNICATIONS SCOPE & VISION
SYSTEM IMPROVEMENT OBJECTIVES (using the PIECES framework)
BUSINESS REQUIREMENTS STATEMENT
BUSINESS DATA REQUIREMENTS
LOGICAL DATA MODELS
BUSINESS PROCESS REQUIREMENTS
LOGICAL PROCESS MODELS
BUSINESS & SYSTEM INTERFACE REQUIREMENTS
LOGICAL INTERFACE MODELS
SYSTEM PROPOSAL (or REQUEST FOR SYSTEM PROPOSALS)
ARCHITECTURAL MODEL
DESIGN PROTOTYPES
PHYSICAL DATABASE DESIGN SPECIFICATIONS
BUSINESS PROCESS DESIGN
PHYSICAL SOFTWARE DESIGN SPECIFICATIONS
PHYSICAL USER & SYSTEM INTERFACE DESIGN SPECIFICATIONS
FUNCTIONAL SYSTEM
TRAINING MATERIALS
DATABASE SOLUTION
MIDDLEWARE
COMMERCIAL SOFTWARE PACKAGES
CUSTOM-BUILT APPLICATION SOFTWARE
GROUPWARE
USER INTERFACE SOLUTIONS
SYSTEM INTERFACE SOLUTIONS
OPERATIONAL SYSTEM
POSTAUDIT REVIEW
SCOPE DEFINITION
PROBLEM ANALYSIS
REQUIREMENTS ANALYSIS
LOGICAL DESIGN
DECISION ANALYSIS
PHYSICAL DESIGN
CONSTRUCTION & TESTING
INSTALLATION & DELIVERY
PROJECT and PROCESS MANAGEMENT
FEASIBILITY ANALYSIS and RISK MANAGEMENT
FACT-FINDING TECHNIQUES: Sampling Research Observation Questionnaire Interview Prototyping JAD
Constraint: APPROVED DATABASE TECHNOLOGIES
Constraint: APPROVED PROCESS TECHNOLOGIES
Constraint: APPROVED INTERFACE TECHNOLOGIES
Constraint: APPROVED NETWORK TECHNOLOGIES
Strategic Enterprise Information Technology Architecture

Introduction to Requirements Discovery

Requirements discovery – the process and techniques used by systems analysts to identify or extract system problems and solution requirements from the user community.

System requirement – something that the information system must do or a property that it must have. Also called a *business requirement*.

- **Functional requirement** - something the information system must do
- **Nonfunctional requirement** - a property or quality the system must have:
  - Performance
  - Security

Non-functional Requirements

TABLE 6-1 PIECES Classification of System Requirements

Nonfunctional Requirement Type	Explanation
Performance	Performance requirements represent the performance the system is required to exhibit to meet the needs of users. • What is the acceptable throughput rate? • What is the acceptable response time?
Information	Information requirements represent the information that is pertinent to the users in terms of content, timeliness, accuracy, and format. • What are the necessary inputs and outputs? When must they happen? • What is the required data to be stored? • How current must the information be? • What are the interfaces to external systems?
Economy	Economy requirements represent the need for the system to reduce costs or increase profits. • What are the areas of the system where costs must be reduced? • How much should costs be reduced or profits be increased? • What are the budgetary limits? • What is the timetable for development?
Control (and security)	Control requirements represent the environment in which the system must operate, as well as the type and degree of security that must be provided. • Must access to the system or information be controlled? • What are the privacy requirements? • Does the criticality of the data necessitate the need for special handling (backups, off-site storage, etc.) of the data?
Efficiency	Efficiency requirements represent the system's ability to produce outputs with minimal waste. • Are there duplicate steps in the process that must be eliminated? • Are there ways to reduce waste in the way the system uses it resources?
Service	Service requirements represent needs in order for the system to be reliable, flexible, and expandable. • Who will use the system, and where are they located? • Will there be different types of users? • What are the appropriate human factors? • What training devices and training materials are to be included in the system? • What training devices and training materials are to be developed and maintained separately from the system, such as stand-alone computer-based training (CBT) programs or databases? • What are the reliability/availability requirements? • How should the system be packaged and distributed? • What documentation is required?

Results of Incorrect Requirements

- The system may cost more than projected.
- The system may be delivered later than promised.
- The system may not meet the users' expectations and they may not to use it.
- Once in production, costs of maintaining and enhancing system may be excessively high.
- The system may be unreliable and prone to errors and downtime.
- Reputation of IT staff is tarnished as failure will be perceived as a mistake by the team.

Phase in Which Error Discovered	Cost Ratio
Requirements	1
Design	3–6
Coding	10
Development Testing	15–40
Acceptance Testing	30–70
Operation	40–1000

Criteria for System Requirements

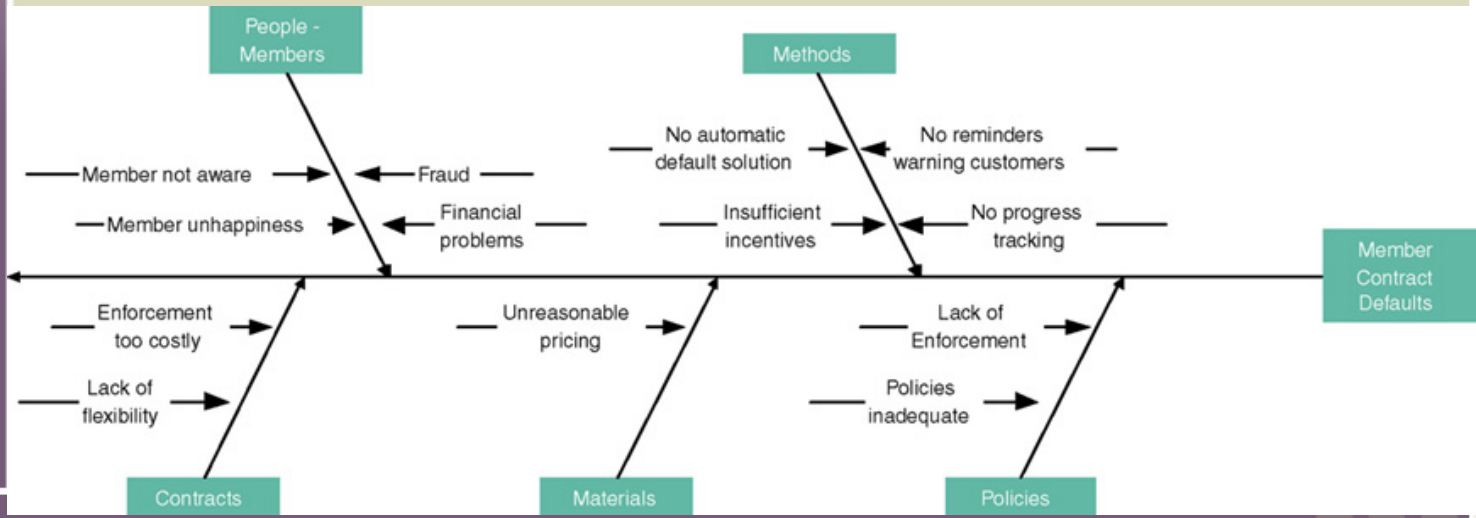
- **Consistent** – not conflicting or ambiguous.
- **Complete** – describe all possible system inputs and responses.
- **Feasible** – can be satisfied based on the available resources and constraints.
- **Required** – truly needed and fulfill the purpose of the system.
- **Accurate** – stated correctly.
- **Traceable** – directly map to functions and features of system.
- **Verifiable** – defined so can be demonstrated during testing.

Process of Requirement Discovery

1. Problem discovery and analysis
2. Requirements discovery
3. Documenting and analyzing requirements
4. Requirements management

1. Problem discovery and analysis (Ishikawa Diagram)

- Graphical tool used to identify, explore, and depict problems and the causes and effects of those problems. It is often referred to as a cause-and-effect diagram or a fishbone diagram.
  - Problem at right (fish head)
  - Possible causes drawn as "bones" off main backbone
  - Brainstorm for 3-6 main categories of possible causes



2. Requirements Discovery

- Given an understand of problems, the systems analyst can start to define requirements.

Fact-finding – the formal process of using research, meetings, interviews, questionnaires, sampling, and other techniques to collect information about system problems, requirements, and preferences. It is also called *information gathering* or *data collection*.

3. Documenting and Analyzing Req

- Documenting the draft requirements
  - Use cases
  - Decision tables
  - Requirements tables
- Analyzing requirements to resolve problems
  - Missing requirements
  - Conflicting requirements
  - Infeasible requirements
  - Overlapping requirements
  - Ambiguous requirements
- Formalizing requirements
  - Requirements definition document
  - Communicated to stakeholders or steering body

3. Documenting and Analyzing Req (Requirements Definition Document)

Requirements Definition Document – A formal document that communicates the requirements of a proposed system to key stakeholders and serves as a contract for the systems project.

- Synonyms
  - Requirements definition report
  - Requirements statement
  - Requirements specification
  - Functional specifications

Sample Requirements Definition Report Outline

REQUIREMENTS DEFINITION REPORT

1. Introduction
    - 1.1. Purpose
    - 1.2. Background
    - 1.3. Scope
    - 1.4. Definitions, Acronyms, and Abbreviations
    - 1.5. References
2. General Project Description
    - 2.1. Functional Requirements
3. Requirements and Constraints
    - 3.1. Functional Requirements
    - 3.2. Nonfunctional Requirements
4. Conclusion
    - 4.1. Outstanding Issues

Appendix (optional)

4. Requirements Management

Requirements management - the process of managing change to the requirements.

- Over the lifetime of the project it is very common for new requirements to emerge and existing requirements to change.
- Studies have shown that over the life of a project as much as 50 percent or more of the requirements will change before the system is put into production.

Fact-Finding Ethics

- Fact-Finding often brings systems analysts into contact with sensitive information.
  - Company plans
  - Employee salaries or medical history
  - Customer credit card, social security, or other information
- Ethical behavior
  - Systems analysts must not misuse information.
  - Systems analysts must protect information from people who would misuse it.
- Otherwise
  - Systems analyst loses respect, credibility, and confidence of users and management, impairing ability to do job
  - Organization and systems analyst could have legal liability
  - Systems analyst could lose job

Seven Fact-Finding Methods

1. Sampling of existing documentation, forms, and databases.
2. Research and site visits (CA).
3. Observation of the work environment.
4. Questionnaires.
5. Interviews.
6. Prototyping.
7. Joint requirements planning (JRP).

1. Sampling

Sampling –process of collecting a representative sample of documents, forms, and records. E.g. Causes of problems, Persons, Business Functions, Data, Questions of interviews.

- Organization chart
- Memos and other documents that describe the problem
- Standard operating procedures for current system
- Completed forms
- Manual and computerized screens and reports
- Samples of databases
- Flowcharts and other system documentation

Determining Sample Size for Forms

- Sample Size = 0.25 x (Certainty factor/Acceptable error) 2
- Sample Size = 0.25(1.645/0.10) 2 = 68
- Sample Size =0.10(1 – 0.10)(1.645/0.10)2 = 25

Or if analyst knows 1 in 10 varies from norm.

Certainty factor from certainty table. 10% acceptable error.

Desired Certainty	Certainty Factor
95%	1.960
90	1.645
80	1.281

Sampling Techniques

Randomization – a sampling technique characterized by having no predetermined pattern or plan for selecting sample data.

Stratification – a systematic sampling technique that attempts to reduce the variance of the estimates by spreading out the sampling—for example, choosing documents or records by formula—and by avoiding very high or low estimates.

3. Observation

Observation – a fact-finding technique wherein the systems analyst either participates in or watches a person perform activities to learn about the system.

Advantages?

Disadvantages?

Work sampling - a fact-finding technique that involves a large number of observations taken at random intervals.

Observation

Advantages

- Data gathered can be very reliable
- Can see exactly what is being done in complex tasks
- Relatively inexpensive compared with other techniques
- Can do work measurements

Disadvantages

- People may perform differently when being observed
- Work observed may not be representative of normal conditions
- Timing can be inconvenient
- Interruptions
- Some tasks not always performed the same way
- May observe wrong way of doing things

Observation Guidelines

- Determine the who, what, where, when, why, and how of the observation.
- Obtain permission from appropriate supervisors.
- Inform those who will be observed of the purpose of the observation.
- Keep a low profile.
- Take notes.
- Review observation notes with appropriate individuals.
- Don't interrupt the individuals at work.
- Don't focus heavily on trivial activities.
- Don't make assumptions.

4. Questionnaires

Questionnaire – a special-purpose document that allows the analyst to collect information and opinions from respondents.

Free-format questionnaire – a questionnaire designed to offer the respondent greater latitude in the answer. A question is asked, and the respondent records the answer in the space provided after the question.

Fixed-format questionnaire – a questionnaire containing questions that require selecting an answer from predefined available responses.

Questionnaires

Advantages

- Often can be answered quickly
- People can complete at their convenience
- Relatively inexpensive way to gather data from a large number
- Allow for anonymity
- Responses can be tabulated quickly

Disadvantages

- Return rate is often low
- No guarantee that an individual will answer all questions
- No opportunity to reword or explain misunderstood questions
- Cannot observe body language
- Difficult to prepare

Types of Fixed-Format Questions

- Multiple-choice questions
- Rating questions
- Ranking questions

Rank the following transactions according to the amount of time you spend processing them.

___ % new customer orders
___ % order cancellations
___ % order modifications
___ % payments

The implementation of quality discounts would cause an increase in customer orders.

___ Strongly agree
___ Agree
___ No opinion
___ Disagree
___ Strongly disagree

Is the current accounts receivable report that you receive useful?

___ Yes
___ No

Developing a Questionnaire

1. Determine what facts and opinions must be collected and from whom you should get them.
2. Based on the facts and opinions sought, determine whether free- or fixed-format questions will produce the best answers.
3. Write the questions.
4. Test the questions on a small sample of respondents.
5. Duplicate and distribute the questionnaire.

5. Interviews

Interview - a fact-finding technique whereby the systems analysts collect information from individuals through face-to-face interaction.

- Find facts
- Verify facts
- Clarify facts
- Generate enthusiasm
- Get the end-user involved
- Identify requirements
- Solicit ideas and opinions

The personal interview is generally recognized as the most important and most often used fact-finding technique.

Types of Interviews and Questions

Unstructured interview –conducted with only a general goal or subject in mind and with few, if any, specific questions. The interviewer counts on the interviewee to provide a framework and direct the conversation.

Structured interview –interviewer has a specific set of questions to ask of the interviewee.

Open-ended question – question that allows the interviewee to respond in any way.

Closed-ended question – a question that restricts answers to either specific choices or short, direct responses.

Interviews

Advantages

- Give analyst opportunity to motivate interviewee to respond freely and openly
- Allow analyst to probe for more feedback
- Permit analyst to adapt or reword questions for each individual
- Can observe nonverbal communication

Disadvantages

- Time-consuming
- Success highly dependent on analyst's human relations skills
- May be impractical due to location of interviewees

Procedure to Conduct an Interview

1. Select Interviewees
   - End users
   - Learn about individual prior to the interview
2. Prepare for the Interview
   - interview guide
3. Conduct the Interview
   - Summarize the problem
   - Offer an incentive for participation
   - Ask the interviewee for assistance
4. Follow Up on the Interview
   - Memo that summarizes the interview

Sample Interview Guide

Interviewee:	Jeff Bentley, Accounts Receivable Manager
Date:	January 19, 2003
Time:	1:30 P.M.
Place:	Room 223, Admin. Bldg.
Subject:	Current Credit-Checking Policy

Time Allocated	Interviewer Question or Objective	Interviewee Response
1 to 2 min.	**Objective** Open the interview: • Introduce ourselves • Thank Mr. Bentley for his valuable time. • State the purpose of the interview — to obtain an understanding of the existing credit-checking policies.	
5 min.	**Question 1** What conditions determine whether a customer's order is approved for credit? **Follow-up**	
5 min.	**Question 2** What are the possible decisions or actions that might be taken once these conditions have been evaluated? **Follow-up**	
3 min.	**Question 3** How are customers notified when credit is not approved for their order? **Follow-up**	

(continued)

Sample Interview Guide (concluded)

1 min.	**Question 4** After a new order is approved for credit and placed in the file containing orders that can be filled, a customer might request that a modification be made to the order. Would the order have to go through credit approval again if the new total order cost exceeds the original cost? **Follow-up**	
1 min.	**Question 5** Who are the individuals who perform the credit checks? **Follow-up**	
1 to 3 min.	**Question 6** May I have permission to talk to those individuals to learn specifically how they carry out the credit-checking process? **Follow-up** If so: When would be an appropriate time to meet with each of them?	
1 min.	**Objective** Conclude the interview: • Thank Mr. Bentlry for his cooperation and assure him that he will be receiving a copy of what transpired during the interview.	
21 minutes	Time allotted for questions and objectives	
9 minutes	Time allotted for follow-up questions and redirection	
30 minutes	Time allotted for interview (1:30 p.m. - 2:00 p.m.)	
General Comments and Notes:		

Prepare for the Interview

- Types of Questions to Avoid
  - Loaded questions
  - Leading questions
  - Biased questions
- Interview Question Guidelines
  - Use clear and concise language.
  - Don’t include your opinion as part of the question.
  - Avoid long or complex questions.
  - Avoid threatening questions.
  - Don’t use “you” when you mean a group of people.

Conduct the Interview

- Dress to match interviewee
- Arrive on time
  - Or early if need to confirm room setup
- Open interview by thanking interviewee
- State purpose and length of interview and how data will be used
- Monitor the time
- Ask follow-up questions
  - Probe until you understand
  - Ask about exception conditions ("what if...")

Interviewing Do's and Don'ts

Do

- Dress appropriately
- Be courteous
- Listen carefully
- Maintain control of the interview
- Probe
- Observe mannerisms and nonverbal communication
- Be patient
- Keep interviewee at ease
- Maintain self-control
- Finish on time

Don't

- Assume an answer is finished or leading nowhere
- Reveal verbal and nonverbal clues
- Use jargon
- Reveal personal biases
- Talk more than listen
- Assume anything about the topic or the interviewee
- Tape record (take notes instead)

Body Language and Proxemics

Body language – the nonverbal information we communicate.

- Facial disclosure
- Eye contact
- Posture

Proxemics – the relationship between people and the space around them.

- Intimate zone—closer than 1.5 feet
- Personal zone—from 1.5 feet to 4 feet
- Social zone—from 4 feet to 12 feet
- Public zone—beyond 12 feet

6. Discovery Prototyping

Discovery prototyping – the act of building a small-scale, representative or working model of the users' requirements in order to discover or verify those requirements.

Discovery Prototyping

Advantages

- Can experiment to develop understanding of how system might work
- Aids in determining feasibility and usefulness of system before development
- Serves as training mechanism
- Aids in building test plans and scenarios
- May minimize time spent on fact-finding

Disadvantages

- Developers may need to be trained in prototyping
- Users may develop unrealistic expectations
- Could extend development schedule

7. Joint Requirements Planning

Joint requirements planning (JRP) – a process whereby highly structured group meetings are conducted for the purpose of analyzing problems and defining requirements.

- JRP is a subset of a more comprehensive joint application development or JAD technique that encompasses the entire systems development process.

