OCD II
SSRD I

CS577a

Fall 2001

System Analysis

4. Proposed System (Analysis of)

- System Analysis involves several steps
  - Components - models, attributes, relationships, constraints, roles, and states
  - Behavior models
  - Engineering - Abstraction, enterprise class engineering
- This section is an overview of Analysis for OCD 4.0
- Details follow in later sections for SSAD

System Analysis

- The creation of precise, consistent description of a conceptual system in terms of its high-level components
- Description is within the organization domain, independent of implementation
- Analysis goes beyond simple checklists and pictures
- Analysis ties the domain description to the system design and implementation

Analysis Defined

- A separation of a whole into its component parts
- An examination of a complex system, its elements, and their relations
- A statement of such an analysis
- A method in philosophy of resolving complex expressions into simpler or more basic ones

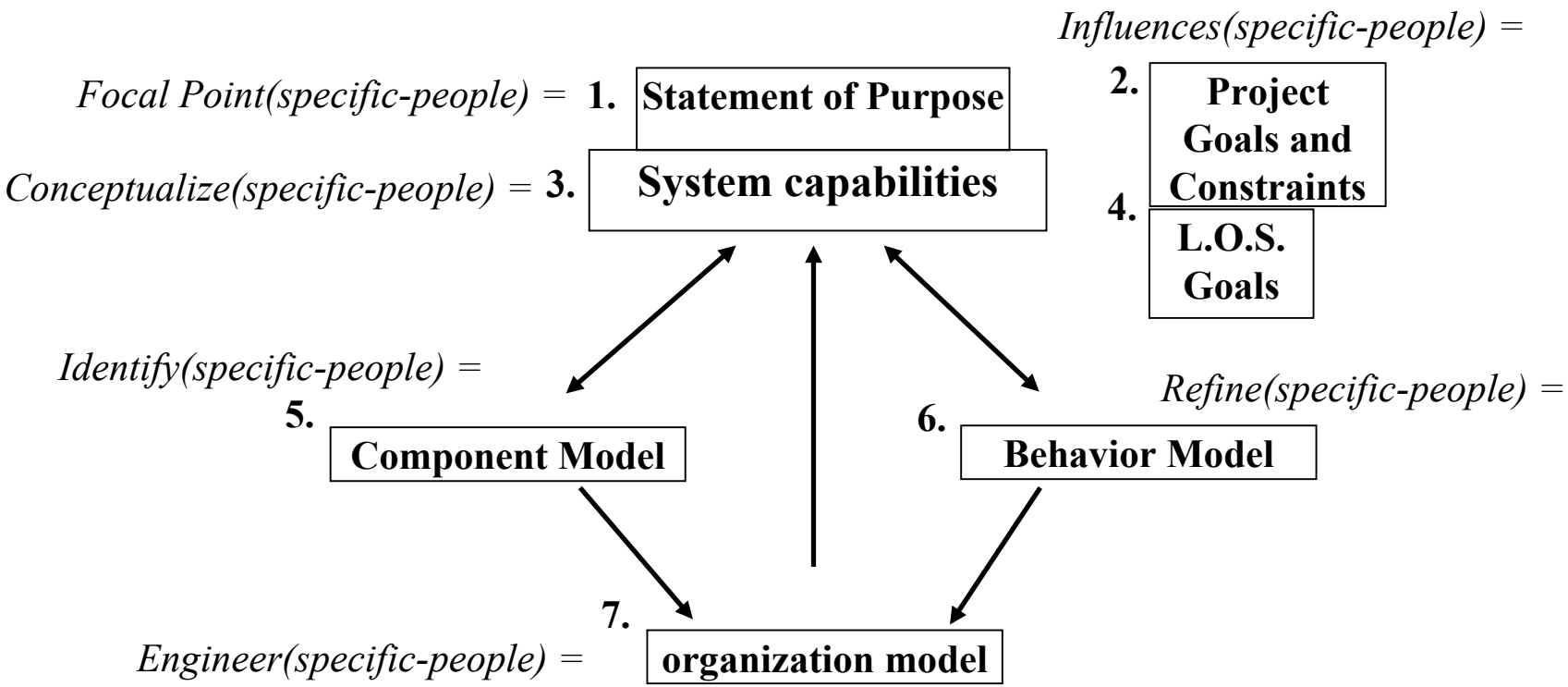
Analysis Goals

- Quantify *what* we want to represent, not *how* it is done
- Formalize and refine the specific parts of the organizations capabilities, entities, activities, and interactions described in the domain description that are to be automated
- Capture the high level architectural information that will represent (I.e. model) the conceptual system

Analysis Audience

- The Domain Description is for all constituents of the project
- Analysis is for Domain Experts - the high level leaders who understand the domain, know what they want, and have the authority to make decisions
- Not for implementers, who prefer design and implementation details (“hows”)

Analysis Models



Proposed System Description

Example
Vacation Sick Leave OCD

- OCD 2.2 Stakeholders: customer(HR manager), users (employees, supervisors, and HR), maintainers (ISD people), project manager, and developers. manager. Users include HR staff and 350 employees who are employed under ISD.

- OCD 3.4 Current System: utilized in the Human Resources(HR), is based on a current software system and a lot of paper work. It is used by staff to store and monitor vacation/sick leave of 350 employees. It is part of a University software system for payroll, personnel and benefits. Current system's user interface is menu driven textual mode system. It is accessible only by HR people. It does not provide systematic vacation leave processing. Modifications of vacation data (inserts, updates and deletes) require a lot of work that is redundant.

