Week 7: Chapter 10

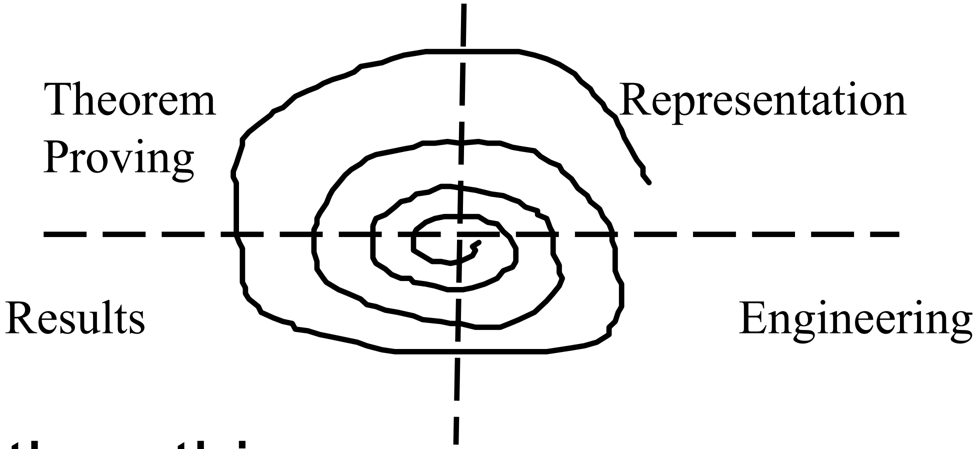
Knowledge Elicitation – Converting Tacit Knowledge to Explicit

Chapter Objectives

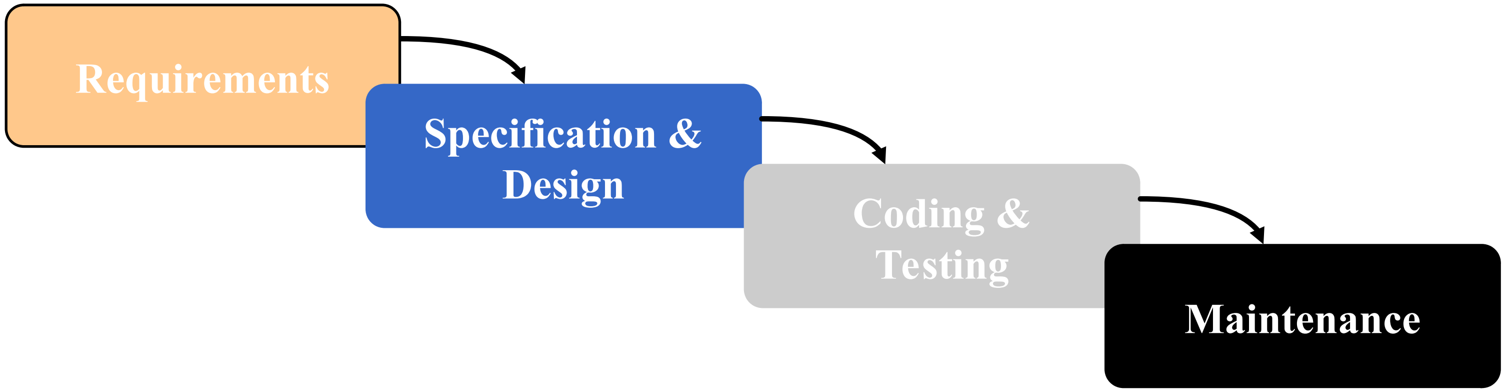
- Introduce the student to the knowledge-based systems life cycle
- Introduce the student to capturing tacit knowledge from human sources and convert it into explicit knowledge.
- Introduce the student to the various stages of the traditional one-on-one interview and how they can be managed for effectiveness.
- Other elicitation techniques such as observation, role-reversal, etc.
- The variations of the one-on-one interview when more than one person participates.