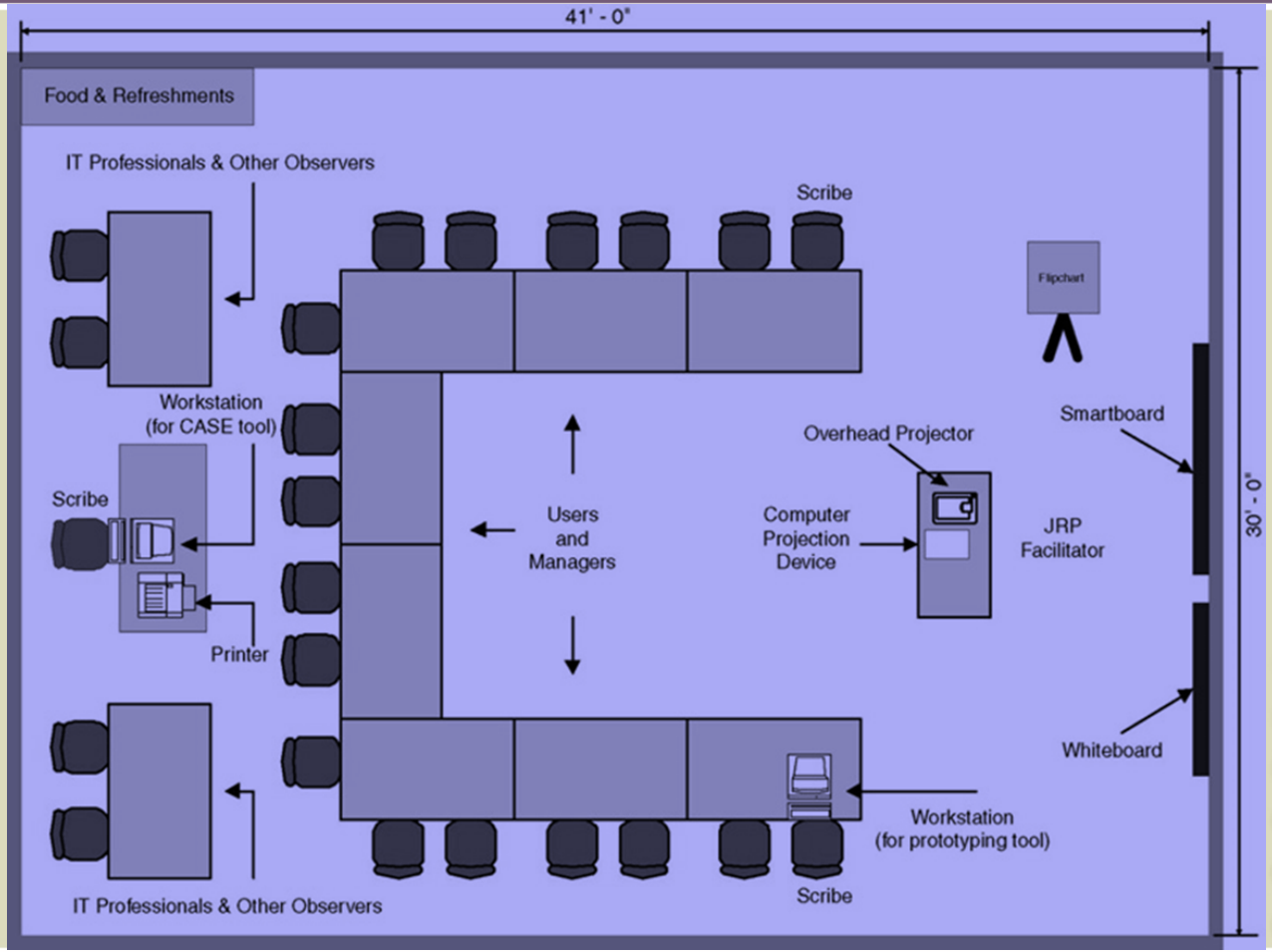
JRP Participants

- Sponsor
- Facilitator
- Users and Managers
- Scribes
- IT Staff

Steps to Plan a JRP Session

1. Selecting a location
   - Away from workplace when possible
   - Requires several rooms
   - Equipped with tables, chairs, whiteboard, overhead projectors
   - Needed computer equipment
2. Selecting the participants
   - Each needs release from regular duties
3. Preparing the agenda
   - Briefing documentation
   - Agenda distributed before each session

Typical Room Layout for JRP session



Guidelines for Conducting a JRP Session

- Do not unreasonably deviate from the agenda
- Stay on schedule
- Ensure that the scribe is able to take notes
- Avoid the use of technical jargon
- Apply conflict resolution skills
- Allow for ample breaks
- Encourage group consensus
- Encourage user and management participation without allowing individuals to dominate the session
- Make sure that attendees abide by the established ground rules for the session

Brainstorming

- Sometimes, one of the goals of a JRP session is to generate possible ideas to solve a problem.
  - Brainstorming is a common approach that is used for this purpose.

Brainstorming – a technique for generating ideas by encouraging participants to offer as many ideas as possible in a short period of time without any analysis until all the ideas have been exhausted.

Brainstorming Guidelines

- Isolate appropriate people in a place that free from distractions and interruptions.
- Make sure everyone understands purpose of the meeting.
- Appoint one person to record ideas.
- Remind everyone of brainstorming rules.
- Within a specified time period, team members call out their ideas as quickly as they can think of them.
- After group has run out of ideas and all ideas have been recorded, then and only then should ideas be evaluated.
- Refine, combine, and improve ideas generated earlier.

Benefits of JRP

- JRP actively involves users and management in the development project (encouraging them to take “ownership” in the project).
- JRP reduces the amount of time required to develop systems.
- When JRP incorporates prototyping as a means for confirming requirements and obtaining design approvals, the benefits of prototyping are realized

A Fact-Finding Strategy

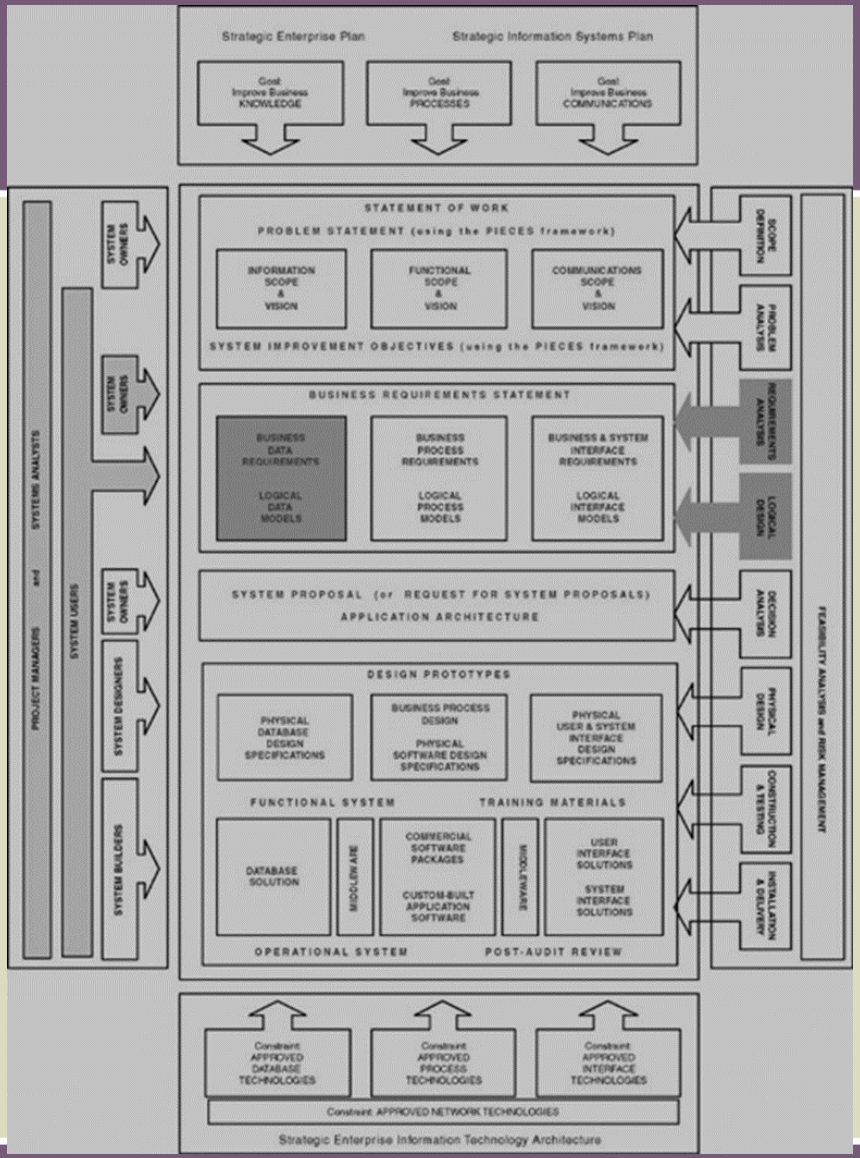
1. Learn from existing documents, forms, reports, and files.
2. If appropriate, observe the system in action.
3. Given all the facts that already collected, design and distribute questionnaires to clear up things that aren't fully understood.
4. Conduct interviews (or group work sessions).
5. (Optional). Build discovery prototypes for any functional requirements that are not understood or for requirements that need to be validated.
6. Follow up to verify facts.

Chapter 8

Data Modeling and Analysis

Objectives

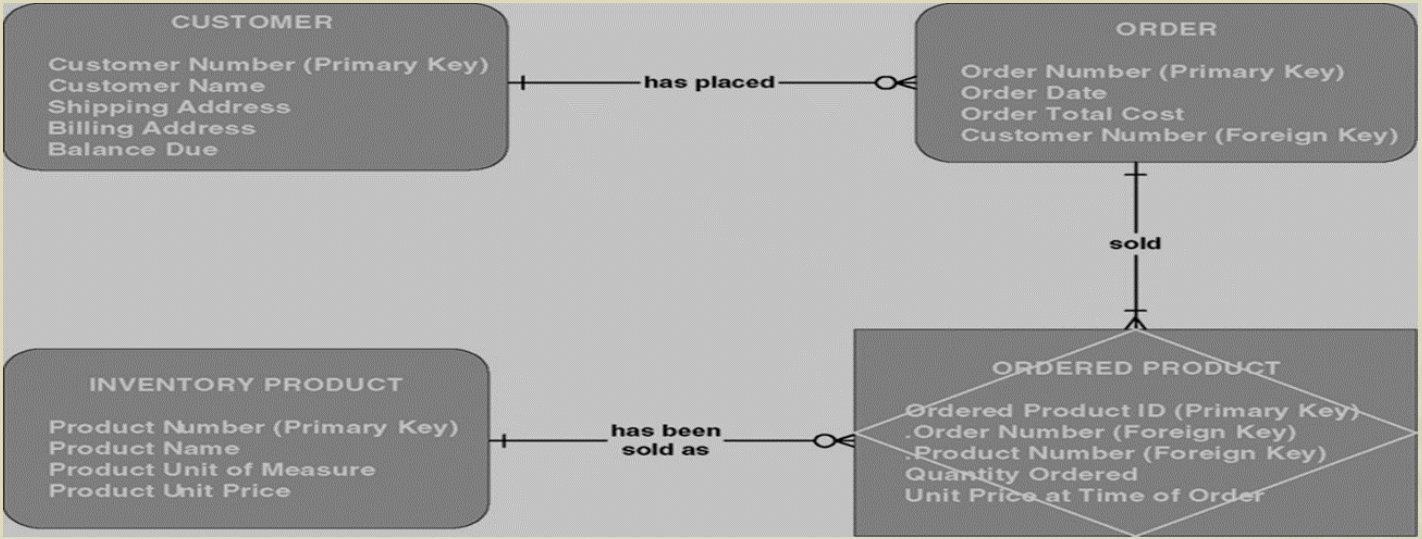
- Define data modeling and explain its benefits.
- Recognize and understand the basic concepts and constructs of a data model.
- Read and interpret an entity relationship data model.
- Explain when data models are constructed during a project and where the models are stored.
- Discover entities and relationships.
- Construct an entity-relationship context diagram.
- Discover or invent keys for entities and construct a key-based diagram.
- Construct a fully attributed entity relationship diagram and describe data structures and attributes to the repository.
- Normalize a logical data model to remove impurities that can make a database unstable, inflexible, and nonscalable.
- Describe a useful tool for mapping data requirements to business operating locations.


Strategic Enterprise Plan
Strategic Information Systems Plan
Goal
Improve Business
KNOWLEDGE
Goal
Improve Business
PROCESSES
Goal
Improve Business
COMMUNICATIONS
PROJECT MANAGERS and SYSTEMS ANALYSTS
USER TEAMS
SYSTEM OWNERS
SYSTEM OWNERS
SYSTEM OWNERS
SYSTEM DESIGNERS
SYSTEM BUILDERS
STATEMENT OF WORK
PROBLEM STATEMENT (using the PIECES framework)
INFORMATION SCOPE & VISION
FUNCTIONAL SCOPE & VISION
COMMUNICATIONS SCOPE & VISION
SYSTEM IMPROVEMENT OBJECTIVES (using the PIECES framework)
BUSINESS REQUIREMENTS STATEMENT
BUSINESS DATA REQUIREMENTS
LOGICAL DATA MODELS
BUSINESS PROCESS REQUIREMENTS
LOGICAL PROCESS MODELS
BUSINESS & SYSTEM INTERFACE REQUIREMENTS
LOGICAL INTERFACE MODELS
SYSTEM PROPOSAL (or REQUEST FOR SYSTEM PROPOSALS)
APPLICATION ARCHITECTURE
DESIGN PROTOTYPES
PHYSICAL DATABASE DESIGN SPECIFICATIONS
BUSINESS PROCESS DESIGN
PHYSICAL SOFTWARE DESIGN SPECIFICATIONS
PHYSICAL USER & SYSTEM INTERFACE DESIGN SPECIFICATIONS
FUNCTIONAL SYSTEM
TRAINING MATERIALS
DATABASE SOLUTION
MODULAR AXE
COMMERCIAL SOFTWARE PACKAGES
CUSTOM-BUILT APPLICATION SOFTWARE
MIDDLEWARE
USER INTERFACE SOLUTIONS
SYSTEM INTERFACE SOLUTIONS
OPERATIONAL SYSTEM
POST-AUDIT REVIEW
SCOPE DEFINITION
PROBLEM ANALYSIS
REQUIREMENTS ANALYSIS
LOGICAL DESIGN
DECISION ANALYSIS
PHYSICAL DESIGN
CONSTRUCTION & TESTING
INSTALLATION & DELIVERY
FEASIBILITY ANALYSIS and RISK MANAGEMENT
Constraint
APPROVED DATABASE TECHNOLOGIES
Constraint
APPROVED PROCESS TECHNOLOGIES
Constraint
APPROVED INTERFACE TECHNOLOGIES
Constraint APPROVED NETWORK TECHNOLOGIES
Strategic Enterprise Information Technology Architecture

Data Modeling

- Data modeling – a technique for organizing and documenting a system's data. Sometimes called *database modeling*.

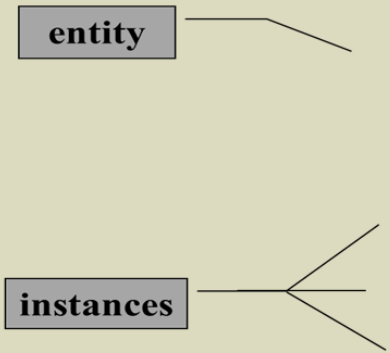
- Entity relationship diagram (ERD) – a data model utilizing several notations to depict data in terms of the entities and relationships described by that data.



Data Modeling Concepts:
1. Entity

Entity – a class of persons, places, objects, events, or concepts about which we need to capture and store data.

- **Persons**: Agency, contractor, customer, department, division, employee, instructor, student, supplier.
- **Places**: Sales region, building, room, branch office, campus.
- **Objects**: Book, machine, part, product, raw material, software license, software package, tool, vehicle model, vehicle.
- **Events**: Application, award, cancellation, class, flight, invoice, order, registration, renewal, requisition, reservation, sale, trip.
- **Concepts**: Account, block of time, bond, course, fund, qualification, stock.



Student ID	Last Name	First Name
2144	Arnold	Betty
3122	Taylor	John
3843	Simmons	Lisa
9844	Macy	Bill
2837	Leath	Heather
2293	Wrench	Tim

Data Modeling Concepts:
2. Attributes

Attribute – a descriptive property or characteristic of an entity. Synonyms include *element*, *property*, and *field*.

- Just as a physical student can have attributes, such as hair color, height, etc., data entity has data attributes

Compound attribute – an attribute that consists of other attributes. Synonyms in different data modeling languages are numerous: concatenated attribute, composite attribute, and data structure.

Data Modeling Concepts:
3. Data Type

Data type – a property of an attribute that identifies what type of data can be stored in that attribute.

Representative Logical Data Types for Attributes	
Data Type	**Logical Business Meaning**
NUMBER	Any number, real or integer.
TEXT	A string of characters, inclusive of numbers. When numbers are included in a TEXT attribute, it means that we do not expect to perform arithmetic or comparisons with those numbers.
MEMO	Same as TEXT but of an indeterminate size. Some business systems require the ability to attach potentially lengthy notes to a give database record.
DATE	Any date in any format.
TIME	Any time in any format.
YES/NO	An attribute that can assume only one of these two values.
VALUE SET	A finite set of values. In most cases, a coding scheme would be established (e.g., FR=Freshman, SO=Sophomore, JR=Junior, SR=Senior).
IMAGE	Any picture or image.

Data Modeling Concepts:
4. Domains

Domain – a property of an attribute that defines what values an attribute can legitimately take on.

Representative Logical Domains for Logical Data Types

Data Type	Domain	Examples
NUMBER	For integers, specify the range. For real numbers, specify the range and precision.	{10-99} {1.000-799.999}
TEXT	Maximum size of attribute. Actual values usually infinite; however, users may specify certain narrative restrictions.	Text(30)
DATE	Variation on the MMDDYYYY format.	MMDDYYYY MMYYYY
TIME	For AM/PM times: HHMMT For military (24-hour times): HHMM	HHMMT HHMM
YES/NO	{YES, NO}	{YES, NO} {ON, OFF}
VALUE SET	{value#1, value#2,…value#n} {table of codes and meanings}	{M=Male F=Female}

Data Modeling Concepts:
5. Default Value

Default value – the value that will be recorded if a value is not specified by the user.

Permissible Default Values for Attributes		
Default Value	**Interpretation**	**Examples**
A legal value from the domain	For an instance of the attribute, if the user does not specify a value, then use this value.	0 1.00
NONE or NULL	For an instance of the attribute, if the user does not specify a value, then leave it blank.	NONE NULL
Required or NOT NULL	For an instance of the attribute, require that the user enter a legal value from the domain. (This is used when no value in the domain is common enough to be a default but some value must be entered.)	REQUIRED NOT NULL

Data Modeling Concepts:
6. Identification

Key – an attribute, or a group of attributes, that assumes a unique value for each entity instance. It is sometimes called an *identifier*.

- **Concatenated key** - group of attributes that uniquely identifies an instance. Synonyms: composite key, compound key.
- **Candidate key** – one of a number of keys that may serve as the primary key. Synonym: *candidate identifier*.
- **Primary key** – a candidate key used to uniquely identify a single entity instance.
- **Alternate key** – a candidate key not selected to become the primary key. Synonym: secondary key.

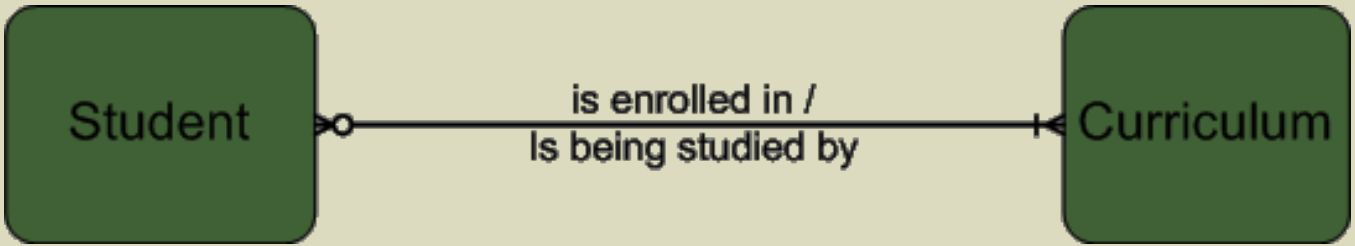
Data Modeling Concepts:
7. Subsetting Criteria

Subsetting criteria – an attribute(s) whose finite values divide all entity instances into useful subsets. Sometimes called an inversion entry.

Data Modeling Concepts:
8. Relationships

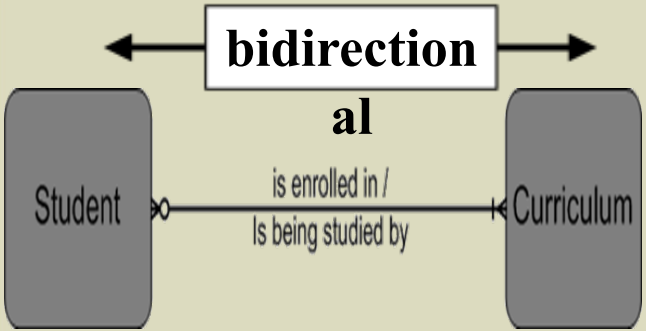
Relationship – a natural business association that exists between one or more entities.

The relationship may represent an event that links the entities or merely a logical affinity that exists between the entities.



Data Modeling Concepts:
9. Cardinality

Cardinality – the minimum and maximum number of occurrences of one entity that may be related to a single occurrence of the other entity.