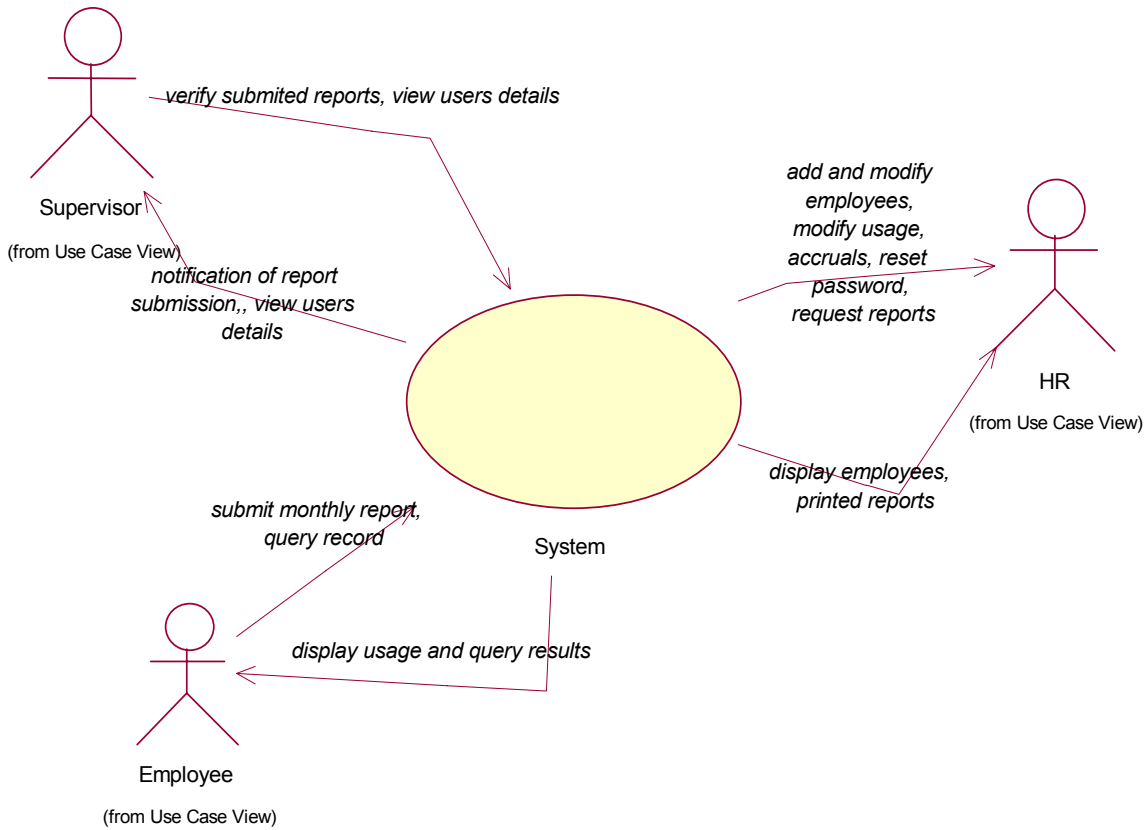
- OCD 4.3 Capabilities

  *The system will provide a web based service for employees and supervisors; Submissions will be done via web page, they will be able to post a query and receive results via web, and communicate and negotiate via email

  *The system should provide the search capability for administrators

  *The system should provide the users with help

OCD 2.3 System Boundary and Environment Context Diagram



4.3 System Capabilities

- Broad, high-level system behaviors.
- ***What*** the user should expect from the system
- Should be in line with Organization Goals (OCD 3.2) and Activities (OCD 3.3)
- Outline the desired modifications to the current systems entities and activities (OCD 3.3, 3.5)
- **"Just Do It"** approach. Work with what you know and flush out SR's later.

OCD 4.3 System Capabilities

- Relation to WinWin : priorities, rationale
- For each system responsibility, indicate:
  - Name
  - Description
  - Priority
  - Rationale
  - Reference to WinWin artifact [if one exists]
- Forward consistency with Capability Requirements (SSRD 3.)

Example: System Capabilities for the University Intranet

1. Provide distributed access to student admissions list (see OCD 2.5.1)
2. Manage asynchronous collaborations on admission evaluations (see OCD 2.5.4)
3. Provide auto-updated list of current research areas (See OCD 2.5.2)
4. Manage secure access to and from the intranet (See OCD 2.2.3)
5. ...

4.3 System Capabilities

- High level overview of broad categories of system behaviors
- Not an operational breakdown (provided by System Requirements)
- System capabilities realize high-level activities in the Organization Activity Model (Reference as appropriate).

4.3 System Capabilities (cont.)

- Describe a few system capabilities and work with domain experts to clarify them.
  - Think about “What in our domain description do we want represented with technology?”
  - Look at organization wide goals and consider what is required to carry them out.
- Each system responsibility may require several iterations: Consistency and redundancy are not issues at this point

4.3 System Capabilities(cont.)

- "Just do it" approach eliminates the pressure to get it all right on the first pass
  - "Go with what you know" and plan to iterate back through it.
  - Multiple iterations of SR's reduce complexity through balanced partitions, promote focus, remove redundancies and contradictions
- As more capabilities are documented, architects get a better idea of how the domain experts are viewing the proposed system
- Allow tangents to continue, even if they cross perceived system boundaries.

Example:
System Capabilities

4.3 System Capabilities

The system capabilities are the high-level system behaviors described according to broad categories of system behaviors.

Responsibility: SC-01
Name: Find/Search Material
Description: Once the archive is accessed the user needs to find the materials desired or if there are materials of interest.
Priority: Very High
Rationale: Once at the site the user must be able to search for an item in the archive. This is deemed a basic requirement.
Relates to: I.A [OCD 2.5]
WinWin reference: eballew-WINC-4, eballew-WINC-9, eballew-WINC-3

Responsibility: SC-02
Name: View Material
Description: Materials in the archive can be viewed using the system.
Priority: Very High
Rationale: This is basic required functionality for the customer and user.
Relates to: II.C.3 [OCD 2.5]
WinWin reference: eballew-WINC-6, eballew-WINC-7, eballew-WINC-16, eballew-WINC-2, eballew-WINC-13

Finding System Capabilities

- Ask Constructive Questions
- Avoid counter productive questions
- Don't obsess over issues that are addressed as part of the process (e.g. consistency, redundancy, completeness, soundness)

Constructive Questions

- Some constructive, informal questions that may be useful in gaining clarity (organized brainstorming)
  - "What does this mean?"
  - "Can you give me an example of this?"
  - "What do you need to do this?"
  - "What is involved with this?"