AI Lifecycle

- AI projects typically follow a spiral lifecycle like this :



- rather than this :



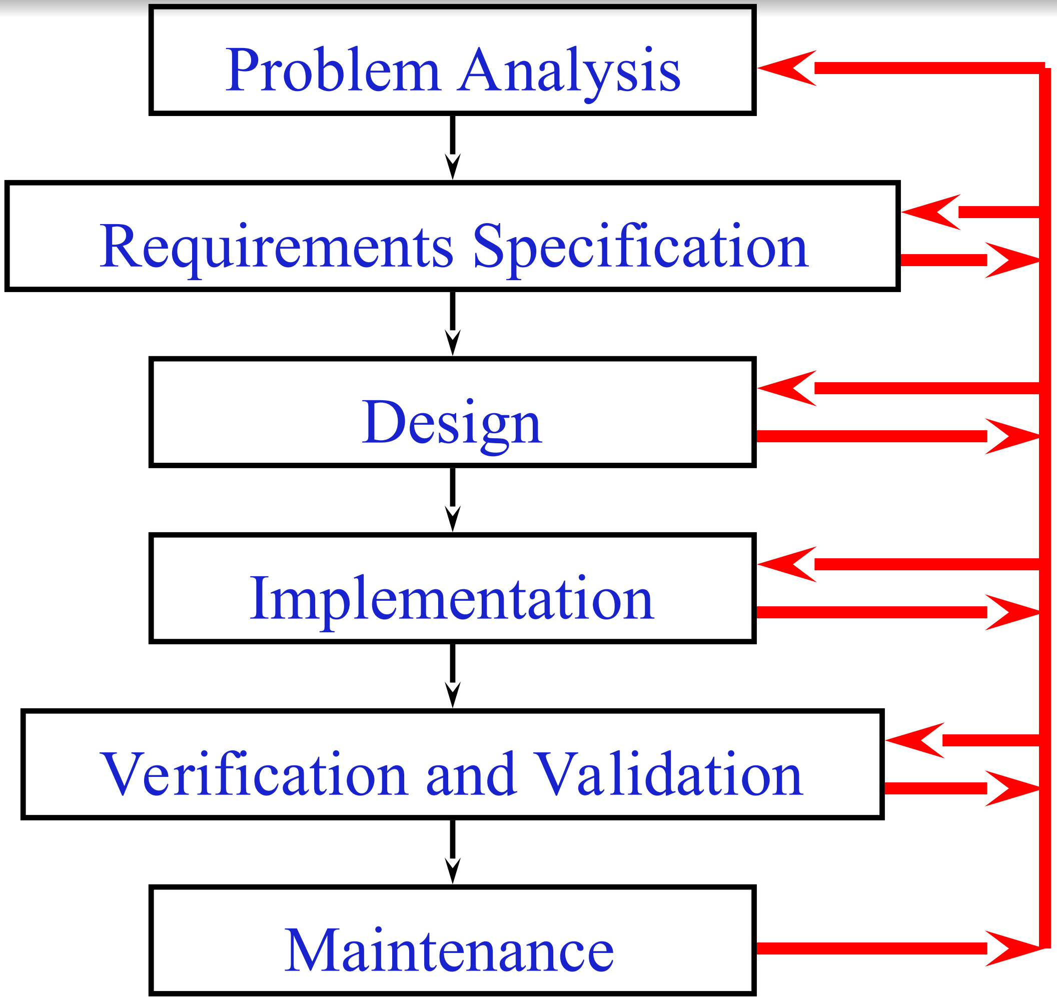
AI Lifecycle (Cont'd)

- Differences with traditional systems design methodologies
    - capturing *knowledge*, not data;
    - specialists in the field 'know best' - heavy use of *heuristics*;
    - empirical research within AI attempts to show evidence of intelligent *behaviour* 'by any means necessary'…

Key difference :

- knowledge elicitation - people issues (individuals, teams, organisations).

Traditional Software Lifecycle VS AI Lifecycle



Identify the problem characteristics

Find concepts to represent knowledge

Design structure to organize knowledge

Formulate "rules" to embody knowledge

Test knowledge

Update knowledge

Differences From Knowledge-based Systems

- Type of Representation (Data VS knowledge)
  - SE: well-known & well defined algorithmic procedures.
  - KE: extensive, imprecise, and ill-defined heuristic knowledge that needs to be transferred from minds of experts to a computerized representation (through *knowledge Acquisition, i.e. externalization*).
- Nature and quantity of knowledge
  - SE: can be estimated reasonably well
  - KE: not well known even by experts themselves. This leads to paradigm shift (*rapid prototyping & incremental Development*)

Knowledge-based System Lifecycle Model (*Incremental Development*)

1. Problem analysis
2. Requirements specification
3. Preliminary design
4. Initial (rapid) prototype
5. Final (detailed) design
6. Implementation
7. Verification & validation (V&V) or testing
8. Design adjustment
9. Maintenance

Problem Analysis

- Determine the important problem features
  - What is the problem
  - Who are the participants
  - What are the required resources
  - What are the goals and objectives

Problem Analysis (Cont.)