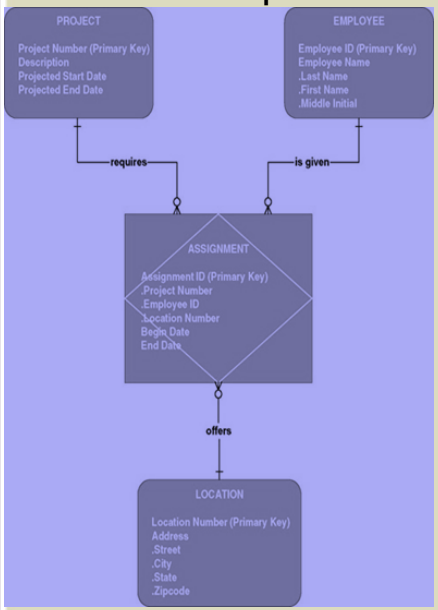


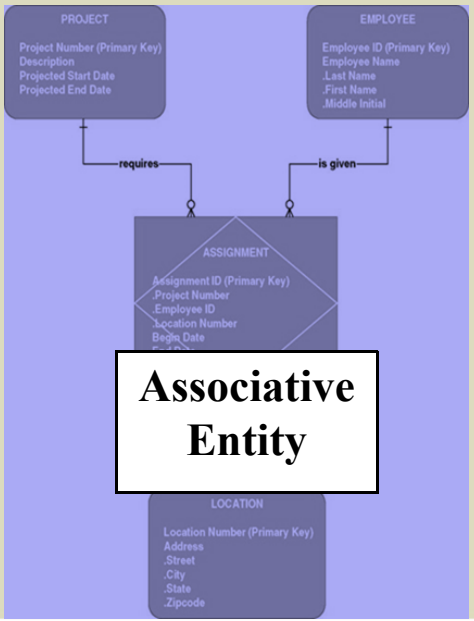
Cardinality Interpretation	Minimum Instances	Maximum Instances	Graphic Notation
Exactly one (one and only one)	1	1	– or –
Zero or one	0	1	
One or more	1	many (>1)	
Zero, one, or more	0	many (>1)	
More than one	>1	>1	

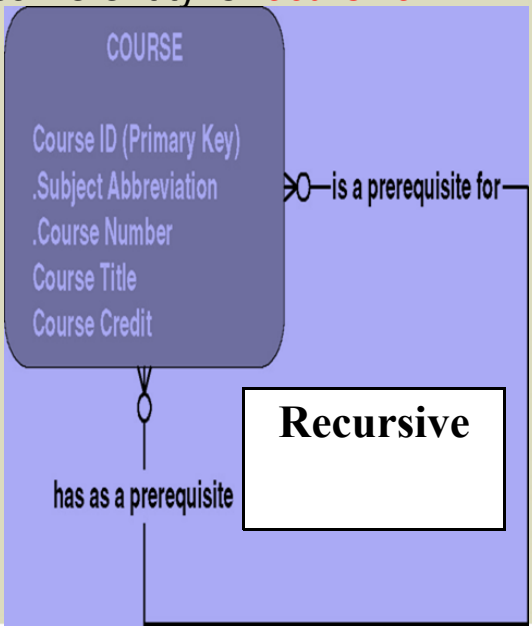
Data Modeling Concepts:
10. Degree

Degree – the number of entities that participate in the relationship.

- A relationship between two entities is called a *binary*.
- A relationship between three entities is called a *3-ary* or *ternary*.
- A relationship between different instances of the same entity is *recursive*.







Data Modeling Concepts:
11. Foreign Keys

Foreign key – a primary key of an entity that is used in another entity to identify instances of a relationship.

- A foreign key is a primary key of one entity that is contributed to (duplicated in) another entity to identify instances of a relationship.
- A foreign key always matches the primary key in the another entity
- A foreign key may or may not be unique (generally not)
- The entity with the foreign key is called the child.
- The entity with the matching primary key is called the parent.

Primary Key

Student ID	Last Name	First Name	Dorm
2144	Arnold	Betty	Smith
3122	Taylor	John	Jones
3843	Simmons	Lisa	Smith
9844	Macy	Bill	
2837	Leath	Heather	Smith
2293	Wrench	Tim	Jones

Primary Key

Dorm	Residence Director
Smith	Andrea Fernandez
Jones	Daniel Abidjan

Foreign Key Duplicated from primary key of Dorm entity (not unique in Student entity)

Data Modeling Concepts:
12. Parent and Child Entities

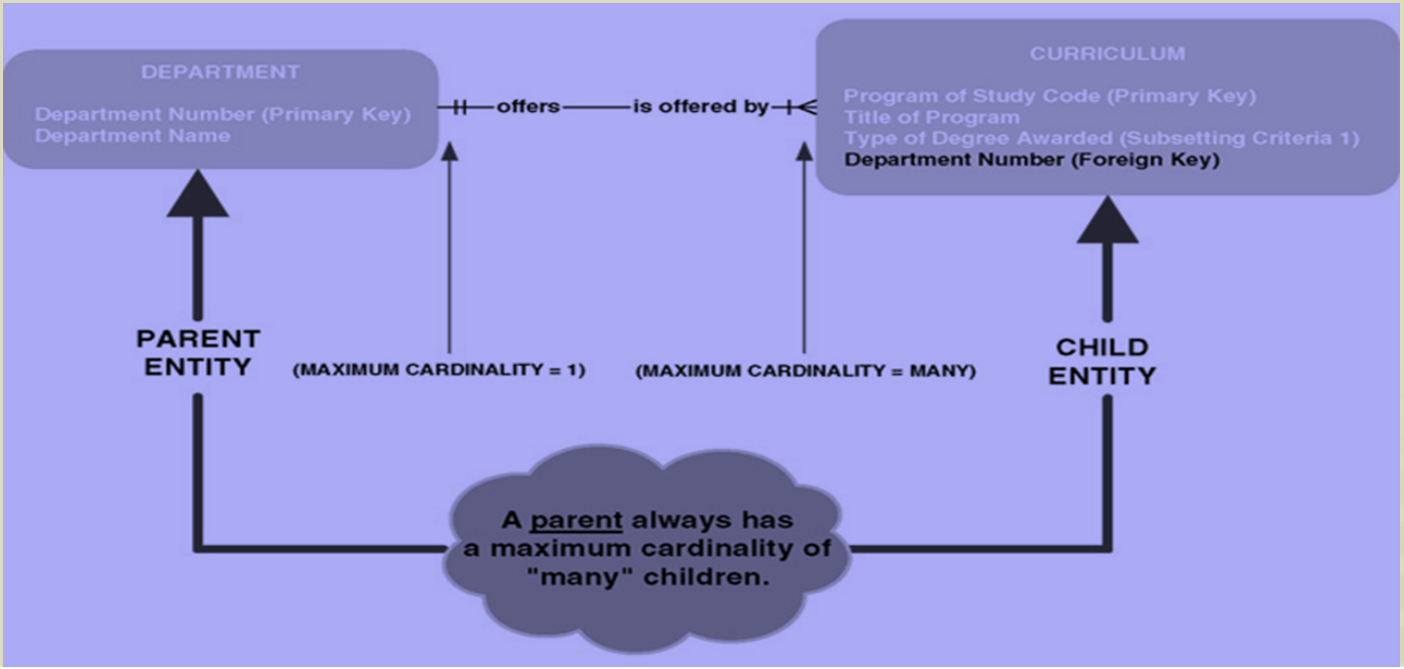
Parent entity - a data entity that contributes one or more attributes to another entity, called the child. In a one-to-many relationship the parent is the entity on the "one" side.

Child entity - a data entity that derives one or more attributes from another entity, called the parent. In a one-to-many relationship the child is the entity on the "many" side.

Data Modeling Concepts:
13. Nonidentifying Relationships

Nonidentifying relationship – relationship where each participating entity has its own independent primary key

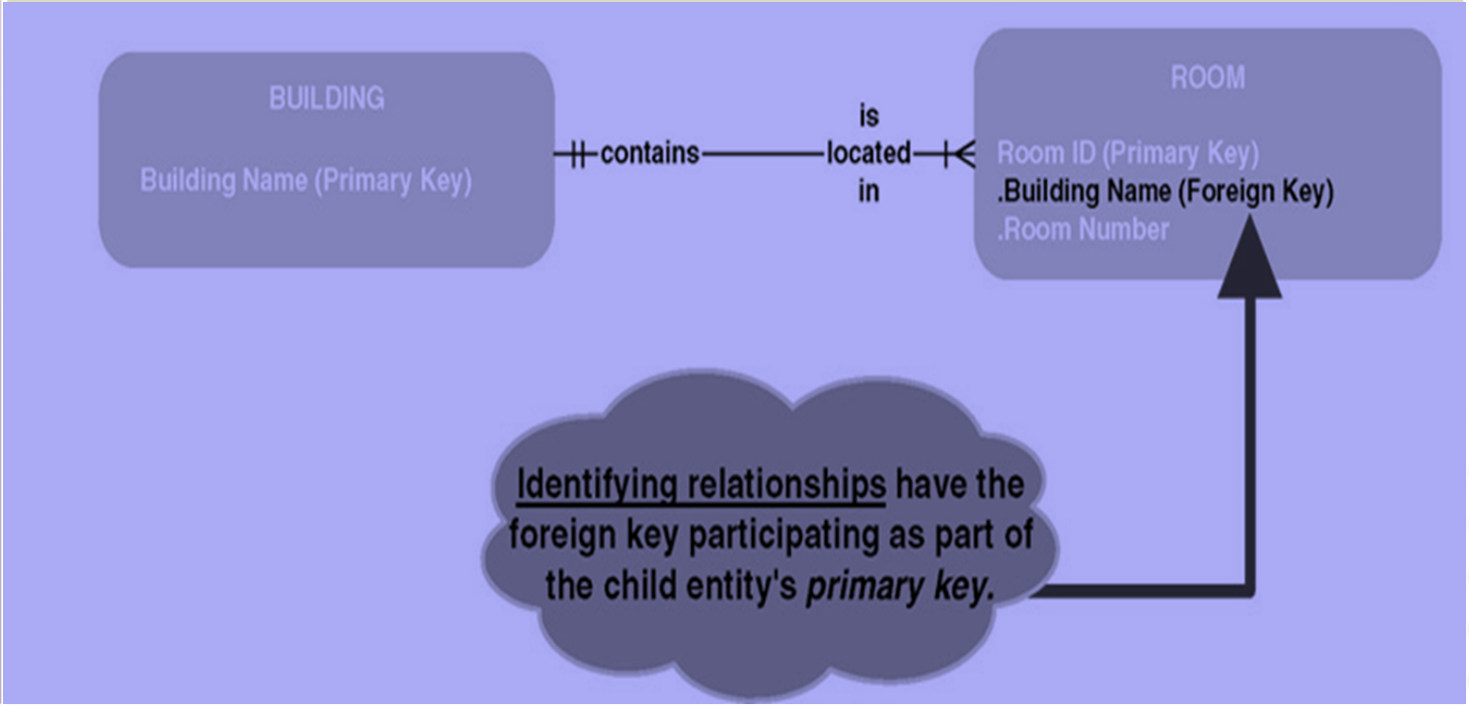
- Primary key attributes are not shared.
- The entities are called *strong* entities



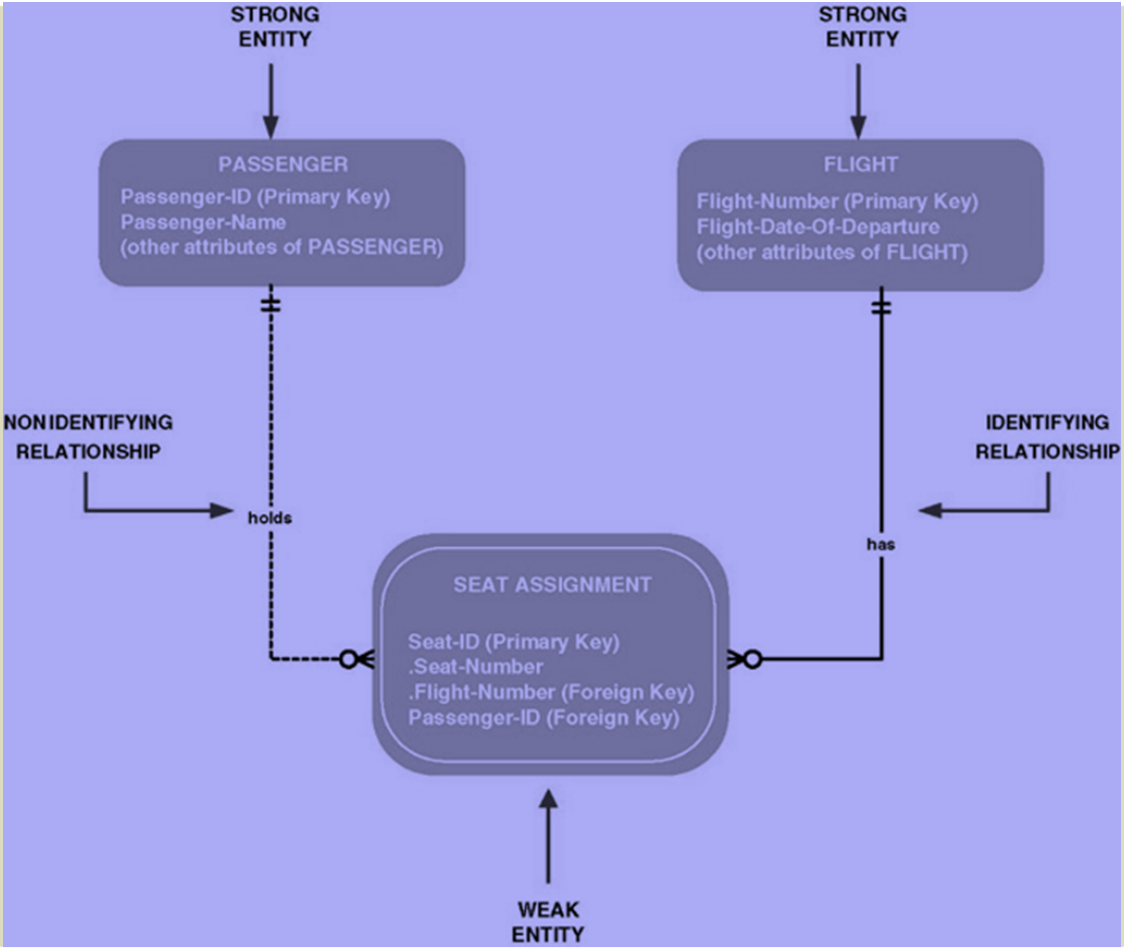
Data Modeling Concepts:
14. Identifying Relationships

Identifying relationship – relationship in which the parent entity' key is also part of the primary key of the child entity.

- The child entity is called a *weak* entity.



Data Modeling Concepts:
15. CASE Tool Notations



Data Modeling Concepts:
16. Generalization

Generalization – a concept wherein the attributes that are common to several types of an entity are grouped into their own entity.

Supertype – an entity whose instances store attributes that are common to one or more entity subtypes.

Subtype – an entity whose instances may inherit common attributes from its entity supertype

And then add other attributes unique to the subtype.

Process of Logical Data Modeling

- **Strategic Data Modeling**
  - Many organizations select IS development projects based on strategic plans.
    - Includes vision and architecture for information systems
    - Identifies and prioritizes develop projects
    - Includes enterprise data model as starting point for projects
- **Data Modeling during Systems Analysis**
  - Data model for a single information system is called an application data model.

Logical Model Development Stages

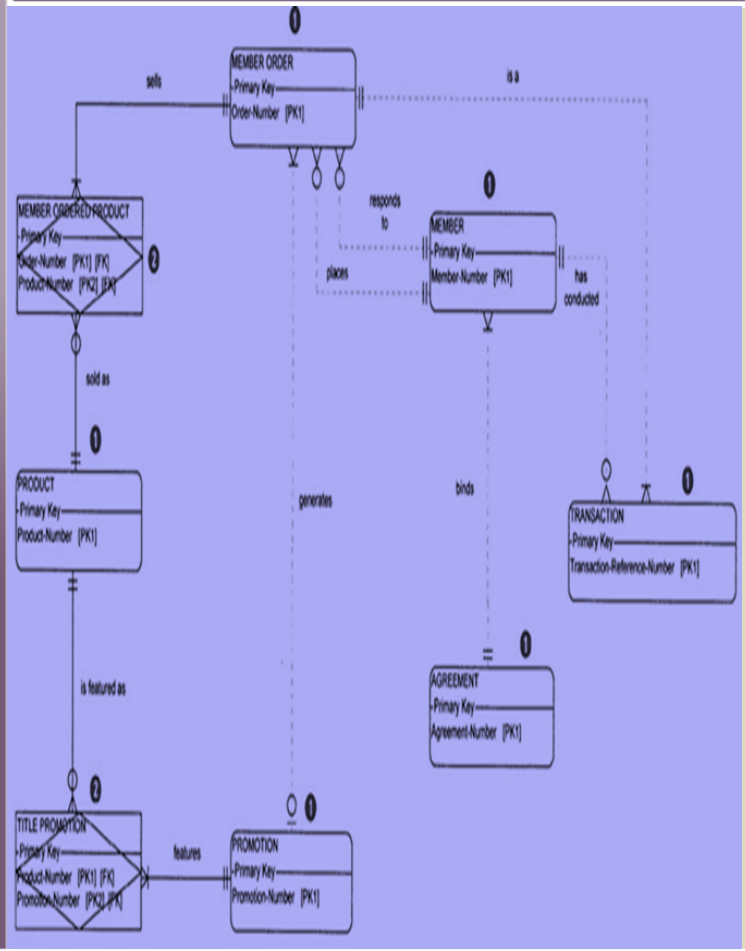
1. **Context Data model**
   - Includes only entities and relationships
   - To establish project scope
2. **Key-based data model**
   - Eliminate nonspecific relationships
   - Add associative entities
   - Include primary and alternate keys
   - Precise cardinalities
3. **Fully attributed data model**
   - All remaining attributes
   - Subsetting criteria
4. **Normalized data model**

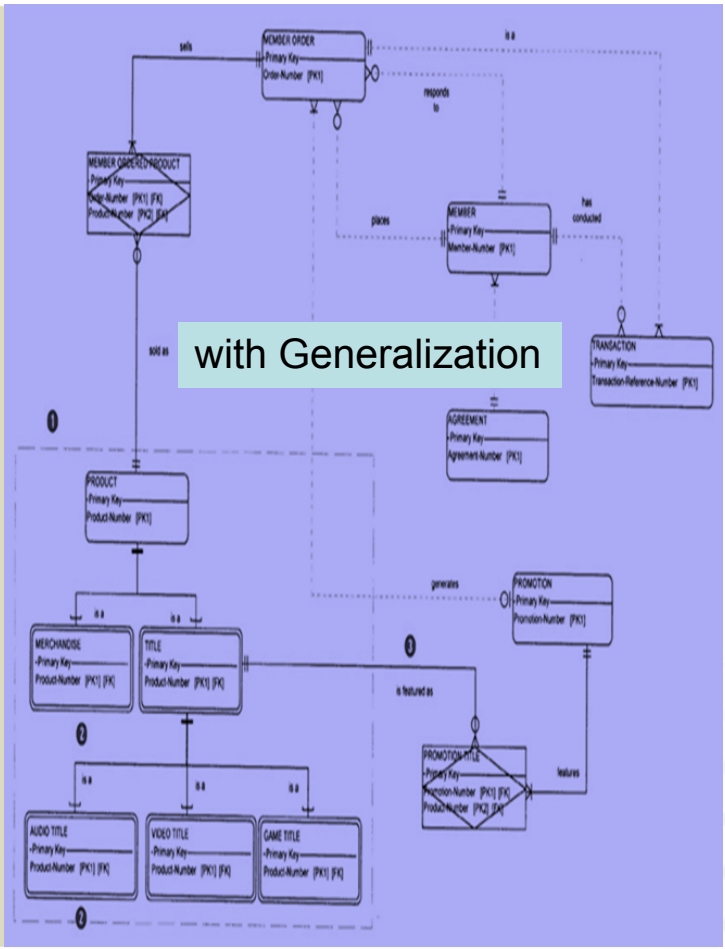
Metadata - data about data.

JRP and Interview Questions for Data Modeling

Purpose	**Candidate Questions** (see textbook for a more complete list)
Discover system entities	What are the subjects of the business?
Discover entity keys	What unique characteristic (or characteristics) distinguishes an instance of each subject from other instances of the same subject?
Discover entity subsetting criteria	Are there any characteristics of a subject that divide all instances of the subject into useful subsets?
Discover attributes and domains	What characteristics describe each subject?
Discover security and control needs	Are there any restrictions on who can see or use the data?
Discover data timing needs	How often does the data change?
Discover generalization hierarchies	Are all instances of each subject the same?
Discover relationships?	What events occur that imply associations between subjects?
Discover cardinalities	Is each business activity or event handled the same way, or are there special circumstances?