Construction Question 1

"What does this mean?"

Useful to define words and phrases

(document in CDL)

Construction Question 2

"Can you give me an example of this?"

Useful to draw out scenarios of desired system operations

Construction Question 3

"What do you need to do this?"

To find out information required to carry out a responsibility

Construction Question 4

- ***"What is involved with this?"***
- To discover sub-capabilities and the steps required to fulfill them
- This question is important as it is used to continue the process.

Counterproductive Questions

- Avoid Counterproductive Questions:
  - ***"Didn't we already cover this?"***
  - ***"How can we possibly implement that?"***
  - ***"Do we really need this?"***
- Begin by discovering what is actually wanted
- The modeling process answers these "save it for later" questions

Counterproductive Question 1

"Didn't we already cover this?"

☞ Don't worry about overlapping capabilities: these will be cleared up as we build the model

Counterproductive Question 2

"How can we possibly implement that?"

☞ Stay focused: this is not relevant to domain experts, only to implementers

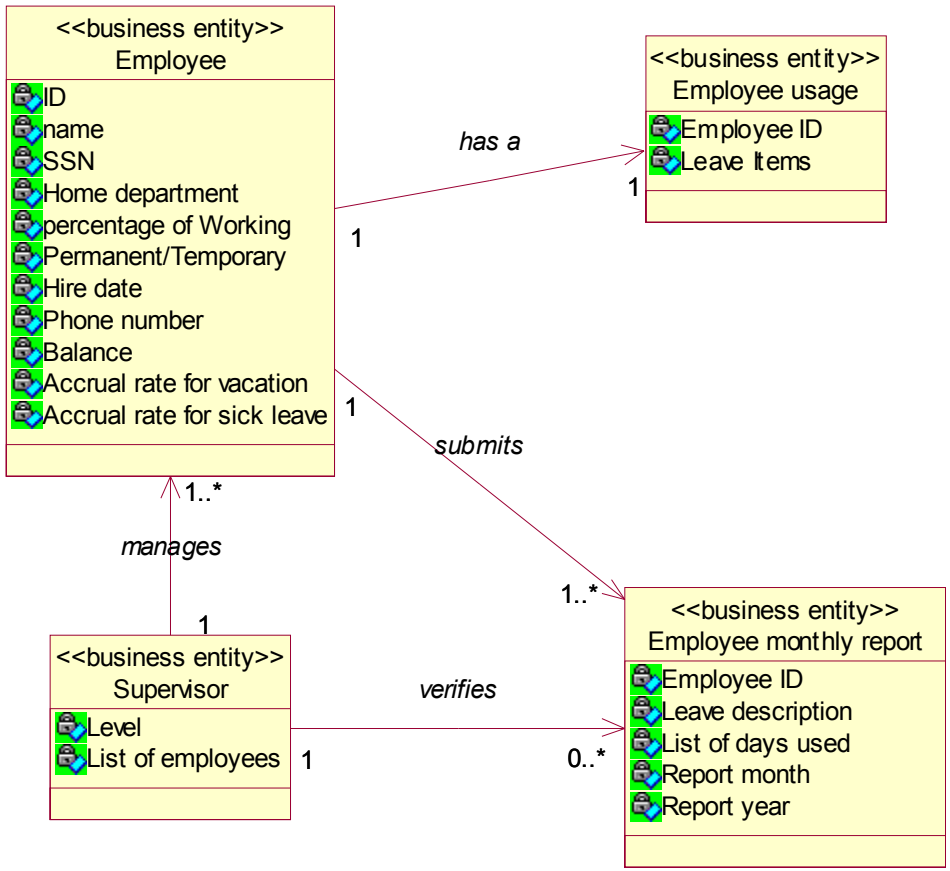
Counterproductive Question 3

"Do we really need this?"