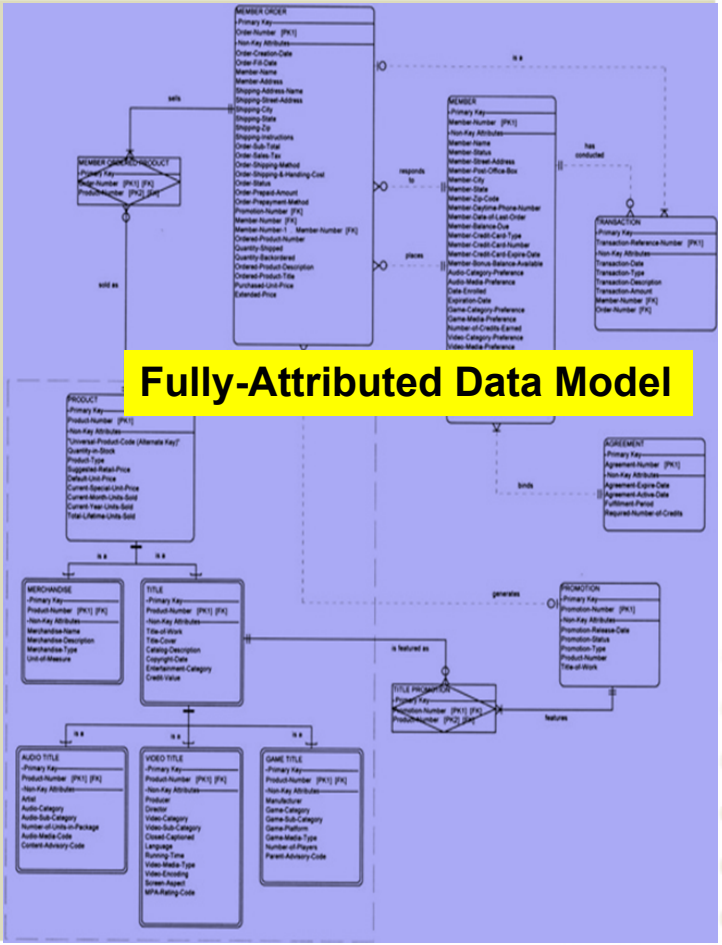
The Key-based Data Model





Good Data Model

- Simple.
  - Data attributes that describe any given entity should describe that entity.
  - Each attribute of an entity instance can have only one value.
- Nonredundant.
  - Each data attribute, other than foreign keys, describes at most one entity.
  - Look for the same attribute recorded more than once under different names.
- Flexible and adaptable to future needs.



Data Analysis & Normalization

Data analysis – a technique used to improve a data model for implementation as a database.

Goal is a simple, nonredundant, flexible, and adaptable database.

Normalization – a data analysis technique that organizes data into groups to form nonredundant, stable, flexible, and adaptive entities.

Normalization: 1NF, 2NF, 3NF

First normal form (1NF) – entity whose attributes have no more than one value for a single instance of that entity

- Any attributes that can have multiple values actually describe a separate entity, possibly an entity and relationship.

Second normal form (2NF) – entity whose nonprimary-key attributes are dependent on the full primary key.

- Any nonkey attributes dependent on only part of the primary key should be moved to entity where that partial key is the full key. May require creating a new entity and relationship on the model.

Third normal form (3NF) – entity whose nonprimary-key attributes are not dependent on any other non-primary key attributes.

- Any nonkey attributes that are dependent on other nonkey attributes must be moved or deleted. Again, new entities and relationships may have to be added to the data model.

Data-to-Location-CRUD Matrix

Entity . Attribute / Location	Customers	Kansas City	. Marketing	. Advertising	. Warehouse	. Sales	. A/R	Boston	. Sales	. Warehouse	San Francisco	. Sales	San Diego	. Warehouse
Customer	**INDV**					**ALL**	**ALL**		**SS**	**SS**		**SS**		**SS**
.Customer Number	R				R	CRUD	R		CRUD	R		CRUD		R
.Customer Name	RU				R	CRUD	R		CRUD	R		CRUD		R
.Customer Address	RU				R	CRUD	R		CRUD	R		CRUD		R
.Customer Credit Rating	X					R	RU		R			R		
.Customer Balance Due	R					R	RU		R			R		
Order	**INDV**		**ALL**		**SS**	**ALL**			**SS**	**SS**		**SS**		**SS**
.Order Number	SRD		R	CRUD	R	CRUD	R		CRUD	R		CRUD		R
.Order Date	SRD		R	CRUD	R	CRUD	R		CRUD	R		CRUD		R
.Order Amount	SRD		R	CRUD		CRUD	R		CRUD	R		CRUD		R
Ordered Product	**INDV**		**ALL**		**SS**	**ALL**			**SS**	**SS**		**SS**		**SS**
.Quantity Ordered	SUD		R	CRUD	R	CRUD	R		CRUD			CRUD		
.Ordered Item Unit Price	SUD		R	CRUD		CRUD	R		CRUD			CRUD		
Product	**ALL**		**ALL**	**ALL**	**ALL**	**ALL**			**ALL**	**ALL**		**ALL**		**ALL**
.Product Number	R		CRUD	R	R	R			R	R		R		R
.Product Name	R		CRUD	R	R	R			R	R		R		R
.Product Description	R		CRUD	RU	R	R			R	R		R		R
.Product Unit of Measure	R		CRUD	R	R	R			R	R		R		R
.Product Current Unit Price	R		CRUD	R		R			R	R		R		R
.Product Quantity on Hand	X				RU	R			R	RU		R		RU

INDV = individual **ALL** = ALL **SS** = subset X = no access

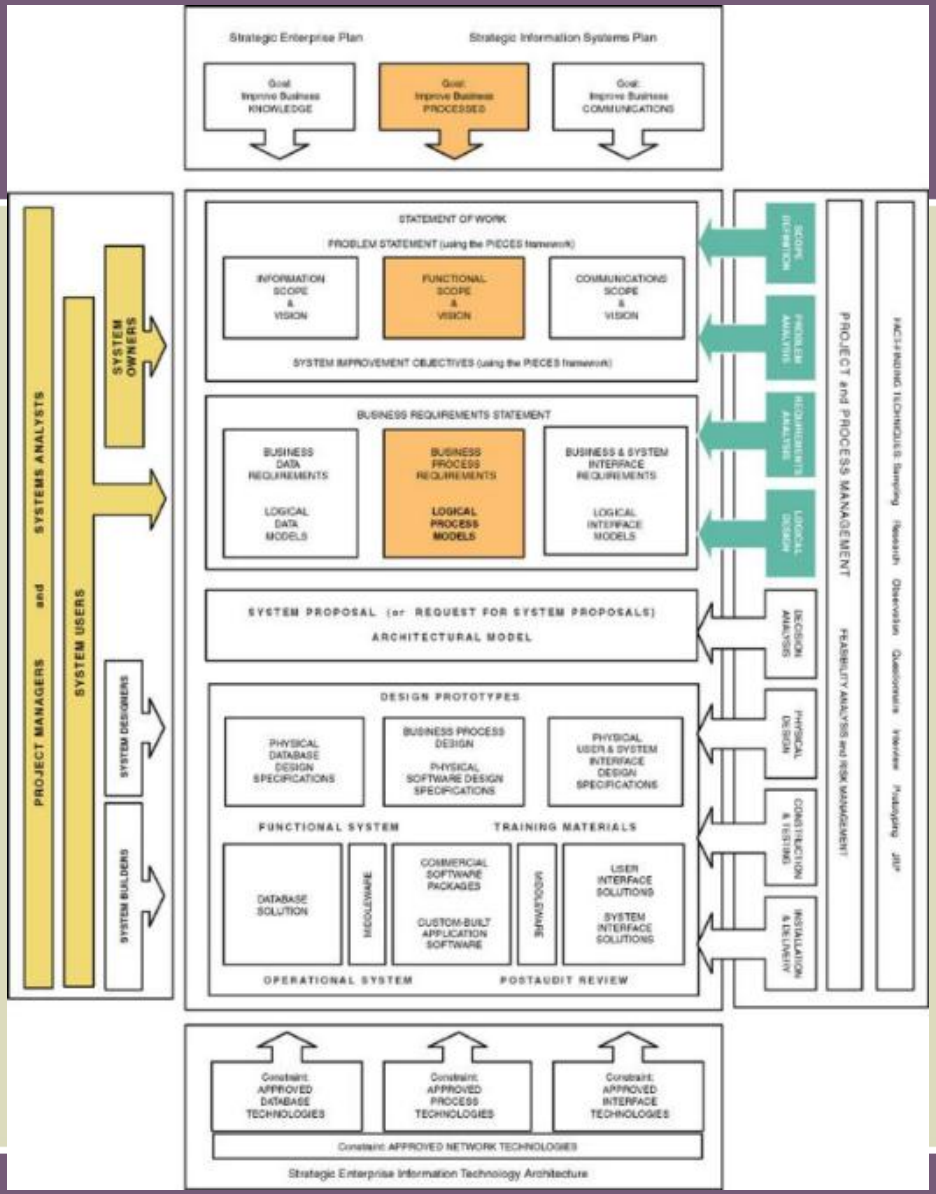
S = submit C = create R = read U = update D = delete

Chapter 9

Process Modeling

Objectives

- Define systems modeling and differentiate logical and physical models.
- Define process modeling and explain its benefits.
- Recognize and understand basic concepts and constructs of a process model.
- Read and interpret a data flow diagram.
- Explain when to construct process models and where to store them.
- Construct a context diagram to illustrate a system's interfaces with its environment.
- Identify use cases, external and temporal business events.
- Perform event partitioning and organize events in a functional decomposition diagram.
- Draw event diagrams and merge them into a system diagram.
- Draw primitive data flow diagrams and describe the elementary data flows in terms of data structures and procedural logic.
- Document the distribution of processes to locations.
- Synchronize data and process models using a CRUD matrix.


Strategic Enterprise Plan
Strategic Information Systems Plan
Goal: Improve Business KNOWLEDGE
Goal: Improve Business PROCESSES
Goal: Improve Business COMMUNICATIONS
PROJECT MANAGERS and SYSTEMS ANALYSTS
SYSTEM USERS
SYSTEM OWNERS
SYSTEM DESIGNERS
SYSTEM BUILDERS
STATEMENT OF WORK
PROBLEM STATEMENT (using the PIECES framework)
INFORMATION SCOPE & VISION
FUNCTIONAL SCOPE & VISION
COMMUNICATIONS SCOPE & VISION
SYSTEM IMPROVEMENT OBJECTIVES (using the PIECES framework)
BUSINESS REQUIREMENTS STATEMENT
BUSINESS DATA REQUIREMENTS
LOGICAL DATA MODELS
BUSINESS PROCESS REQUIREMENTS
LOGICAL PROCESS MODELS
BUSINESS & SYSTEM INTERFACE REQUIREMENTS
LOGICAL INTERFACE MODELS
SYSTEM PROPOSAL (or REQUEST FOR SYSTEM PROPOSALS)
ARCHITECTURAL MODEL
DESIGN PROTOTYPES
PHYSICAL DATABASE DESIGN SPECIFICATIONS
BUSINESS PROCESS DESIGN
PHYSICAL SOFTWARE DESIGN SPECIFICATIONS
PHYSICAL USER & SYSTEM INTERFACE DESIGN SPECIFICATIONS
FUNCTIONAL SYSTEM
TRAINING MATERIALS
DATABASE SOLUTION
MIDDLEWARE
COMMERCIAL SOFTWARE PACKAGES
CUSTOM-BUILT APPLICATION SOFTWARE
GROUPWARE
USER INTERFACE SOLUTIONS
SYSTEM INTERFACE SOLUTIONS
OPERATIONAL SYSTEM
POSTAUDIT REVIEW
SCOPE DEFINITION
PROBLEM ANALYSIS
REQUIREMENTS ANALYSIS
LOGICAL DESIGN
DECISION ANALYSIS
PHYSICAL DESIGN
CONSTRUCTION & TESTING
INSTALLATION & DELIVERY
PROJECT and PROCESS MANAGEMENT
FEASIBILITY ANALYSIS and RISK MANAGEMENT
FACILITATING TECHNIQUES: Sampling Research Observation Questionnaire Interview Prototyping JAD
Constraint: APPROVED DATABASE TECHNOLOGIES
Constraint: APPROVED PROCESS TECHNOLOGIES
Constraint: APPROVED INTERFACE TECHNOLOGIES
Constraint: APPROVED NETWORK TECHNOLOGIES
Strategic Enterprise Information Technology Architecture

Models: Logical and Physical

Model – a pictorial representation of reality.

Just as a picture is worth a thousand words, most models are pictorial representations of reality.

Logical model – a nontechnical pictorial representation that depicts what a system is or does. Synonyms or *essential model*, *conceptual model*, and *business model*.

Physical model – a technical pictorial representation that depicts what a system is or does and how the system is implemented. Synonyms are *implementation model* and *technical model*.

Why Logical System Models

- Logical models remove biases that are the result of the way the system is currently implemented, or the way that any one person thinks the system might be implemented.
- Logical models reduce the risk of missing business requirements because we are too preoccupied with technical results.
- Logical models allow us to communicate with end-users in nontechnical or less technical languages.

Process Modeling and DFDs

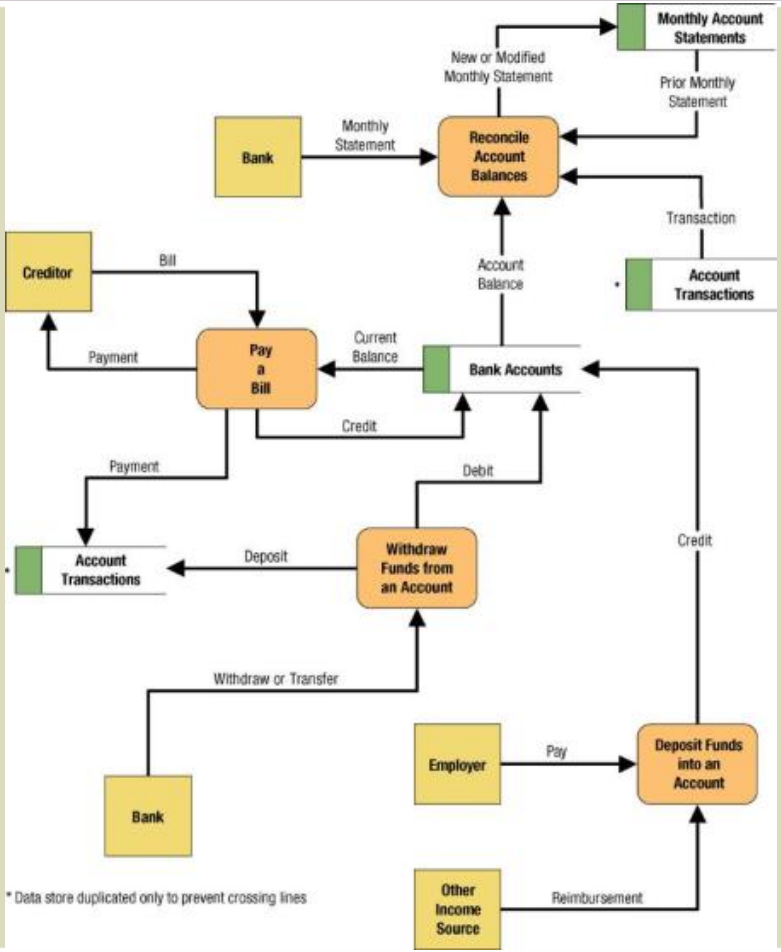
Process modeling – a technique used to organize and document a system's processes.

- Flow of data through processes
- Logic
- Policies
- Procedures

Data flow diagram (DFD) – a process model used to depict the flow of data through a system and the work or processing performed by the system. Synonyms are bubble chart, transformation graph, and process model.

- The DFD has also become a popular tool for business process redesign.

Simple Data Flow Diagram



Differences Between DFDs and Flowcharts

- Processes on DFDs can operate in parallel (at-the-same-time)
  - Processes on flowcharts execute one at a time
- DFDs show the flow of data through a system
  - Flowcharts show the flow of control (sequence and transfer of control)
- Processes on a DFD can have dramatically different timing (daily, weekly, on demand)
  - Processes on flowcharts are part of a single program with consistent timing

External Agents

External agent – an outside person, unit, system, or organization that interacts with a system. Also called an *external entity*.

- External agents define the "boundary" or scope of a system being modeled.
- As scope changes, external agents can become processes, and vice versa.
- Almost always one of the following:
  - Office, department, division.
  - An external organization or agency.
  - Another business or another information system.
  - One of system's end-users or managers
- Named with descriptive, singular noun

Gane and Sarson shape

DeMarco/Yourdon shape

Data Stores

Data store – stored data intended for later use. Synonyms are *file* and *database*.

- Frequently implemented as a file or database.
- A data store is "data at rest" compared to a data flow that is "data in motion."
- Almost always one of the following:
  - Persons (or groups of persons)
  - Places
  - Objects
  - Events (about which data is captured)
  - Concepts (about which data is important)
- Data stores depicted on a DFD store all instances of data entities (depicted on an ERD)
- Named with plural noun

Gane and Sarson shape

DeMarco/Yourdon shape

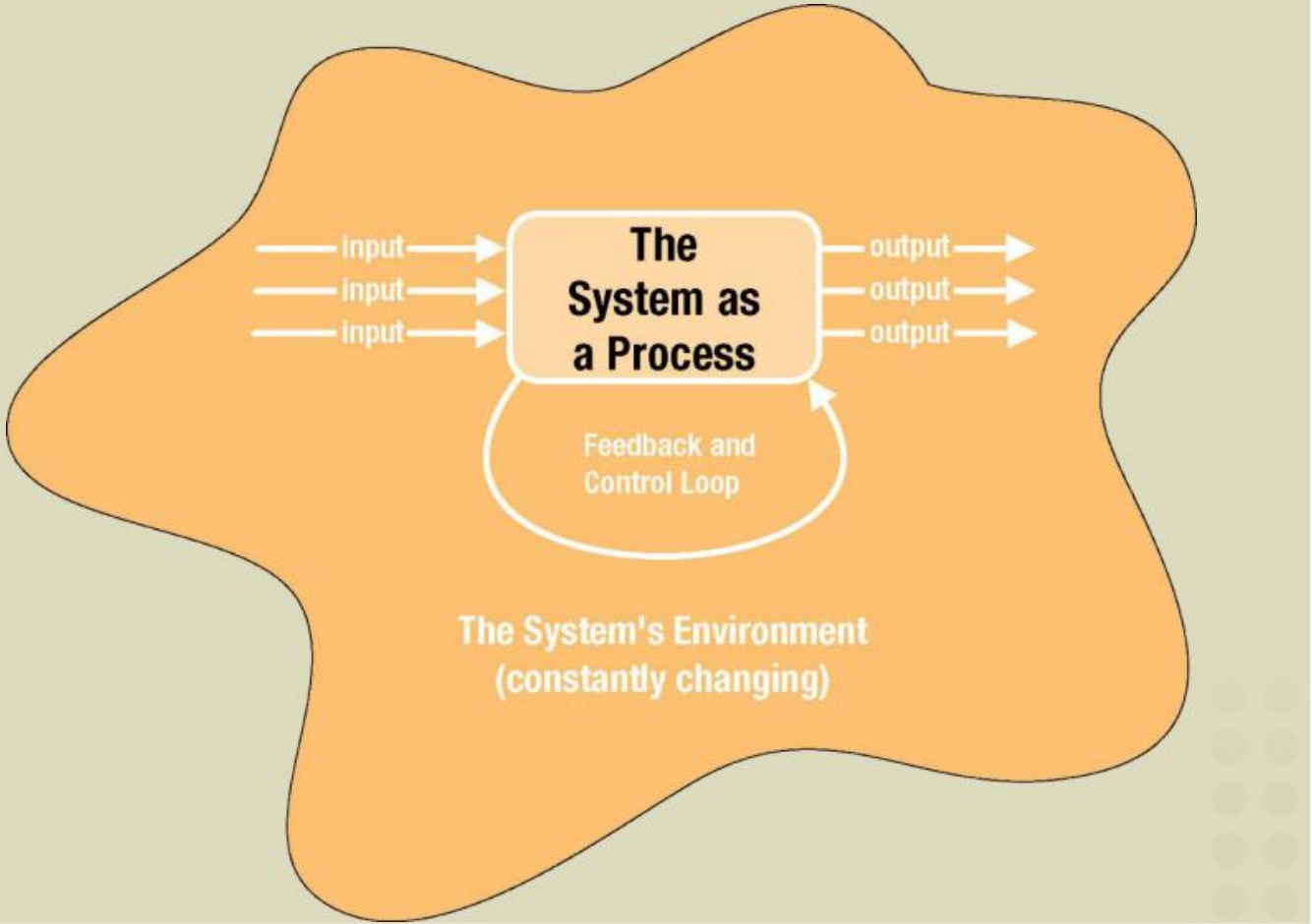
Process Concepts

Process – work performed by a system in response to incoming data flows or conditions. A synonym is *transform*.

- All information systems include processes - usually many of them
- Processes respond to business events and conditions and transform data into useful information
- Modeling processes helps us to understand the interactions with the system's environment, other systems, and other processes.
- Named with a strong action verb followed by object clause describing what the work is performed on/for .

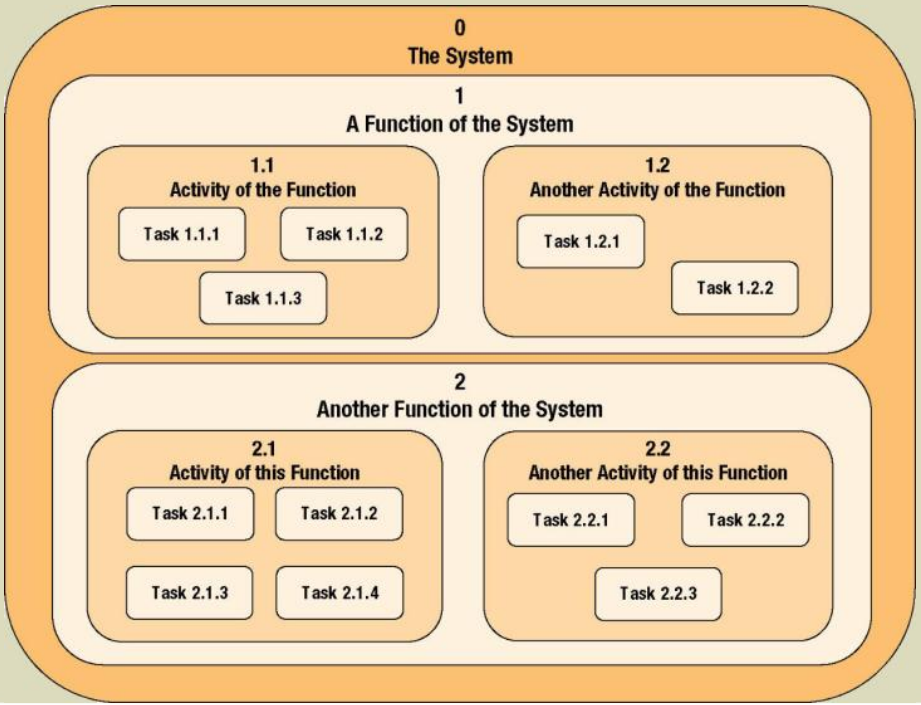
Gane and Sarson

The System is Itself a Process



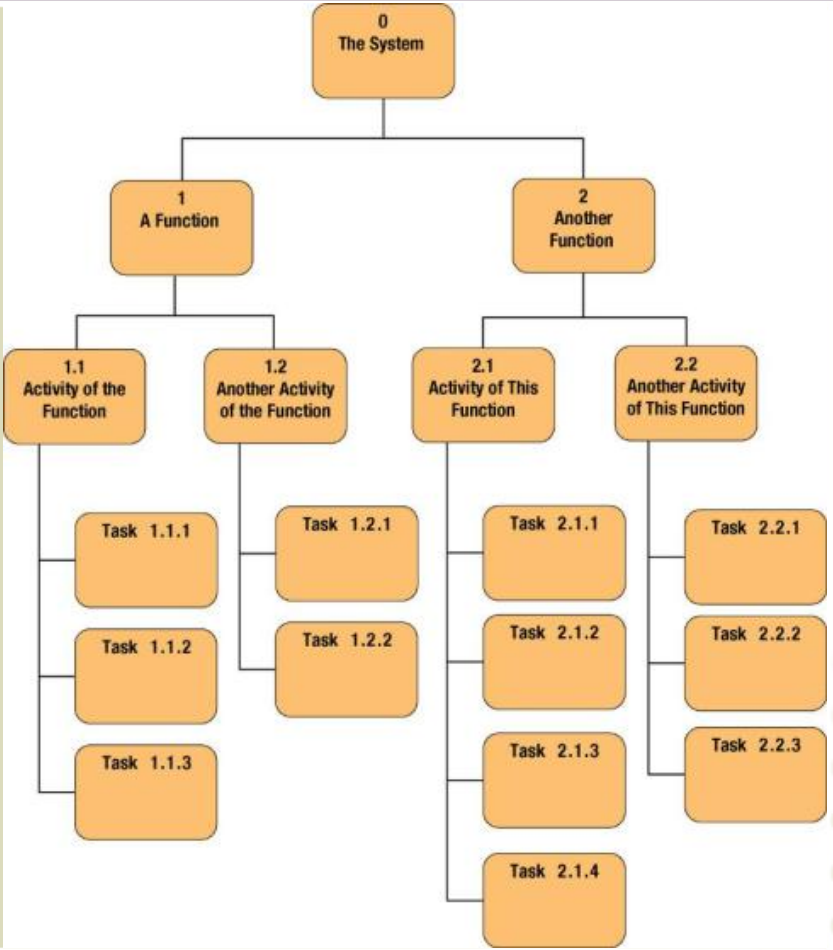
Process Decomposition

Decomposition – the act of breaking a system into sub-components. Each level of abstraction reveals more or less detail.



Decomposition Diagrams

Decomposition diagram – a tool used to depict the decomposition of a system. Also called hierarchy chart.



Types of Logical Processes

Function – a set of related and ongoing activities of a business.

- A function has no start or end.

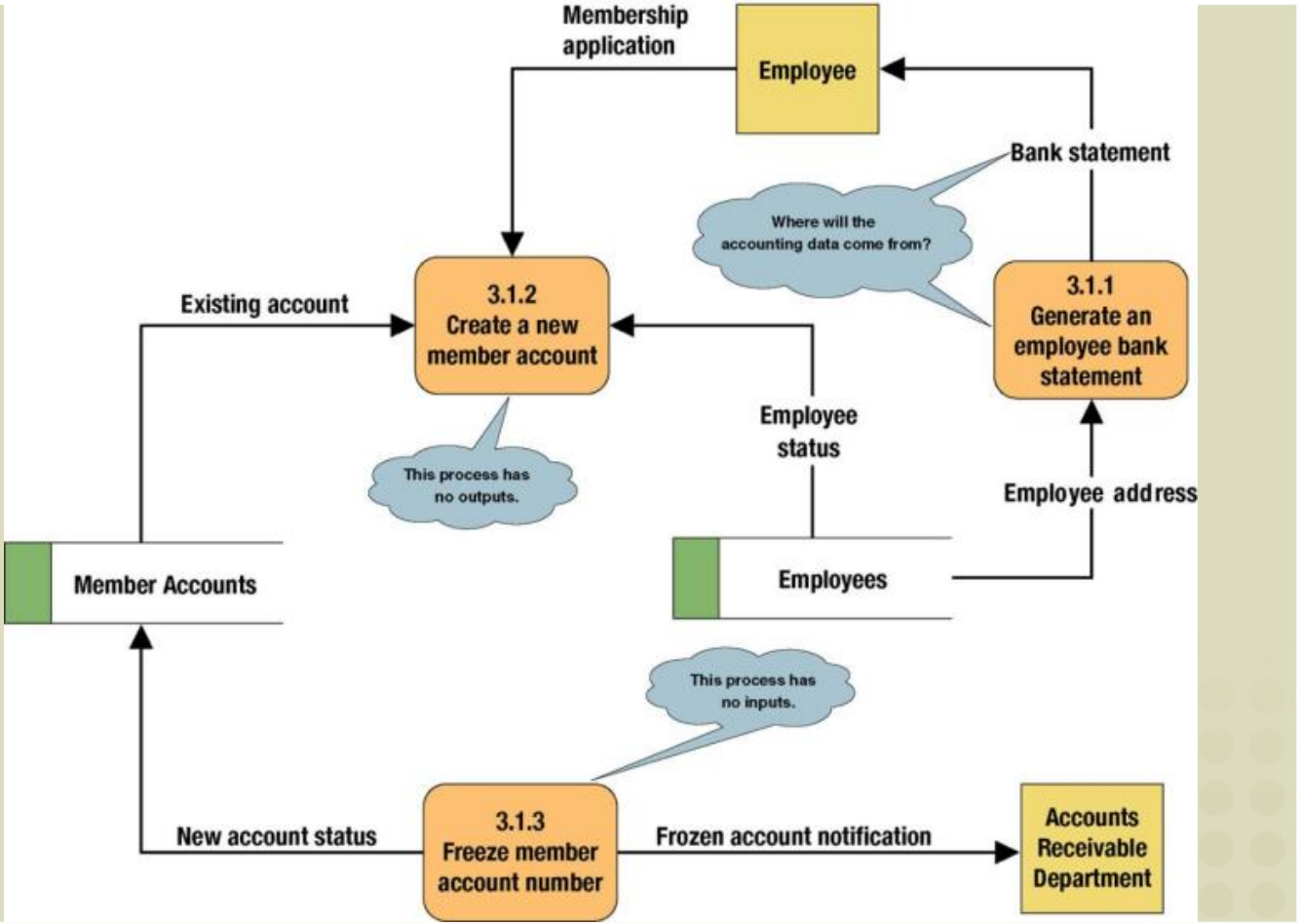
Event – a logical unit of work that must be completed as a whole. Sometimes called a *transaction*.

- Triggered by a discrete input and is completed when process has responded with appropriate outputs.
- Functions consist of processes that respond to events.

Elementary process – a discrete, detailed activity or task required to complete the response to an event. Also called a *primitive process*.

- The lowest level of detail depicted in a process model.

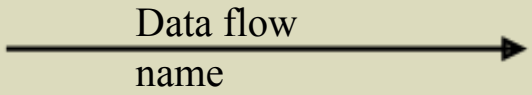
Common Process Errors on DFDs



Data Flows & Control Flows

Data flow – data that is input to or output from a process.

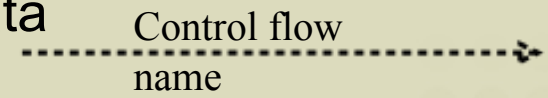
- A data flow is data in motion
- A data flow may also be used to represent the creation, reading, deletion, or updating of data in a file or database (called a data store).



Composite data flow – a data flow that consists of other data flows.

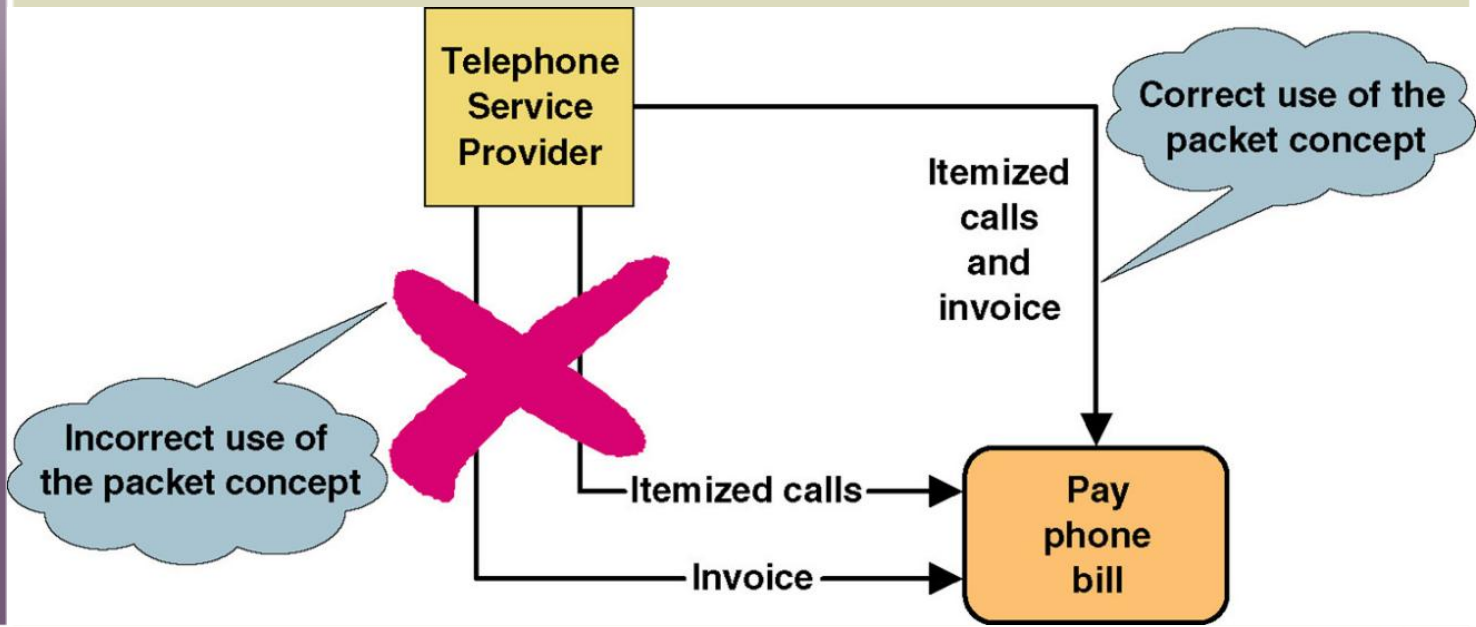
Control flow – a condition or nondata event that triggers a process.

- Used sparingly on DFDs.



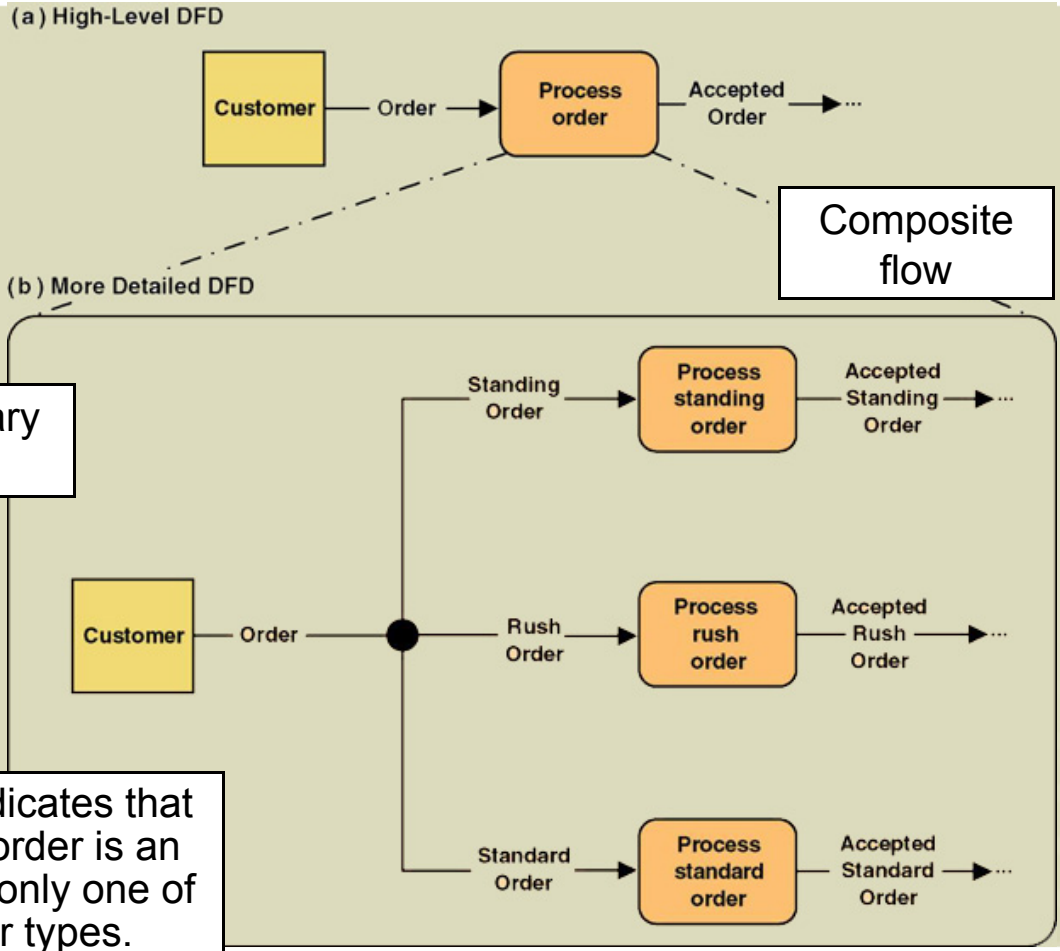
Data Flow Packet Concept

- Data that should travel together should be shown as a single data flow, no matter how many physical documents might be included.



Composite and Elementary Data Flows

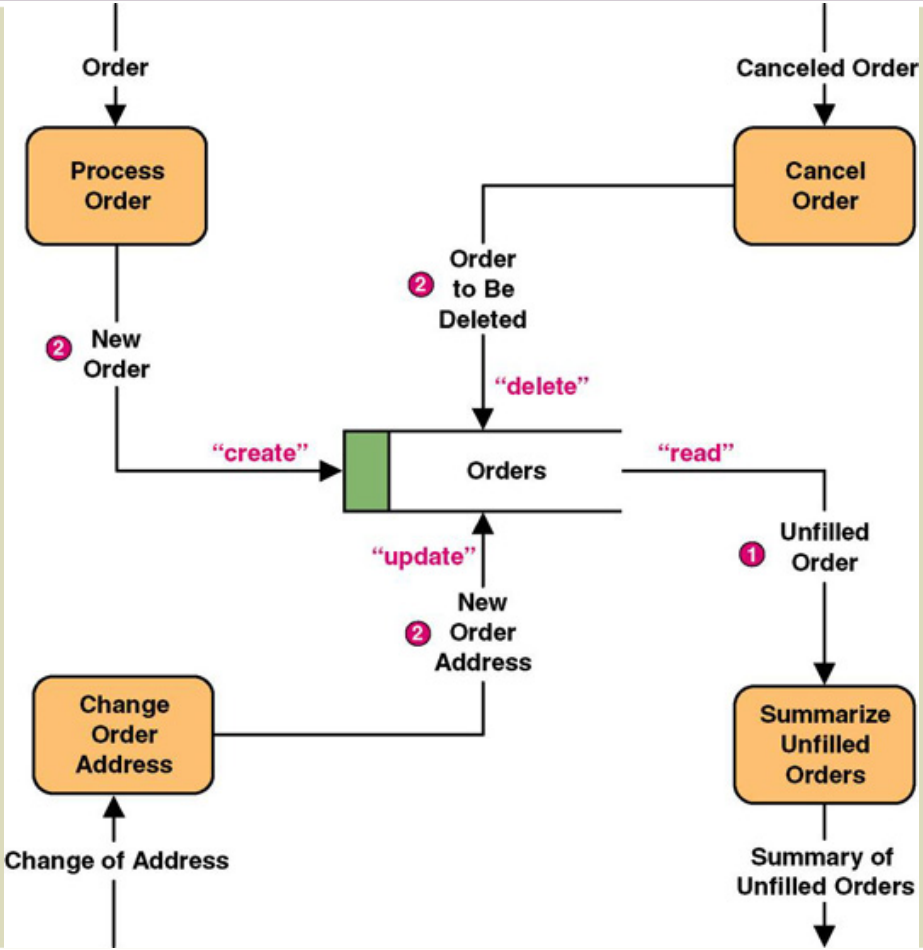
Elementary process – a discrete, detailed activity or task required to complete the response to an event. Also called a *primitive process*.
The lowest level of detail depicted in a process model.



Composite data flow – a data flow that consists of other data flows.