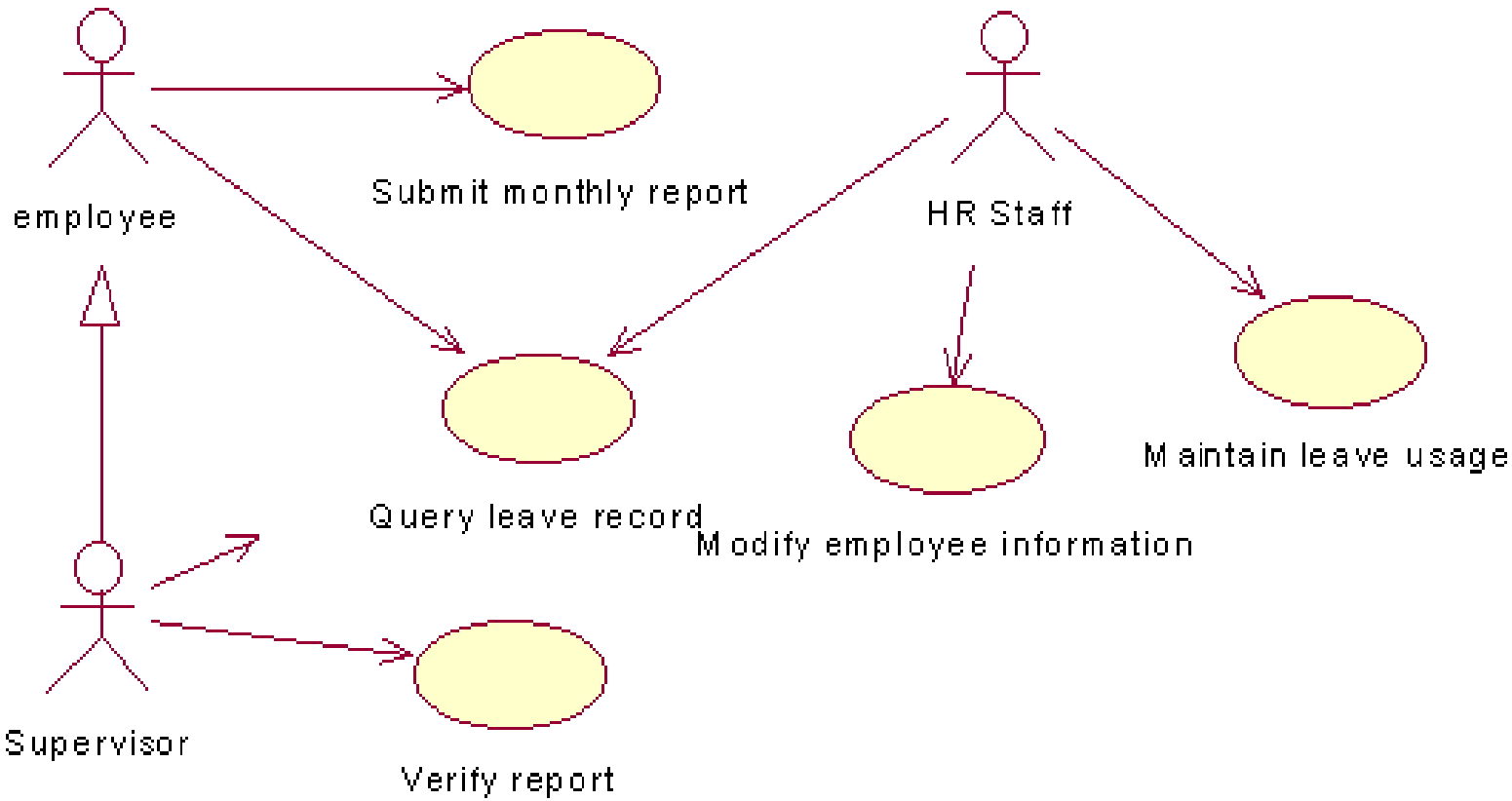
☞ Don't challenge feasibility or relevance:

relevance is determined by domain experts (you can send them the bill later!)

OCD 4.5.2 Proposed Entity Model



OCD 4.5.1 Proposed Activities



4.4 L.O.S. Goals

- Project Goals have global effects on the System capabilities
- L.O.S. Goals should be consistent with Win conditions and agreements
- M.R.S. (Measurable, Relevant, Specific): Initially, one may specify desirable and acceptable levels of capability
- Use a simple enumerated list

4.4 L.O.S. Goals (continued)

- Describe desired qualities of the System (i.e., "how well" the system should perform a given responsibility)
- Do not overburden the system's analysis with L.O.S. Goals that are not expressly requested by the customer.

4.4 L.O.S. Goals (continued)

Level of Service:	<<give a reference number and name>> such as “LS-1: Response time”
Description:	<<describe the level of service>>, such as “1 second desired; 2 seconds acceptable”
Measurable:	<<indicate how this goal can be measured with respect the specific elements it addresses – include as appropriate baseline measurements, minimum values, maximum values, average or typical or expected values, etc. >>, such as “time between hitting Enter and getting useful information on the screen”
Relevant:	<<describe which system capabilities (OCD 4.3) and perhaps project goals (OCD 4.2) this level of service is relevant to>>, such as “larger delays in order processing (see capability 3 in OCD 4.3) cause user frustration”
Specific:	<<describe what in particular within the system capabilities (OCD 4.3) and perhaps project goals (4.2) this level of service addresses>>, such as “credit card validation (in capability 3 OCD 4.3) may cause significant delay when attempting to connect to the verification service”

Example:
L.O.S. Goals

Example OCD 4.4 L.O.S. Goals

1. Simple and easy to use system

The system should be simple and attractive for the users who are searching and accessing the digital archive so that there is ease of access and usability for novice to expert computer users. For example, with regards to navigation there should only be a few steps for the user to get to the search. The main page will have only a brief description, a text entry only search, and a few links to keep the interface simple. The more advanced search will still be limited to 8 search or browse areas. The search results page will be a list of a maximum of 10 items for simplicity and ease of use. Also, 2 to 3 sample users will test the interface to ensure usability. [Organization Goal 2, 7, 8] [bhansali-WINC-6, bhansali-WINC-7, bhansali-WINC-14, bhansali-WINC-15, eballew-WINC-1, eballew-WINC-5]

2. Good Performance

The user desires quick access to the pages and speedy searches. A user only has so much patience, and the system should be usable or it will not be utilized, thereby limiting access. The wait for pages to load should be no longer than 15 seconds, and the search should take no longer than 10 seconds during peak usage times. The number of large graphics will be minimized, and slower modem connections will be used in testing. [Organization Goal 1, 2, 8] [nrm-WINC-7, eballew-WINC-8]

3. Ensure scalability

The archive should allow for increase in the volume of the resources, such as books, manuscripts, pamphlets, videos, etc. There are approximately 50,000 items to be archived. The system should scale from an archive size of 5,000 items to 50,000 items without any visible performance deterioration. The system should also scale from 5 to 100 concurrent users with the same performance. A significant increase in items may require a more powerful DBMS with faster searching and indexing capabilities. [Organization Goal 8] [bhansali-WINC-3, nrm-WINC-5, nrm-WINC-4]

Project Complexity Metrics

- We often want to know how big a project is or will become
- There are many popular ways to get these metrics
- A simple rough guess involves
  - Count Number of System capabilities
  - Adjust with L.O.S. Goals

Example: OCD 4.6 Redressal of Current System Shortfalls

Current System Shortfalls	Redressal of Current System Shortfalls
• Not Easy to Update	• The info base will require very low maintenance by the client. The client will be allowed to modify the info base by canceling courses, adding courses, changing the times that courses are offered, or changing someone's registration.
• Not Easy to Search	• The client will be able to generate various reports from the info base.
• Inefficient registration	• The registration process will be automatic, eliminating the need for client intervention and improving upon the sophistication of the current system. Email notification will be sent to the user confirming registration and notification will be sent to the instructor if a course is full or not enough users have registered.
• No Authentication	• The proposed system will have authentication both for the interface and for the info base.
• No Filtering System	• The system will offer a different set of courses to members of the faculty and to members of the staff and to members of the administration
• Interface Appearance	• The interface of the proposed system will follow user friendly guidelines and will have more color and graphics than the current system.
• Linkage to a database	• The proposed system will automatically transfer information from the web interface to the info base.
• Canceling Registration	• The user will be allowed to cancel registration for a class.