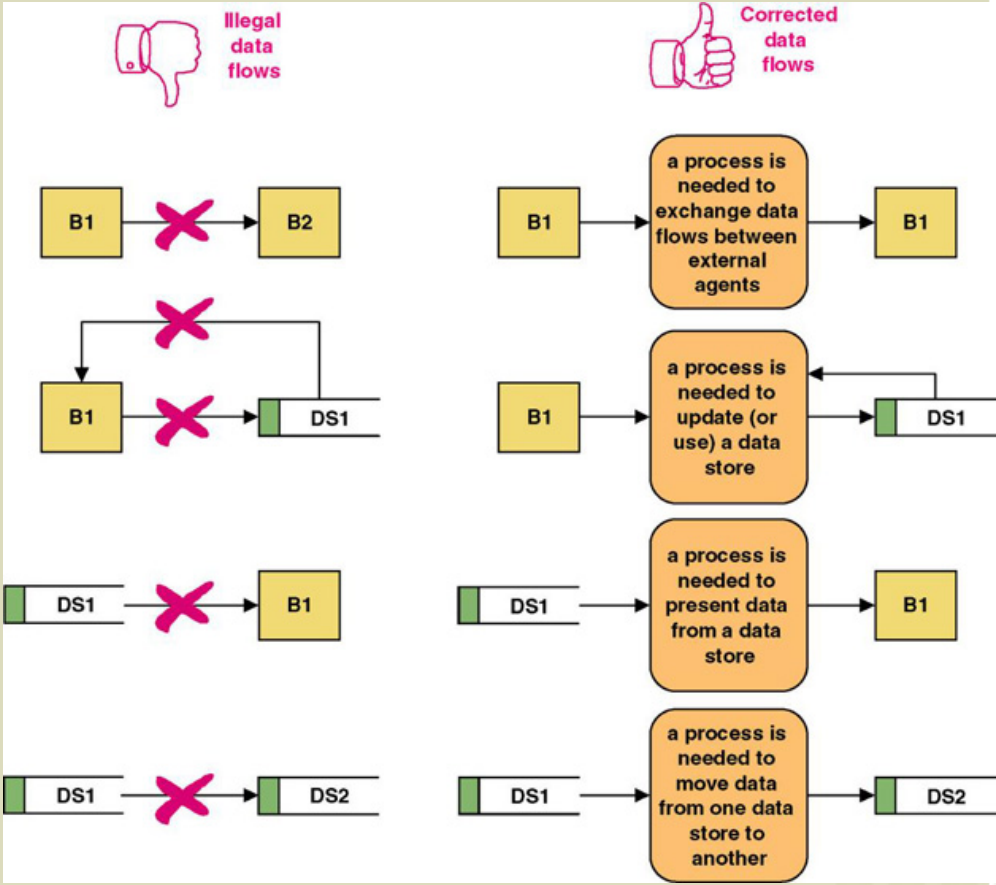
Junction indicates that any given order is an instance of only one of the order types.

Data Flows to and from Data Stores



Rules for Data Flows

- A data flow should never go unnamed.
- In logical modeling, data flow names should describe the data flow without describing the implementation
- All data flows must begin and/or end at a process.



Data Conservation

Data conservation – the practice of ensuring that a data flow contains only data needed by the receiving process.

- Sometimes called *starving the processes*.
- New emphasis on business process redesign to identify and eliminate inefficiencies.
- Simplifies the interface between those processes.
- Must precisely define the data composition of each data flow, expressed in the form of *data structures*.

Data Structures

Data attribute – the smallest piece of data that has meaning to the users and the business.

Data structure – a specific arrangement of data attributes that defines an instance of a data flow.

- The data attributes that comprise a data flow are organized into data structures.
- Data flows can be described in terms of the following types of data structures:
  - A *sequence* or group of data attributes that occur one after another.
  - The *selection* of one or more attributes from a set of attributes.
  - The *repetition* of one or more attributes.

Data Structure for a Data Flow

DATA STRUCTURE	ENGLISH ENTERPRETATION
ORDER= ORDER NUMBER + ORDER DATE+ [PERSONAL CUSTOMER NUMBER, CORPORATE ACCOUNT NUMBER]+ SHIPPING ADDRESS=ADDRESS+ (BILLING ADDRESS=ADDRESS)+ 1 {PRODUCT NUMBER+ PRODUCT DESCRIPTION+ QUANTITY ORDERED+ PRODUCT PRICE+ PRODUCT PRICE SOURCE+ EXTENDED PRICE } N+ SUM OF EXTENDED PRICES+ PREPAID AMOUNT+ (CREDIT CARD NUMBER+EXPIRATION DATE) (QUOTE NUMBER) ADDRESS= (POST OFFICE BOX NUMBER)+ STREET ADDRESS+ CITY+ [STATE, MUNICIPALITY]+ (COUNTRY)+ POSTAL CODE	An instance of ORDER consists of: ORDER NUMBER and ORDER DATE and Either PERSONAL CUSTOMER NUMBER or CORPORATE ACCOUNT NUMBER and SHIPPING ADDRESS (which is equivalent to ADDRESS) and optionally: BILLING ADDRESS (which is equivalent to ADDRESS) and one or more instances of: PRODUCT NUMBER and PRODUCT DESCRIPTION and QUANTITY ORDERED and PRODUCT PRICE and PRODUCT PRICE SOURCE and EXTENDED PRICE and SUM OF EXTENDED PRICES and PREPAID AMOUNT and optionally: both CREDIT CARD NUMBER and EXPIRATION DATE An instance of ADDRESS consists of: optionally: POST OFFICE BOX NUMBER and STREET ADDRESS and CITY and Either STATE or MUNICIPALITY and optionally: COUNTRY and POSTAL CODE

Data Structure Constructs

Data Structure	Format by Example (relevant portion is **boldfaced**)	English Interpretation (relevant portion is **boldfaced**)
Sequence of Attributes - The sequence data structure indicates one or more attributes that may (or must) be included in a data flow.	WAGE AND TAX STATEMENT= **TAXPAYER IDENTIFICATION NUMBER+** **TAXPAYER NAME+** **TAXPAYER ADDRESS+** **WAGES, TIPS, AND COMPENSATION+** **FEDERAL TAX WITHHELD+…**	An instance of WAGE AND TAX STATEMENTS consists of: **TAXPAYER IDENTIFICATION NUMBER and** **TAXPAYER NAME and** **TAXPAYER ADDRESS and** **WAGES, TIPS AND COMPENSATION and** **FEDERAL TAX WITHHELD and…**
Selection of Attributes - The selection data structure allows you to show situations where different sets of attributes describe different instances of the data flow.	ORDER= **(PERSONAL CUSTOMER NUMBER,** **CORPORATE ACCOUNT NUMBER)+** ORDER DATE+…	An instance of ORDER consists of: **Either PERSONAL CUSTOMER NUMBER or CORPORATE ACCOUNT NUMBER;** and ORDER DATE and…

Data Structure Constructs (continued)

Data Structure	Format by Example (relevant portion is **boldfaced**	English Interpretation (relevant portion is **boldfaced**)
Repetition of Attributes - The repetition data structure is used to set off a data attribute or group of data attributes that may (or must) repeat themselves a specific number of time for a single instance of the data flow. The minimum number of repetitions is usually *zero* or *one*. The maximum number of repetitions may be specified as "n" meaning "many" where the actual number of instances varies for each instance of the data flow.	POLICY NUMBER+ POLICYHOLDER NAME+ POLICY HOLDER ADDRESS+ **0 {DEPENDENT NAME+** **DEPENDENT'S RELATIONSHIP} N+** **1 {EXPENSE DESCRIPTION+** **SERVICE PROVIDER+** **EXPENSE AMOUNT} N**	An instance of CLAIM consists of: POLICY NUMBER and POLICYHOLDER NAME and POLICYHOLDER ADDRESS and **zero or more instance of:** **DEPENDENT NAME and** **DEPENDENT'S RELATIONSHIP and** **one or more instances of:** **EXPENSE DESCRIPTION and** **SERVICE PROVIDER and** **EXPENSE ACCOUNT**

Data Structure Constructs (concluded)

Data Structure	Format by Example (relevant portion is **boldfaced**)	English Interpretation (relevant portion is **boldfaced**)
Optional Attributes - The optional notation indicates that an attribute, or group of attributes in a sequence or selection date structure may not be included in all instances of a data flow. *Note: For the repetition data structure, a minimum of "zero" is the same as making the entire repeating group "optional."*	CLAIM= POLICY NUMBER+ POLICYHOLDER NAME+ POLICYHOLDER ADDRESS+ **(SPOUSE NAME+** **DATE OF BIRTH)**+…	An instance of CLAIM consists of: POLICY NUMBER and POLICYHOLDER NAME and POLICYHOLDER ADDRESS and **optionally, SPOUSE NAME and** **DATE OF BIRTH** and…
Reusable Attributes - For groups of attributes that are contained in many data flows, it is desirable to create a separate data structure that can be reused in other data structures.	DATE= MONTH+ DAY+ YEAR+	Then, the reusable structures can be included in other data flow structures as follows: ORDER=ORDER NUMBER…+DATE INVOICE=INVOICE NUMBER…+DATE PAYMENT=CUSTOMER NUMBER…+DATE

Data Types and Domains

Data attributes should be defined by data types and domains.

Data type - a class of data that be stored in an attribute.

- Character, integers, real numbers, dates, pictures, etc.

Domain – the legitimate values for an attribute.

Diverging and Converging Data Flows

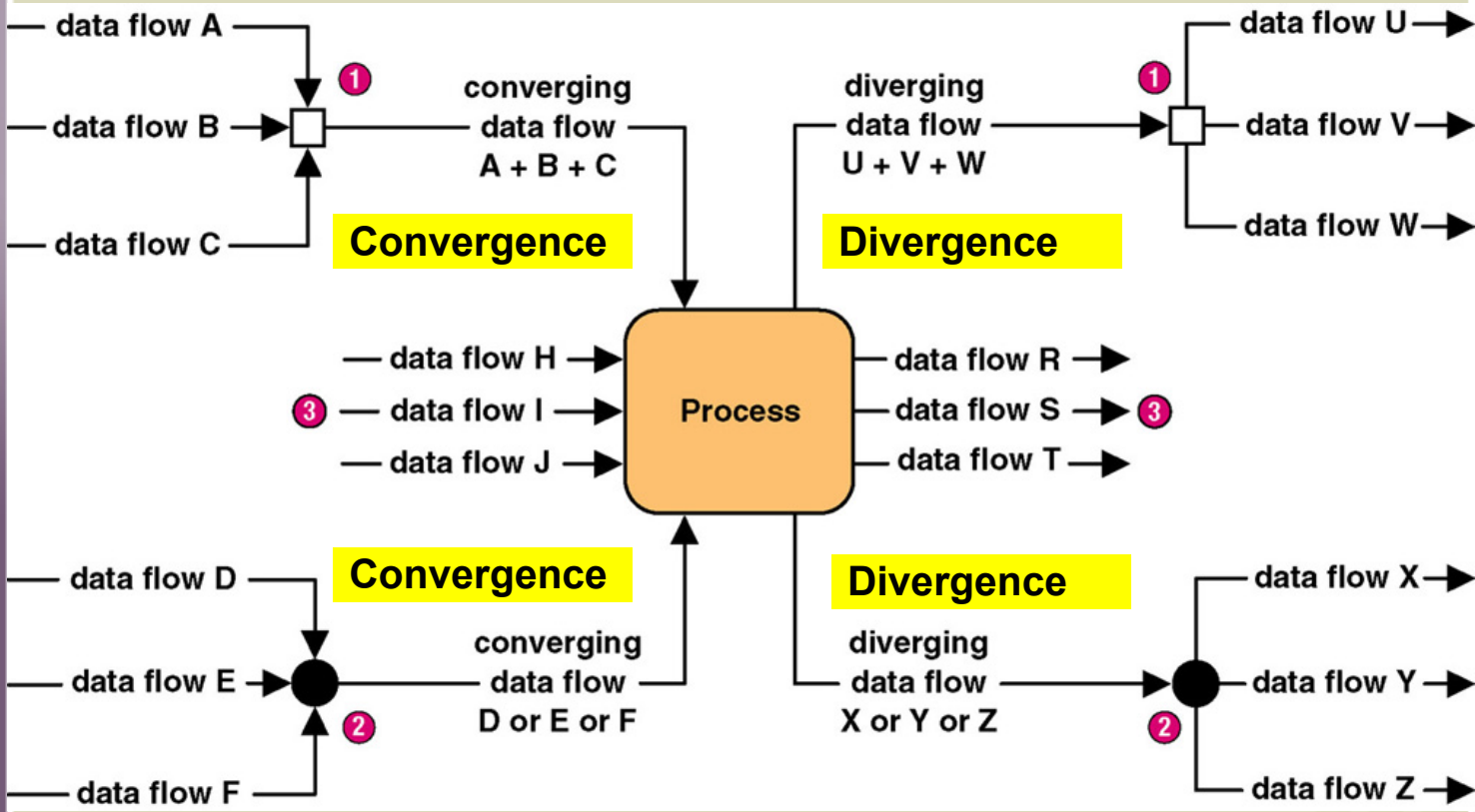
Diverging data flow – a data flow that splits into multiple data flows.

- Indicates data that starts out naturally as one flow, but is routed to different destinations.
- Also useful to indicate multiple copies of the same output going to different destinations.

Converging data flow – the merger of multiple data flows into a single packet.

- Indicates data from multiple sources that can (must) come together as a single packet for subsequent processing.

Diverging and Converging Data Flows



When to Draw Process Models

- Strategic systems planning
  - Enterprise process models illustrate important business functions.
- Business process redesign
  - "As is" process models facilitate critical analysis.
  - "To be" process models facilitate improvement.
- Systems analysis (primary focus of this course)
  - Model existing system including its limitations
  - Model target system's logical requirements
  - Model candidate technical solutions
  - Model the target technical solution

Classical Structured Analysis

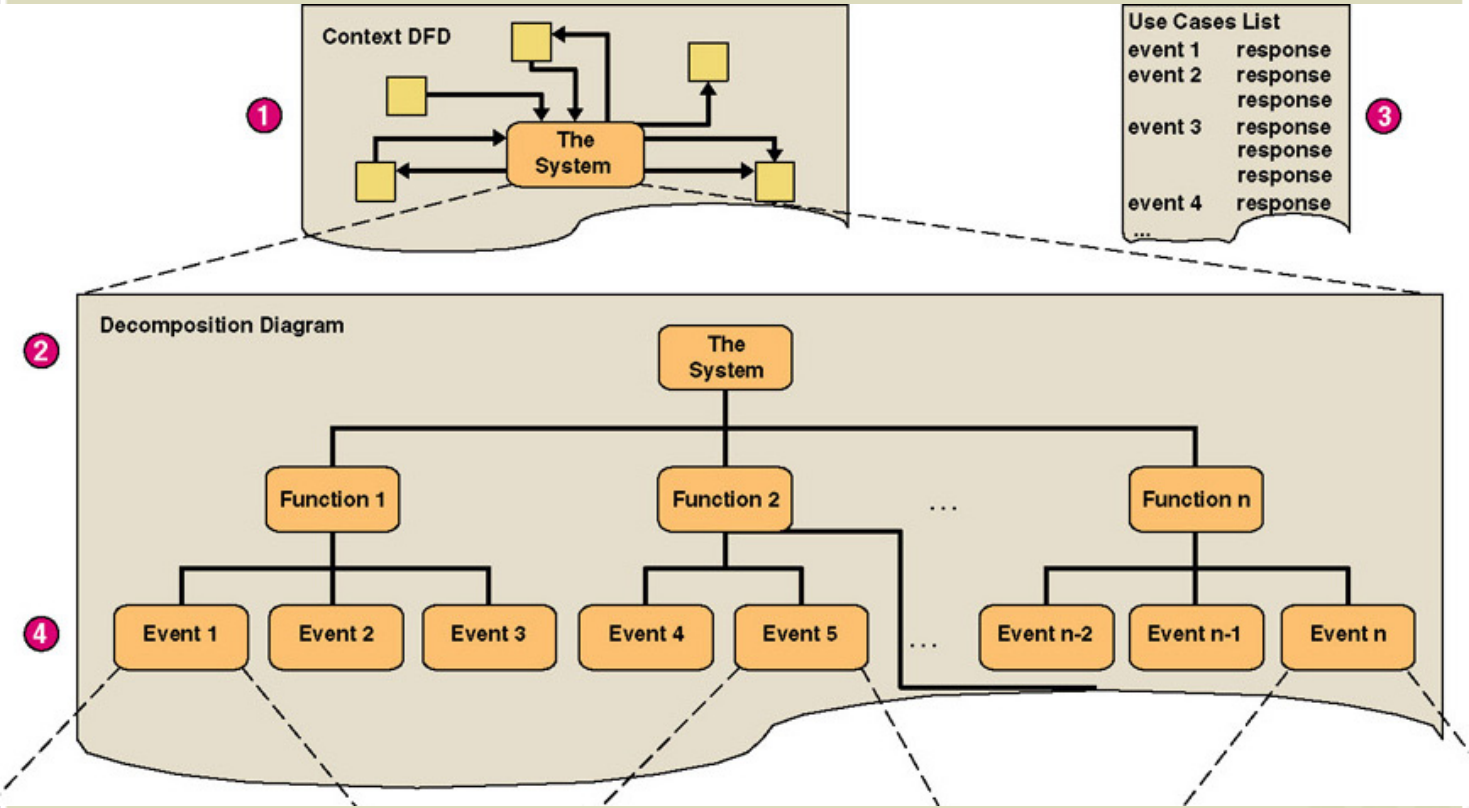
Rarely practiced anymore because cumbersome & time-consuming

1. Draw top-down physical DFDs that represent current physical implementation of the system.
2. Convert physical DFDs to logical equivalents.
3. Draw top-down logical DFDs that represent improved system.
4. Describe all data flows, data stores, policies, and procedures in data dictionary or encyclopedia.
5. Optionally, mark up copies of the logical DFDs to represent alternative physical solutions.
6. Draw top-down physical DFDs representing target solution.