OCD 4.7.1 Stakeholder Operational Impacts

Stakeholder:	The customer - the instructor of the courses at Barnard
Activities Performed:	Her role is to clearly communicate her ideas as to what would make her current or a new system satisfactory to her needs. She must explain why the current system does not work, yet accept a balance between the ideal system and the practical system.
Usage Characteristics:	Our customer will frequently use the system, constantly generating reports and modifying, updating, and maintaining the info base. They are expected to have expertise in the use of the info base.

Stakeholder:	The users - Barnard faculty and staff
Activities Performed:	They will use the interface to register for computer training classes. They must declare what conditions will make the system most useful and more productive than the system currently in use.
Usage Characteristics:	The Barnard Community will frequently use the system to register for courses. They are expected to have no expertise, and their interface should be user-friendly.

Prototyping and the OCD

About the next slides

- Guidelines to describe models of prototypes within your project
  - Purpose of prototypes
  - Scope of prototypes
  - Who will participate
  - Use of tools
  - History and use
  - Results of use
- Does not address why to build or what to build (somewhat implicit)
- Uses a spiral approach (prototype development is highly iterative but must be disciplined to converge on desired results)
- Will need to refer to other models and external items

OCD 5.1 Objectives

- Describe the critical issues and risks that the prototype is attempting to resolve and the uncertainties that the prototype is trying to address

- **Common Pitfall:** One common pitfall when prototyping is to fail to describe the prototype from the perspective of the client. In particular, the prototype should be user-oriented, and should avoid abstracting elements. It helps to use realistic sample data in the various prototype screens. E.g., use ‘Scrabble’, ‘Monopoly’, ‘Clue’, as opposed to ‘Item 1’, ‘Item 2’, ‘Item 3’.

OCD 5.2 Approach

- Describe the type of prototypes , the stakeholders who will participate in prototyping efforts, and the development tools used.

- **5.2.1 Scope and Extent**
  - Describe the type of prototypes (mock-up, functional, etc.) built and how they address the objectives stated in OCD 5.1
  - Explain the degree of faithfulness to the proposed system each prototype is expected to have.
  - Describe the extent that each prototype is expected to contribute to the implementation of the proposed system.

- **5.2.2 Participants**
  - Describe any participation on the part of the clients in the prototyping effort: e.g., changes requested after initial evaluation
  - Describe how effective was the prototype in overcoming initial IKIWISI (I'll Know It When I See It) client expectations

OCD 5.2 Approach (Cont.)

- **5.2.3 Tools**
  - Describe briefly the tool used to develop the prototype and the reasons for choosing that tool.
  - Describe how adequate the tool turned out to be to your needs, or whether you are contemplating using a different tool
    Example: "We started by creating a Web based prototype. But we decide to move to Microsoft Access since the system does not require public access and will be used only at the reference librarian desk".

- **5.2.4 Revision History**
  - Mention whether this is the first prototype, or a revised one, including changes suggested by client, etc...
  - Keep a simple Version Control history for the prototype, independent of the one for the overall OCD

5.3 Initial Results

- For each aspect of the system that you prototyped, describe the:
  - **Current way of performing activity**
    Example: "Currently, orders are entered via phone, email, or fax without interactive confirmation of price and availability."
  - **Proposed way of performing activity**
    - Include screen shot of relevant prototype screen
    - Brief explanations on how system will be used as illustrated by prototype screen (You may annotate explanations directly on screen shots)
    - You may propose multiple screens, and indicate which one your client preferred (or maybe hasn't decided yet which one to use).
- **Example**:
  - *Home page*: Client is provided company and new-specials information, and is asked for name, account number, and indication of user type: consumer, corporate, or dealer (see screen image).
    Search Page: Client is offered the option of a single keyword search of all fields, or a more complex search (see screen

5.4 Conclusions

- List by order of priority the items that you will be looking into next, during the next round of prototyping

- List the most critical risks that you hope to resolve by doing further prototyping

- **Example**: "Current prototype suffers from navigability problems: we will be looking into improving the usability and the navigability using frames, site maps, etc."

System and Software Requirements Definition (SSRD) and Requirements

Purpose of SSRD

- **Describe capability requirements** (both nominal and off-nominal): i.e., the fundamental subject matter of the system, measured by concrete means like data values, decision-making logic and algorithms.
- **Describe Level of Service Requirements** (sometimes referred to as Non-functional requirements): i.e., the behavioral properties that the specified functions must have, such as performance, usability, etc. Level of Service Requirements should be assigned a unit of measurement.
- **Describe global constraints**: requirements and constraints that apply to the system as a whole. For example, the customer for the system is a global constraint, as is the Purpose of the System. Those constraints include: Interface Requirements, Budget and Schedule Requirements, Implementation Requirements
- **Mandates and instructions on how the system must be implemented** ("must", "shall", "will"), with respect to the general

SSRD Purpose in MBASE

Life cycle objective

- Top-level functions, interfaces, quality attribute levels, including:
  - Growth vectors
  - Priorities
- Stakeholders' concurrence on essentials