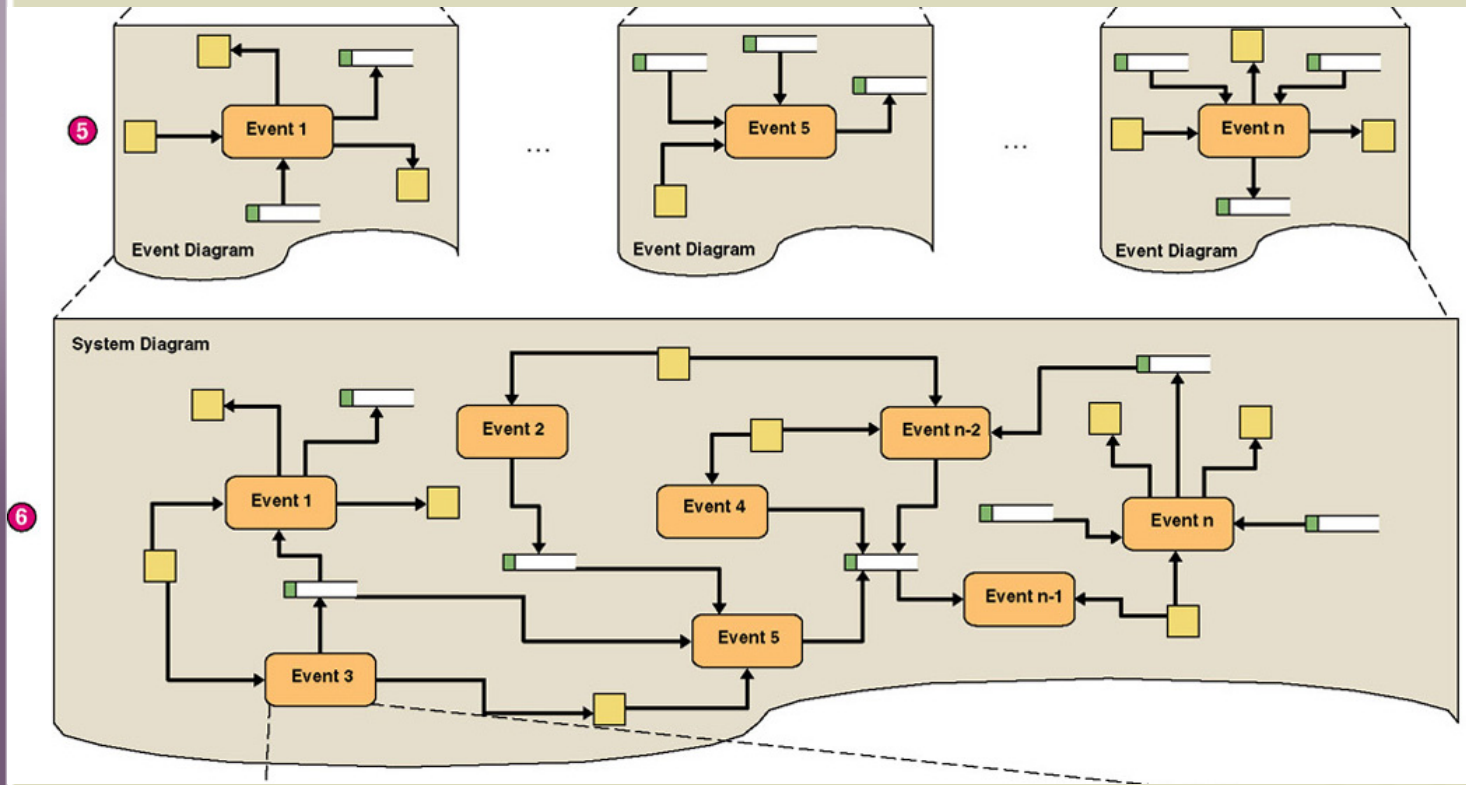
Modern Structured Analysis (More Commonly Practiced)

1. Draw context DFD to establish initial project scope.
2. Draw functional decomposition diagram to partition the system into subsystems.
3. Create event-response or use-case list for the system to define events for which the system must have a response.
4. Draw an event DFD (or event handler) for each event.
5. Merge event DFDs into a system diagram (or, for larger systems, subsystem diagrams).
6. Draw detailed, primitive DFDs for the more complex event handlers.
7. Document data flows and processes in data

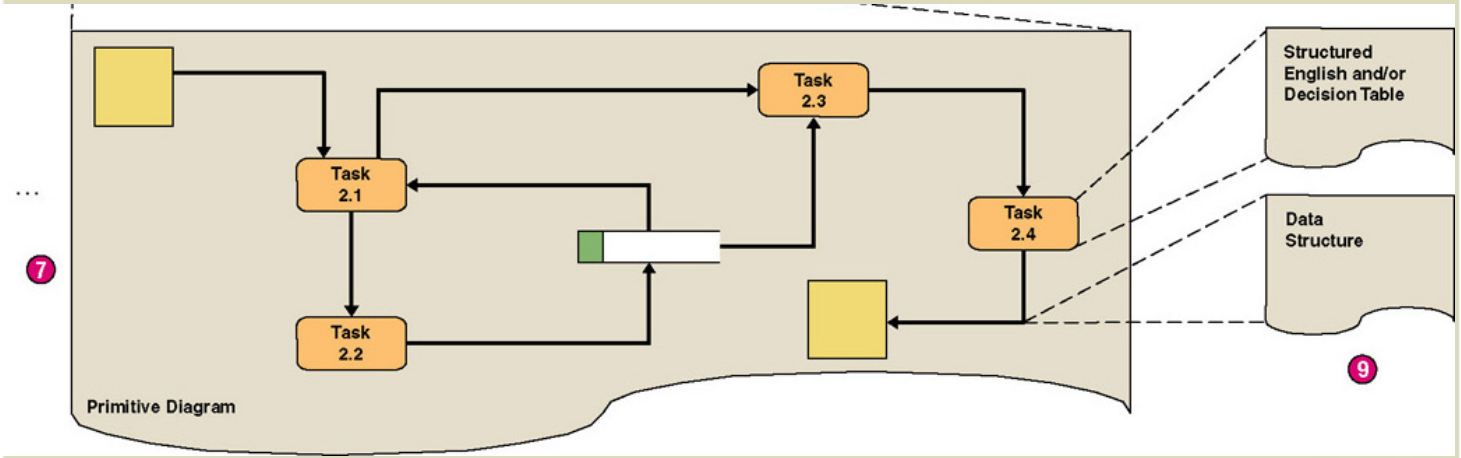
Structured Analysis Diagram Progression (1 of 3)



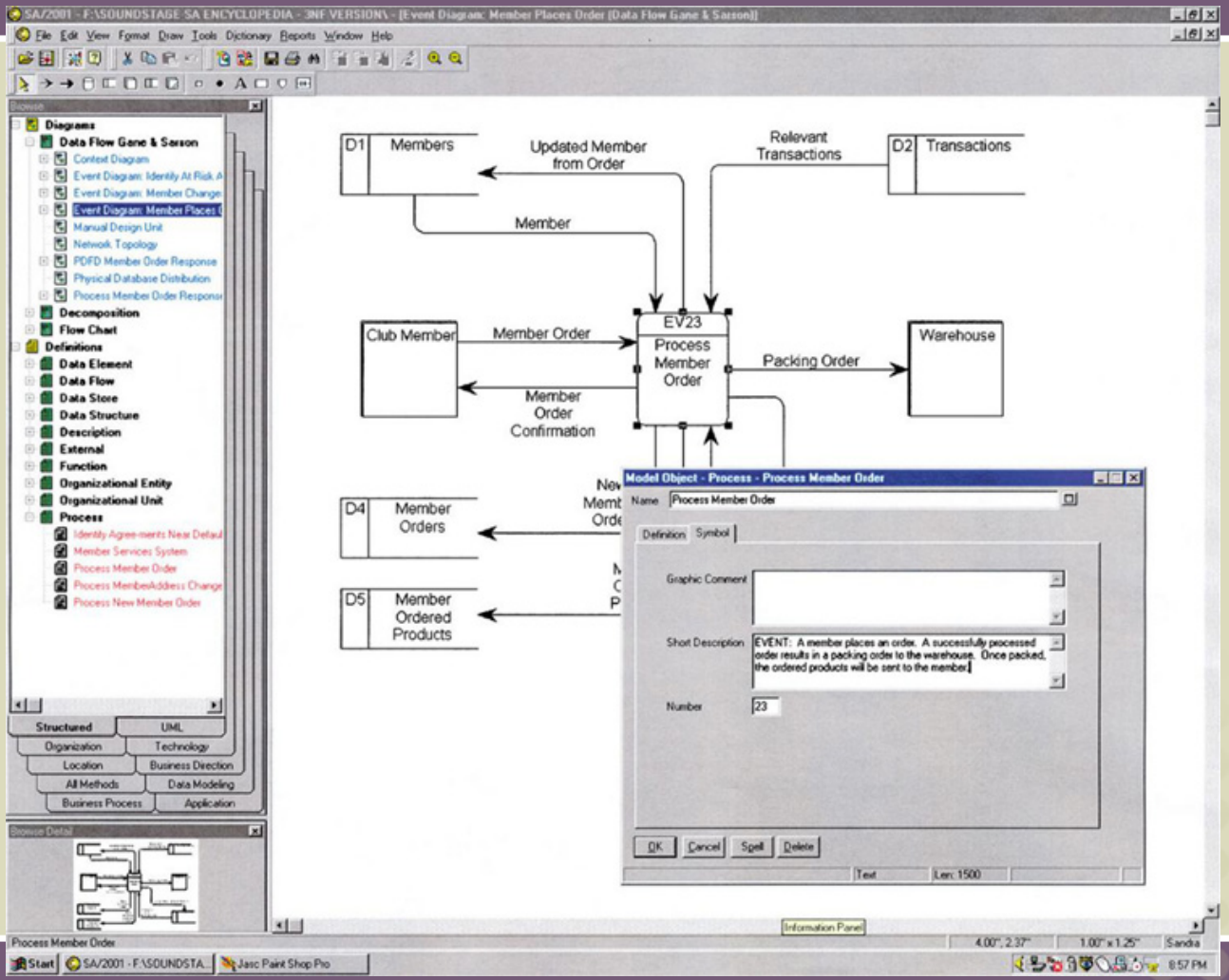
Structured Analysis Diagram Progression (2 of 3)



Structured Analysis Diagram Progression (3 of 3)



CASE for Process Modeling



How to construct Process Models

1. The context Data Flow Diagram
2. The Functional Decomposition Diagram
3. The Event Response or Use Case List
4. Event Decomposition Diagrams
5. Event Diagrams
6. The System Diagrams
7. Primitive Diagrams
8. Completing the Specification

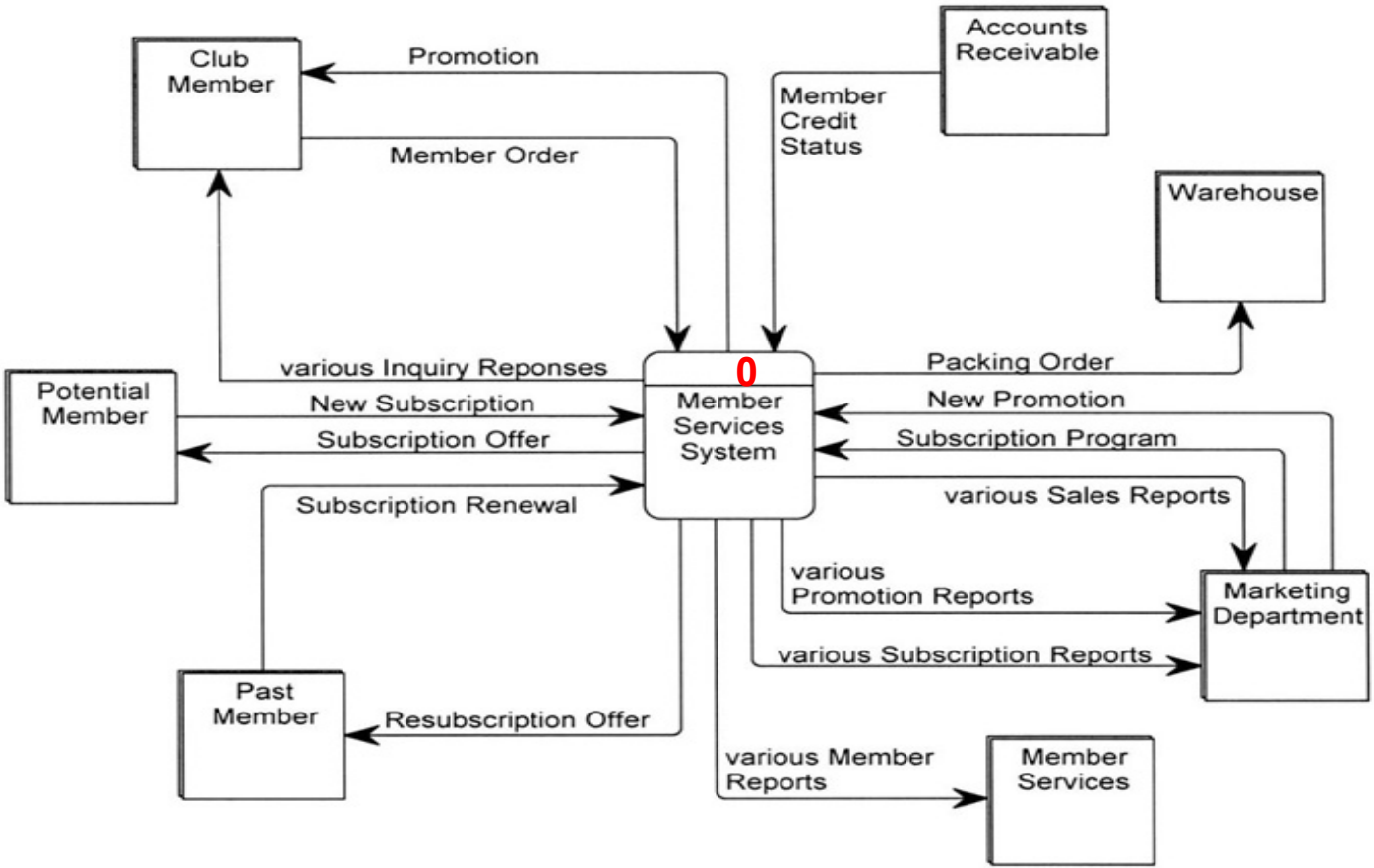
1. Context Data Flow Diagram

- **Context data flow diagram** - a process model used to document the scope for a system. Also called the environmental model.

Initial Project Scope - What aspect a system supports in a business.
- How modeled system interacts with business.

1. Think of the system as a "black box."
2. Ask users what business transactions the system must respond to. These are inputs, and the sources are external agents.
3. Ask users what responses must be produced by the system. These are outputs, and the destinations are external agents.
4. Identify any external data stores, if any.
5. Draw a context diagram.

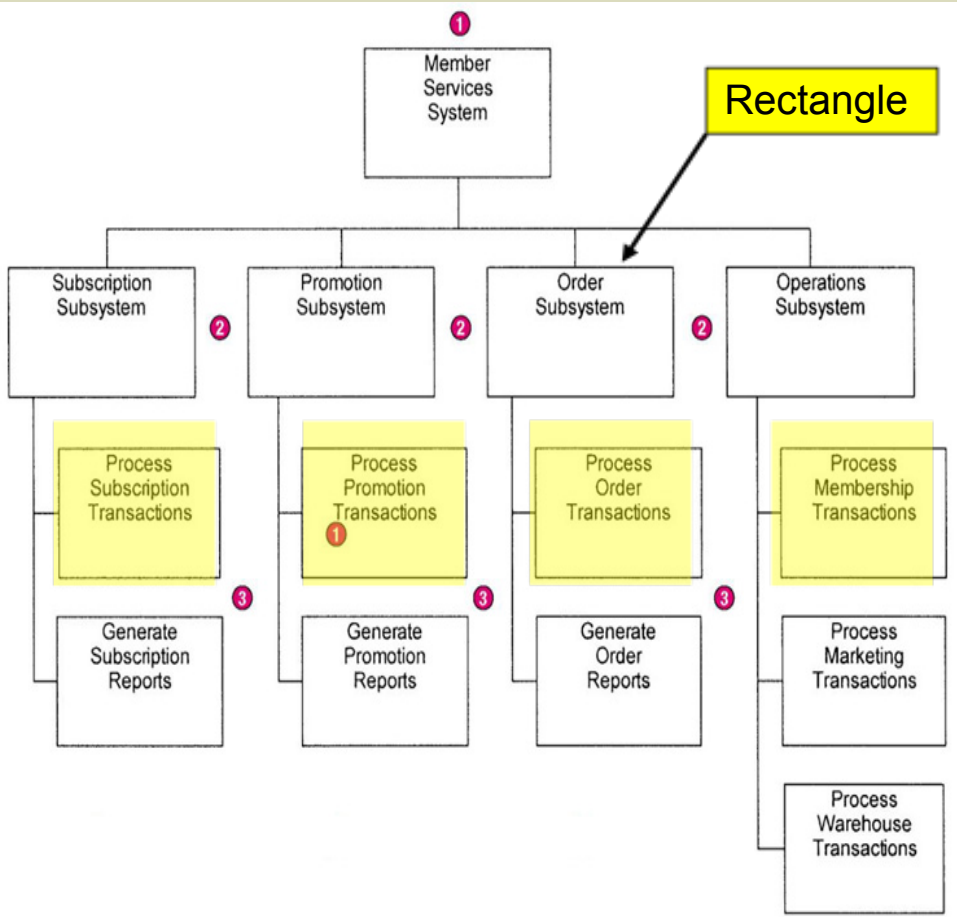
1. Context Data Flow Diagram



2. Functional Decomposition Diagram

Top down structure of a system

1. Break system into sub-components.
2. Every process should be factored into at least two child processes.
3. Larger systems might be factored into subsystems and functions.



3. Events-Response or Use Case List

Business Events and System responses

1. External events

- Are initiated by external agents/actors.

  Eg. Customer Places new order (event)..
  Order (Input data flow)..
  Customer (external agent)

- Result in input transaction or data flow.

2. Temporal events

- Are triggered on the basis of time, or something happens. They are indicated by a control flow.

  Eg. Time to remind customers to pay invoices

3. State events

- Are triggered based on a system's change from one state/condition to another.
- They are indicated by a control flow.

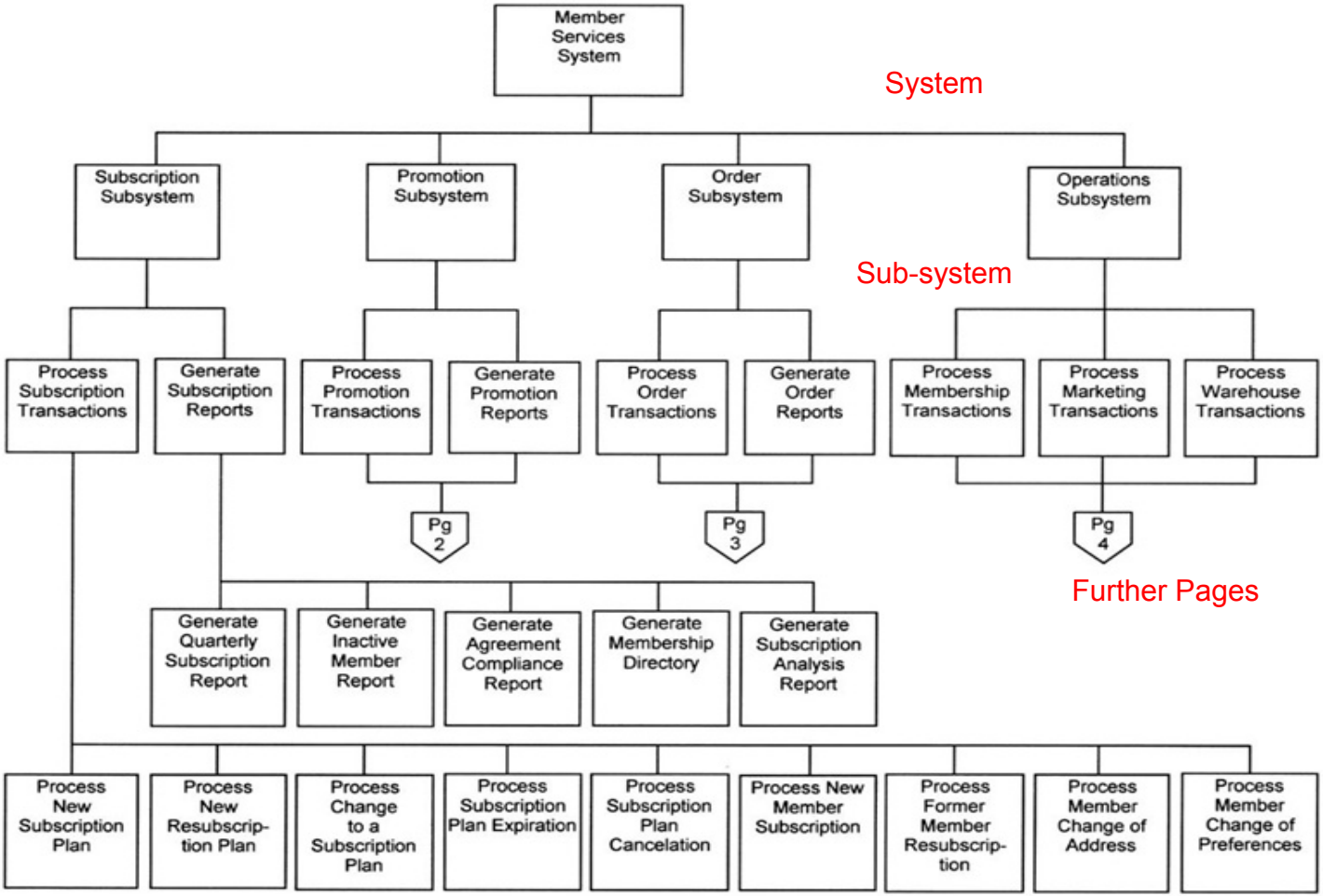
  Eg. Remind customers to pay invoices

- **Use case** – an analysis tool for finding and identifying business events and responses.
- **Actor** – anything that interacts with a system.

Use Case Table

Actor/ External Agent	Event (or Use Case)	Trigger	Response
Marketing	Establishes a new membership subscription plan to entice new members.	New Member Subscription Program	Generate Subscription Plan Confirmation. Create Agreement in the database.
Marketing	Establishes a new membership resubscription plan to lure back former members.	Past Member Resubscription Program	Generate Subscription Plan Confirmation. Create Agreement in the database.
(time)	A subscription plan expires.	(current date)	Generate Agreement Change Confirmation. Logically delete Agreement in database.
Member	Joins club by subscribing.	New Subscription	Generate Member Directory Update Confirmation. Create Member in database. Create first Member Order and Member Ordered Products in database.

4. Event Decomposition Diagrams



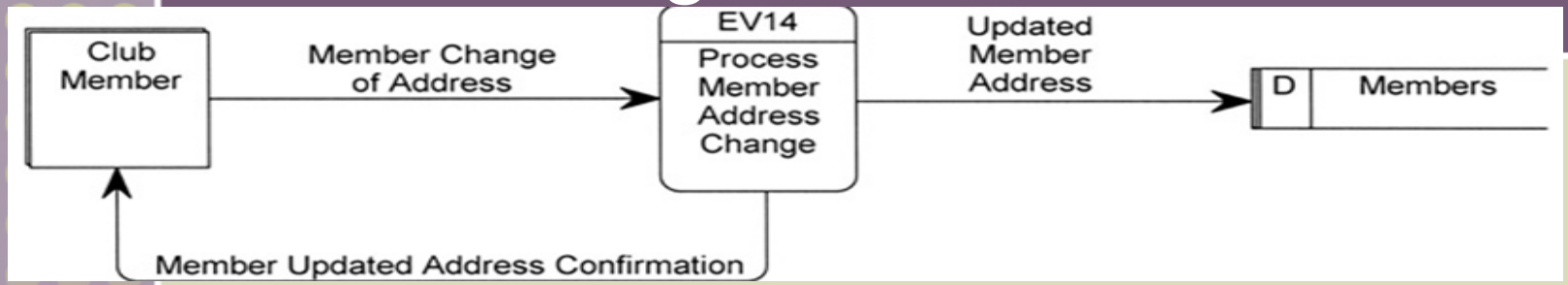
5. Event Diagram

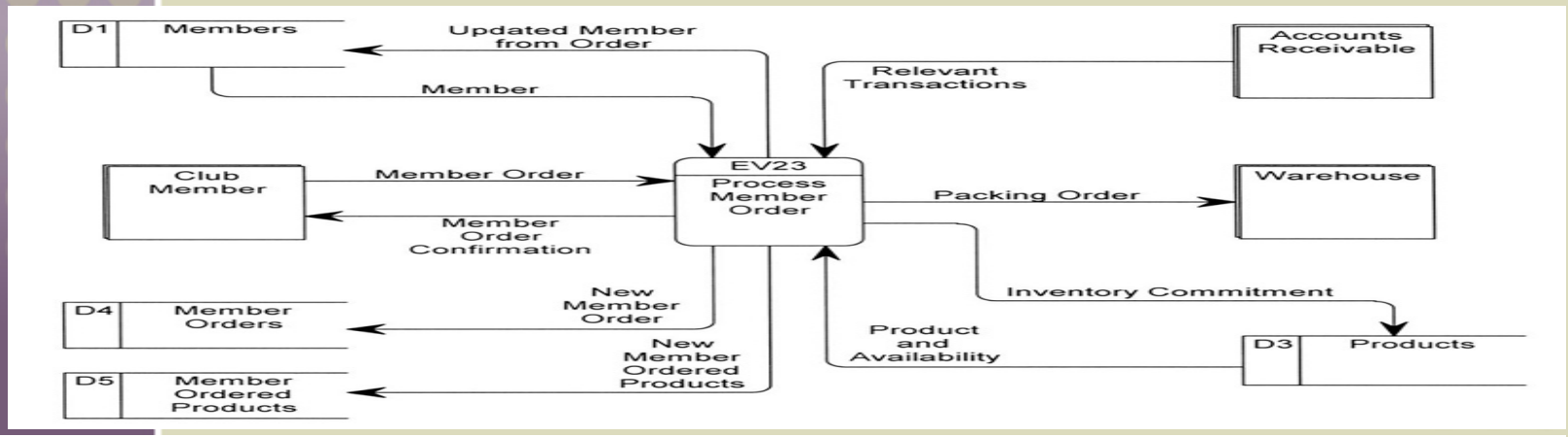
Event diagram – data flow diagram that depicts the context for a single event.

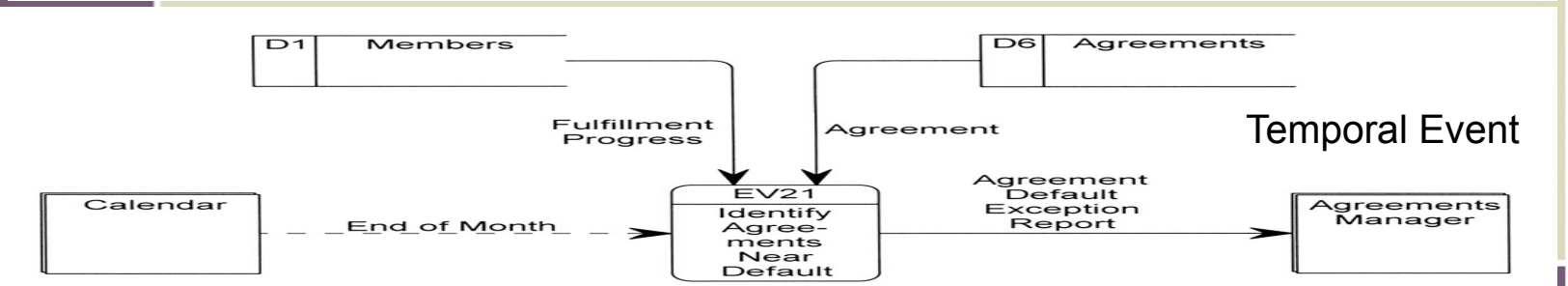
- One diagram for each event process
- Depicts:
  - Inputs from external agents
  - Outputs to external agents
  - Data stores from which records must be "read." Data flows should be added and named to reflect the data that is read.
  - Data stores in which records must be created, deleted, or updated. Data flows should be named to reflect the update.

A context diagram for a single event

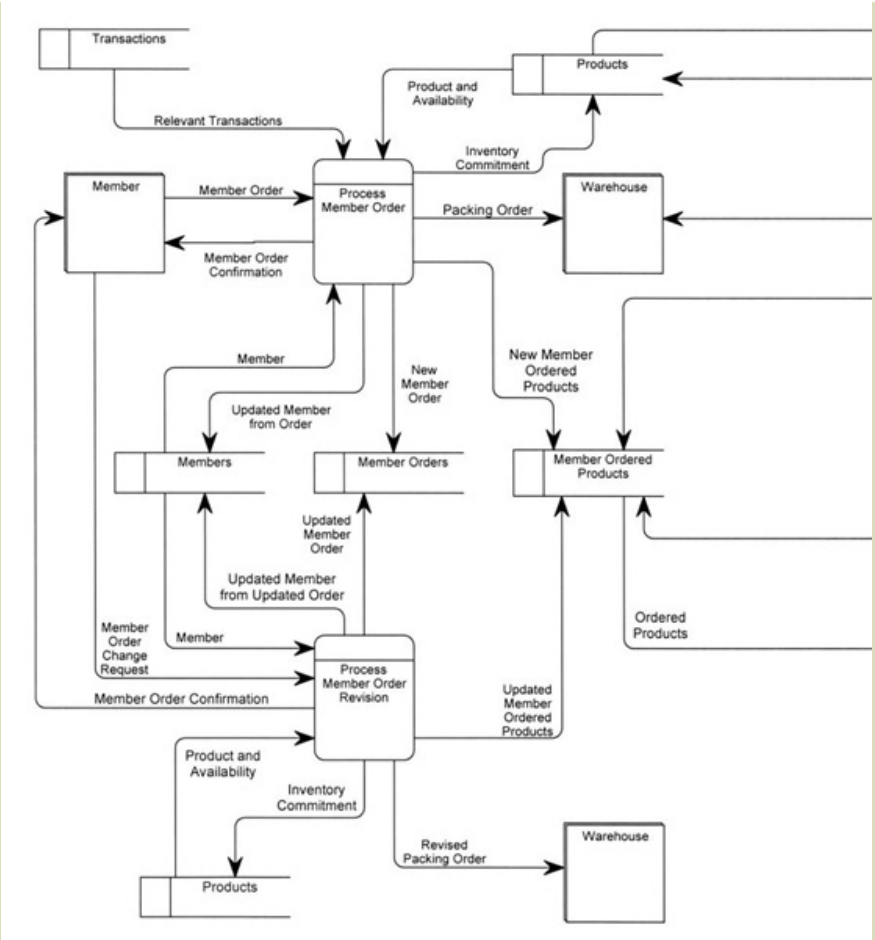
5. Event Diagram







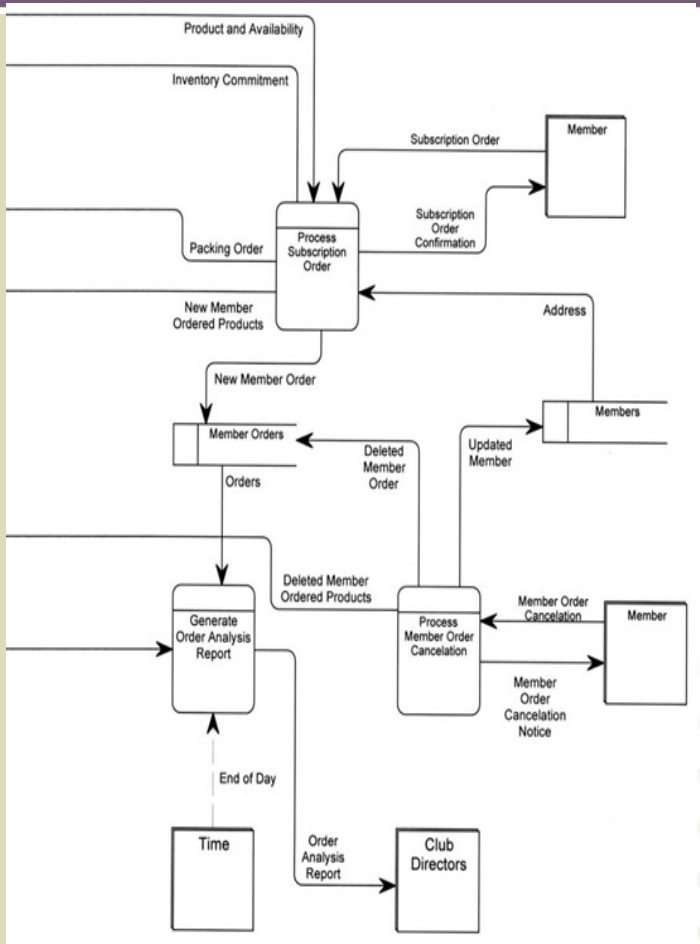
6. System Diagrams



6. System Diagrams

Balancing - a concept that requires that data flow diagrams at different levels of detail reflect consistency and completeness

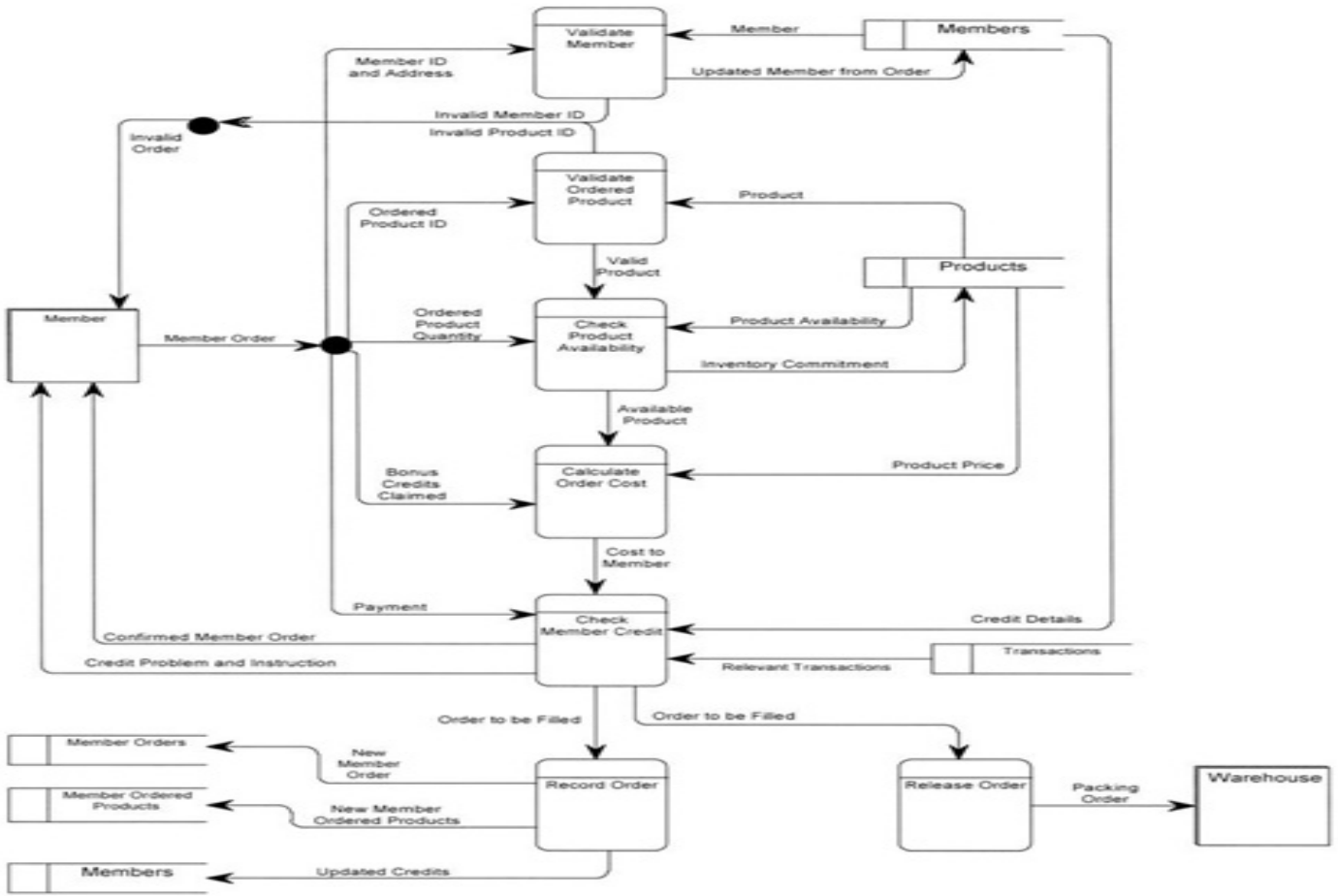
- Quality assurance technique
- Requires that if you explode a process to another DFD to reveal more detail, you must include the same dta flows and data stores



7. Primitive Diagrams

- Some (not necessarily all) event processes may be exploded into primitive diagrams to reveal more detail.
  - Complex business transaction processes
  - Process decomposed into multiple elementary processes
  - Each elementary process is cohesive - it does only one thing
  - Flow similar to computer program structure

7. Primitive Digrams
(see book for more readable copy)



8. Completing the Specifications

Process Logic:

- Data Flow Diagrams are good for identifying and describing processes.
- Not good at showing logic inside processes.

e.g. CHECK CUSTOMER CREDIT

- How to do it?
  - Flowcharts & Pseudocode - Users do not understand them.
  - Natural English - Imprecise and subject to interpretation.
- Structured English – A syntax for specifying the logic of a process.
  - Based on the strengths of structured programming and natural English.

Structured English Constructs

Construct	Sample Template
Sequence of steps – Unconditionally perform a sequence of steps.	[Step 1] [Step 2] … [Step n]
Simple condition steps – If the specified condition is true, then perform the first set of steps. Otherwise, perform the second set of steps. Use this construct if the condition has only two possible values. (Note: The second set of conditions is optional.)	**If** [truth condition] **then** [sequence of steps or other conditional steps] **else** [sequence of steps or other conditional steps] ~~**End If**~~
Complex condition steps – Test the value of the condition and perform the appropriate set of steps. Use this construct if the condition has more than two values.	**Do the following based on** [condition]: **Case 1: If** [condition] = [value] then [sequence of steps or other conditional steps] **Case 2: If** [condition] = [value] then [sequence of steps or other conditional steps] … **Case n: If** [condition] = [value] then [sequence of steps or other conditional steps] ~~**End Case**~~

Structured English Constructs

Multiple conditions – Test the value of multiple conditions to determine the correct set of steps.

Use a decision table instead of nested if-then-else Structured English constructs to simplify the presentation of complex logic that involves combinations of conditions.

***A decision table** is a tabular presentation of complex logic in which rows represent conditions and possible actions and columns indicate which combinations of conditions result in specific actions.*

DECISION TABLE	Rule	Rule	Rule	Rule
[Condition]	value	value	value	value
[Condition]	value	value	value	value
[Condition]	value	value	value	value
[Sequence of steps or conditional steps]	X			
[Sequence of steps or conditional steps]		X	X	
[Sequence of steps or conditional steps]				X

Although it isn't a Structured English construct, a decision table can be named, and referenced within a Structured English procedure.

One-to-many iteration – Repeat the set of steps until the condition is false.

Use this construct if the set of steps must be performed at least once, regardless of the condition's initial value.

Repeat the following until [truth condition]:
[sequence of steps or conditional steps]
~~**End Repeat**~~

Zero-to-many iteration – Repeat the set of steps until the condition is false.

Use this construct if the set of steps is conditional based on the condition's initial value.

Do while [truth condition]:
[sequence of steps or conditional steps]
~~**End Do**~~

- OR -

For [truth condition]:
[sequence of steps or conditional steps]
~~**End For**~~

Decision Table

Decision table – a tabular form of presentation that specifies a set of conditions and their actions.

Conditions and Actions	Rules
Conditions	Condition Alternatives
Actions	Action Entries

	Conditions and Actions	Rule 1	Rule 2	Rule 3	Rule 4
Condition Stubs	C1: Type of check	personal	payroll	personal	payroll
	C2: Check amount less than or equal to $75.00	yes	doesn't matter	no	doesn't matter
	C3: Company accredited by *LMART*	doesn't matter	yes	doesn't matter	no
Action Stubs	A1: Cash the check	X	X		
	A2: Don't cash the check			X	X
		Rules			

Decision Table

Conditions and Actions	Rules 1	2	3	4
Under $50	Y	Y	N	N
Pays by check with two forms of ID	Y	N	Y	N
Uses credit card	N	Y	N	Y
Complete the sale after verifying signature.	X			
Complete the sale. No signature needed.		X		
Call supervisor for approval.			X	
Communicate electronically with bank for credit card authorization.				X

Conditions and Actions	Rules 1	2	3	4	5	6	7	8
Customer ordered from Fall catalog.	Y	Y	Y	Y	N	N	N	N
Customer ordered from Christmas catalog.	Y	Y	N	N	Y	Y	N	N
Customer ordered from specialty catalog.	Y	N	Y	N	Y	N	Y	N
Send out this year's Christmas catalog.		X		X		X		X
Send out specialty catalog.			X				X	
Send out both catalogs.	X				X			

A. Data & Process Model Synchronization CRUD Matrix

Synchronizing of System Models

Data-to-Process-CRUD Matrix

Data / Entity . Attribute	Process Customer Application	Process Customer Credit Application	Process Customer Change of Address	Process Internal Customer Credit Change	Process New Customer Order	Process Customer Order Cancellation	Process Customer Change to Outstanding Order	Process Internal Change to Customer Order	Process New Product Addition	Process Product Withdrawal from	Process Product Price Change	Process Change to Product Specifi	Process Product Inventory Adjustment
										Process			
Customer	C	C			R	R	R	R					
.Customer Number	C	C			R	R	R	R					
.Customer Name	C	C	U		R		R	R					
.Customer Address	C	C	U		RU		RU	RU					
.Customer Credit Rating		C		U	R		R	R					
.Customer Balance Due					RU	U	R	R					
Order					C	D	RU	RU					
.Order Number					C		R	R					
.Order Date					C		U	U					
.Order Amount					C		U	U					
Ordered Product					C	D	CRUD	CRUD		RU			
.Quantity Ordered					C		CRUD	CRUD					
.Ordered Item Unit Price					C		CRUD	CRUD					
Product					R	R	R	R	C	D	RU	RU	RU
.Product Number					R	R	R	R	C			R	
.Product Name					R		R	R	C			RU	
.Product Description					R		R	R	C			RU	
.Product Unit of Measure					R		R	R	C		RU	RU	
.Product Current Unit Price					R		R	R			U		
.Product Quantity on Hand					RU	U	RU	RU					RU

C = create R = read U = update D = delete

Synchronizing of System Models

B. Process Distribution

Process-to-Location-Association Matrix

Process	Customers	Kansas City	. Marketing	. Advertising	. Warehouse	. Sales	. Accounts Receivable	Boston	. Sales	. Warehouse	San Francisco	. Sales	San Diego	. Warehouse
Process Customer Application	X					X			X			X		
Process Customer Credit Application	X						X							
Process Customer Change of Address	X					X			X			X		
Process Internal Customer Credit Change							X							
Process New Customer Order	X					X			X			X		
Process Customer Order Cancellation	X					X			X			X		
Process Customer Change to Outstanding Order	X					X			X			X		
Process Internal Change to Customer Order						X			X			X		
Process New Product Addition			X											
Process Product Withdrawal from Market			X											
Process Product Price Change			X											
Process Change to Product Specification			X	X										
Process Product Inventory Adjustment					X					X				X

Process

Location