Life cycle architecture

- Elaboration of functions, interfaces, quality attributes by iteration
  - Resolution of TDB's (to-be-determined items)
- Stakeholders' concurrence on their priority concerns (prioritization)
- Traces to SSAD (and

Overall Description

- Intended audience
  - Implementers
  - Domain expert (decision makers) for system definition
  - No architecture description elements (e.g., Sequence/collaboration): those belong to the SSAD
- Participants
  - Same stakeholders as WinWin negotiation

Dependencies

- SSRD depends on WinWin taxonomy
  - Outline of SSRD evolves from taxonomy
  - There is no one-size-fits-all taxonomy or requirements description
  - Importance of adapting taxonomy to domain
  - Agreed upon Win conditions and priorities become reqs
  - Options describe “how” for reqs.
- SSRD depends on OCD:
  - Statement of Purpose
  - Project Goals and Constraints
  - System Capabilities

Dependencies (Continued)

- SSRD depends on FRD
  - Changes considered but not included
- SSRD depends on prototype for:
  - User/system interface requirements
  - Nominal (feature) requirements
- Additional documents depend on SSRD:
  - SSAD to obtain (and consistency trace):
    - System and Project Requirements
  - FRD to check for satisfaction of:
    - All requirements

Requirements

- Defines the system concept (from OCD) with respect to general technology considerations
- “Must”, “Shall”, “Will” instructions for implementers
- An assurance contract for the customers
- Necessarily a top-level design activity
  - ties OCD to SSAD with respect to FRD
  - allows planning within LCP
  - provides an outlet from WinWin
  - provides tangible means of high-assurance through testing and inspections to meet IOC completion criteria
  - not a good way to start a project
- Very misunderstood and abused

Main Kinds of Requirements

- Project Requirements (SSRD 2)
  - global to project, affects overall system requirements
- Capability Requirements (SSRD 3)
  - local to system, specific system functionality
- System Interface Requirements (SSRD 4)
  - varies, affects groups system requirements
- Level of Service Requirements (SSRD 5)
  - local to system, may affect many system requirements
- Evolutionary Requirements (SSRD 6)
  - varies, effects design and implementation

Necessary Condition

- All requirements must be testable and implementable (subject to risk considerations)
  - There must be some way to demonstrate that a requirement has been satisfied by the system (will be documented in FRD)
  - **System Capability:** either supports or does not support a typical or non-trivial scenario (Use-Case)
  - **Project:** must have a measure, what is being measured, definition of satisfactory
  - **Level of Service:** must have a measure, specific instances with respect to capabilities, satisfactory threshold (relative measures are useful)
  - **System Interface:** must specify checklist for all interface parameters
  - **Evolutionary:** must refer to a design or implementation scenario that supports a possible future satisfaction

SSRD 2. Project Requirements

- General assumptions and constraints placed upon the design team
  - If not met, stakeholders would not be satisfied or would not accept system
- Describe non-negotiable global constraints: e.g., solution constraints on the way that the problem must be solved, such as a mandated technology
- Refine Project Goals (OCD 4.2)
- Should be M.A.R.S. (Measurable, achievable, relevant, specific)

SSRD 2. Project Requirements (cont.)

- Budget and Schedule
  - Development and transition time
  - Cost limits for development, transition and support
- Development Requirements
  - As appropriate include
  - Tools and Programming Languages
  - Computer Resources
  - Standards compliance
- Packaging Requirements
  - Installation, post- installation, delivery

SSRD 2. Project Requirements (cont.)

- Implementation Requirements
  - Personnel and staffing
  - Training
  - Development environment including hardware and software
- Support Environment Requirements
  - Tools required
  - Personnel and skills required

Example
Vacation Sick Leave SSRD

- SSRD 2.0 Project Requirements

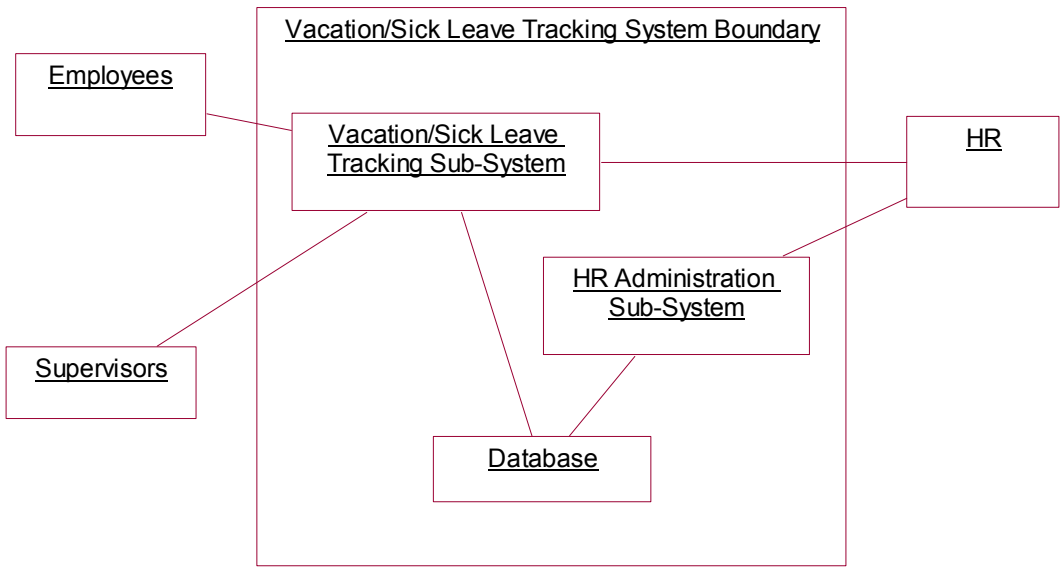
Budget: The system requires $8350 for transition cost, there is not hardware and software cost for this system. $400/month for maintenance cost and $1316/month for operational cost. (see LCP 2, 5; FRD 2.1)

Schedule: The system is to be completed within 24 weeks. The first 12 weeks are used to complete system Life Cycle Objectives (LCO) and Architecture (LCA), which includes system operational concept (OCD), prototype, system requirements (SSRD), system and software architecture (SSAD), life-cycle plan (LCP), feasibility rationale (FRD) and WinWin negotiations. The second 12 weeks are used to implement and deliver the system. (refer to LCP 2,3,4,5)

SSRD 3. Capability Requirements

- System Definition
  - Brief overview of the proposed system including the major components of the system
  - Refined from Statement of purpose (OCD 4.1
  - Provide a block diagram using UML Collaboration diagram
  - Do not repeat from the OCD

SSRD 3.1 System Definition



SSRD 3.1 System Requirements

- System requirements should be a refinement of Capabilities (OCD 4.3)
  - Refinement of the Capabilities (OCD 4.3)
  - Define the nominal and the off- nominal cases
  - Nominal cases represent typical system conditions
  - Off- nominal cases handle exceptional and abnormal conditions.
  - Off- nominal requirements indicate the required robustness.
  - During LCO include the most important ones
  - Add more requirements before LCA

Example System Requirements VSL

- The subsystem would provide two levels of access: employee (staff or faculty) and Supervisor. The system checks authentication and then displays different web pages and performs different functions for different roles.
- Each user inputs and submits his/her monthly Vacation/Sick Leave report
- User can query his/her vacation/sick leave history records
- The supervisor reviews the monthly Vacation/Sick Leave reports submitted by the employees in his/her department.
- Supervisor can request system to show a summary report which lists the employee' usage and balance of leave information.
- When the user wants to input the date into Monthly Report, the calendar will pop-up to help user choose the date.
- Etc….

SSRD 3.1 System Requirements (cont.)

- System Requirements
  - Prioritize the requirements based on the WinWin negotiations
  - Every capability requirement should be testable
  - Modes and user classes possible
    - E. g. Operational mode vs. Training mode
    - Administrators vs. Surfers
  - Use structured Use- case diagrams with attached Activity diagrams where the actions and events exceed what can be explained in 3 sentences

Example of System Modes

- A multimedia archive system may operate in the following modes:
  - *User mode:* when users access the archive (database opened in shared mode)
  - *Administrator mode:* when the administrator is modifying the archive (database opened in exclusive mode)
  - *Maintenance mode:* when the administrator is performing repair, backup or compacting operations; (database is shutdown)

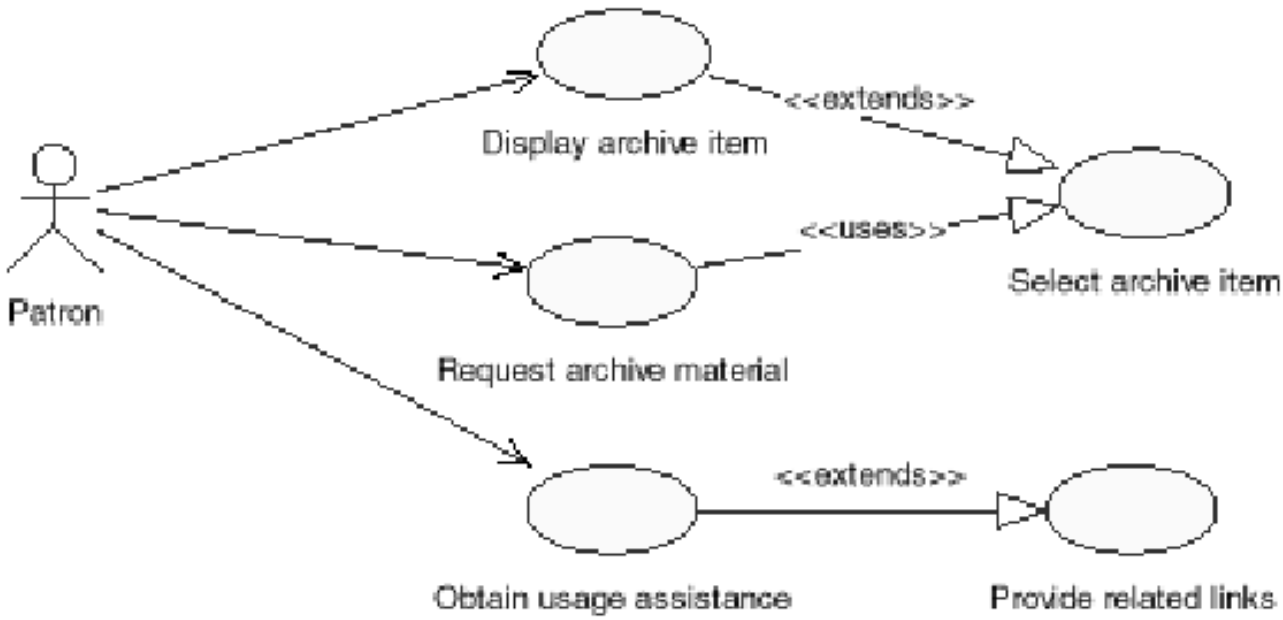
Requirement Specifications

- Number
- Name
- Description
- Priority
- Rationale
- Constraints
- Dependencies
- Risks
- Traceability: reference to WinWin artifact and System Responsibility (OCD)
- Inputs
- Actions
- Events
- Interactions
- Sources
- Outputs
- Stimuli
- Destinations
- Pre-conditions
- Post-conditions
- Side effects
- Use case diagrams (optional)

Example of Nominal Requirement

Requirement:	RQ-13
Function:	The user subsystem allows the user to view information on archive items.
Description:	The Archive user subsystem allows the user to view the list of archive items, select the item of interest, deselect if required and view the overview on the selected archive items.
Input(s):	- Selected archive items - The database with the overviews of the archive items.
Source(s):	User
Output(s):	Overview display of the archive items.
Destination(s):	User Display
Pre-condition(s):	The user has performed a search by keyword or has browsed the archive.
Post-condition(s):	The user either makes an advance request or starts another search or exits from the system.
Relates To:	[OCD SR-02]
WinWin Agreements:	[nrm-AGRE-7]
Mainstream Scenario:	[OCD 5.2.1 SC-03]
Exception Handling Scenario:	[OCD 5.2.1 SC-07]

Example Use Case Diagram



SSRD 4. System Interface Requirements

- User Interface Requirements
  - Explain various required user interfaces if applicable
  - GUI, command line, textual menu, diagnostics and logs
- Hardware Interface Requirements
  - Describe interfaces to special hardware such as scanners
- Communications Interface Requirements
  - Interface with network devices part of the system
- Software Interfaces
  - APIs used and provided

SSRD 5. Level of Service Requirements

- Describe desired and required qualities of the system
- "How well" should the system perform
- More specific than Levels of Service (OCD 4.4) and explain how they are achieved
- Provide MARS Criteria
- Specify the unit of measurement.
- Provide desired and acceptable levels
- FRD should validate how the architecture can meet the level of service requirements

Example Levels Of Service

- Dependability
  - Reliability
  - Availability
- Usability
  - Ease of learning
  - Ease of use
- Performance
- Maintainability
- Portability
- Inter-operability (or binary portability)
- Reusability

Example Level Of Service

3.4.1 Response Time

Requirement:	QR-08
Specific:	Search Engine's response time to user queries should not be more than 10 seconds
Description:	The HDA system should be able to respond to the user search by keyword within 10 seconds, eg. Searching the database for all the items that have the keyword in the title, author name, descriptor, and notes.
Relevant:	If the system does not respond to a search within a short time the user's interest would wane gradually. This could affect the usability of the system adversely. This is only relevant to SQ-2 (search for items in archive)
Measurable:	500 searches that return a uniformly randomly distributed set of up to 100 items must be conducted wherein it should be verified if the system can respond within 10 seconds for each search for a local T1 connected client.
Achievable:	The search requirement (SQ-2) will be implemented using the IBM digital library standard query functions. The attributes title, author name, descriptor, and notes must be contained in the same table. See Feasibility Rationale FRD 2.2.3.4 for the feasibility of this option.

Example Feasibility (for FRD)

Sample FRD achievability assessment (FRD 2.2.3.4):

2.2.3.4 QR-8 Response Time

The support of heterogeneous servers in IBM digital library allows you to use the system for all kinds of data, while optimizing the processing of individual data types. This would in turn reduce the response time for a search request. Also, the support for multiple, distributed object servers allows digital objects to be placed close to the users who need to access these objects frequently. This helps in delivering large multimedia objects (like images) fast. The specifications for V.2.3 of the IBM digital library package indicates that the query rate is average of 10,000 rows per second for a single attribute query. The query is over a maximum of four attributes (title, author name, descriptor, and notes), so the query rate is no more than four independent queries over these attributes (or 10,000/4 = 2,500 rows per second) minus the time to find intersections (for "and" searches) which is O(n^2). Thus in 10 seconds 25,000 rows (the items) can be returned and compared for intersections giving approximately 150 items. The T1 transmission can deliver 1MB/s and each item is less than 500 ASCII characters long, so transmission is not an issue. Thus it is feasible that QR-8 can be achieved.

Poor Example

- M: The system should be as fast as possible
- R: The system should be available 24/ 7 even if organization does not support activities beyond day time
- S: The system shall be implemented as per the standards laid out by USC
- A: "The system shall be available 100% of the time" for a unreliable network- based system

SSRD 5. Evolution Requirements

- Describe required levels of flexibility and expandability
- Identify foreseeable directions of system evolution
- Describe the maintenance of software and data assets
  - Facilities and equipment
  - Maintenance levels and cycles
  - Emergency software support

SSRD 6. Evolution Requirements

- Description of the foreseeable directions of the system growth and change
- Description of how the software and data assets will be maintained
  - Facilities
  - Equipment
  - Service-provider relations
  - Maintenance levels
  - Maintenance cycles
  - etc...

SSRD 6.1 Capability Evolution

- Major post-IOC capability requirements
- Changes considered but ceferred
- May be used to “risk manage” system requirements with respect to project and level of service requirements
- More than a “feel good” place holder - must be accounted for within architecture, must also be testable

SSRD 6.2 Interface Evolution

- How must the system adapt to interface changes in the future?
  - Organizational changes in use on system
    - Personal changes (more, less, different style)
    - New or expanded product lines
    - Policy changes
    - Organization restructure
    - New/additional/dissolved relationships
  - External systems
    - New/additional/replace system
    - Changes in external interfaces

SSRD 6.3 Technology Evolution

- Describe how system adapts to future releases of external and COTS software
- Future technology change adaptation
- Workload evolution

SSRD 6.4 Technology Evolution

- Environment and Workload Evolution
  - Projected workload and data storage characteristics
  - Performance evolution

Examples of Capability Evolution

1. Data input using Voice Recognition

One proposed feature that has not been kept in the system design is one of voice recognition. In future, when voice recognition is available the system should allow the operator to simply speak the data that has to be entered and the database should be accordingly modified. For this an interface has to be provided.

2. Different levels of access to the archive (subscription)

In future, different levels of access to the archive items could be provided to the user. This would allow a set of users to have a wider access to items that could not have been archived otherwise. Also, the Boeckmann Center would get funds for its maintenance.

3. Full Browse Capability

This would have been an enormous additional task. The implementation of this could be a project on its own. It was not a requirement of the customer so it was not even considered in the WinWin negotiations.

SSRD 7. Common Definition Language for Requirements

- Definitions of unfamiliar terms, and acronyms encountered or introduced during the requirements elicitation process
- Do not repeat the common definition language for the domain description (will make it harder to ensure consistency)
- SSRD should be understood by everyone in the target audience

Example:

Context- related help: This help describes the help for a given context. For example, this kind of help would describe a screen, its contents, and its use.