Suitability of the Application-*Ask the right questions!*

1) does a problem really exist?

- If not, this may indicate a **failure of the technology**
- What is needed is a **real problem** looking for the **solution**.

2) is a knowledge-based approach suited?

- Is human knowledge *being* represented?
- Is this knowledge *heuristic* or algorithmic?
- Does the knowledge *change* or remain constant?
- Is the expertise *well* understood?
- Are the inputs *complete* and *correct*?
- Can the problem be solved using *other methods*?

3) is a knowledge-based approach justified (cost/ benefit)?

Problem Analysis (Cont.)

Do you have the required resources?

Management support:

- Time
- Tools and training
- Expert available

Do you have the support of the expert?

- Cooperative

Is the expert competent?

- True expert
- More than one source of expertise

Is the expert articulate?

- Ability to communicate

Is the expert in close physical proximity?

- Expert is physically distant

Requirement Specification

- Decide what concepts, relations, and control mechanisms are needed
- Address the problem of knowledge granularity
- Avoid a complete characterization before starting the implementation
- You need to allow for the possibility of change!

Requirement Specification (Cont.)

Develop a document which:

1. Describes the problem
2. Sets goals
3. Addresses needs, desires, and concerns of the end users
4. Address issues of verification and validation here!

Requirement Specification (Cont.)

Suggested Document Template:

1. Introduction
   - Problem overview
   - Application domain
   - Project goals
2. Functions
   - System outputs
   - System inputs
   - Auxiliary features
   - Implementation priorities
3. Constraints
   - External interfaces
   - Compatibilities with other products:
     - speed of execution
     - reliability
     - maintainability
     - security
     - error identification
4. Misc. Issues
   - validation method
   - verification method
   - documentation
   - others

Preliminary Design

- Determine the structure and formulation of the key concepts
- Start forming ideas about what would be an appropriate tool

Preliminary Design (Cont.)

Select or determine:

1. Knowledge representation paradigm
2. Reasoning scheme
3. Tool
   - Does it support the above needs?
   - Flexibility
   - Special requirements (uncertainty, graphics …)
   - Auxiliary features (editor, trace, v&v tools …)
   - Performance
   - Vendor support
   - Cost
4. Development team
   - Knowledge engineer
   - Team leader
   - Expert

Initial (Rapid) Prototype:

- Test the design structure
- Illustrate the feasibility of the approach
- Make the initial system *deep* to show it can solve complex problems yet *narrow* in that it solves a very limited number of problems
- This should take 2 to 4 weeks to develop

Detailed (Final) Design:

- Modify the design based on the findings of the initial prototype
- Use modified structured charts (MSC) to show organization of system
- Use knowledge diagrams to show the structure of the knowledge
- Use design document templates

Detailed (Final) Design Guidelines:

- Suggested Document Template:
  - Each component of the knowledge model is mapped to its implementation primitives.
  - Knowledge representation schemes are mapped to the implementation details.

- General Knowledge Engineering tips :
  - Write rules initially in structured English (easy to read and understand by *everyone*).
  - Organize similar rules together (e.g. class hierarchy).
  - Place most likely rules first.
  - Refrain from "Proceduralizing" the Knowledge base (use OO concepts wherever possible !)
  - Visualise decision paths (frames, semantic nets).

Concepts, Objects, or Frames

- The design should describe all the concepts and its properties
- Values of a properties should be specified in detail
  - **Value type:** nominal, integer, real, etc.
  - **Multiple or single:** accepts either single or multiple of values
  - **Value source**: rules, functions, tables, database, user, etc.
  - **Constraint:** e.g. Upper limit & lower limit
  - **Possible values:** legal values in case of nominal
  - **Prompt:** if input from user
- Example

Concept	Property	Type	S/M	Source	N	UL	LL	Possible values	Prompt
Leaf observations	color	nominal	M	user	Y			spotted, yellow, brown, mosaic	What is the color of the leaves ?

Implementation (1)

Turn design into a working system

This requires:

- Content: domain knowledge
- Form: an implementation language
- Integration: combining and reorganizing to eliminate mismatches

Implementation (2)

There are four ways to implement the system:

1) using automatic knowledge acquisition tool

2) using expert system shells, e.g. CLIPS.

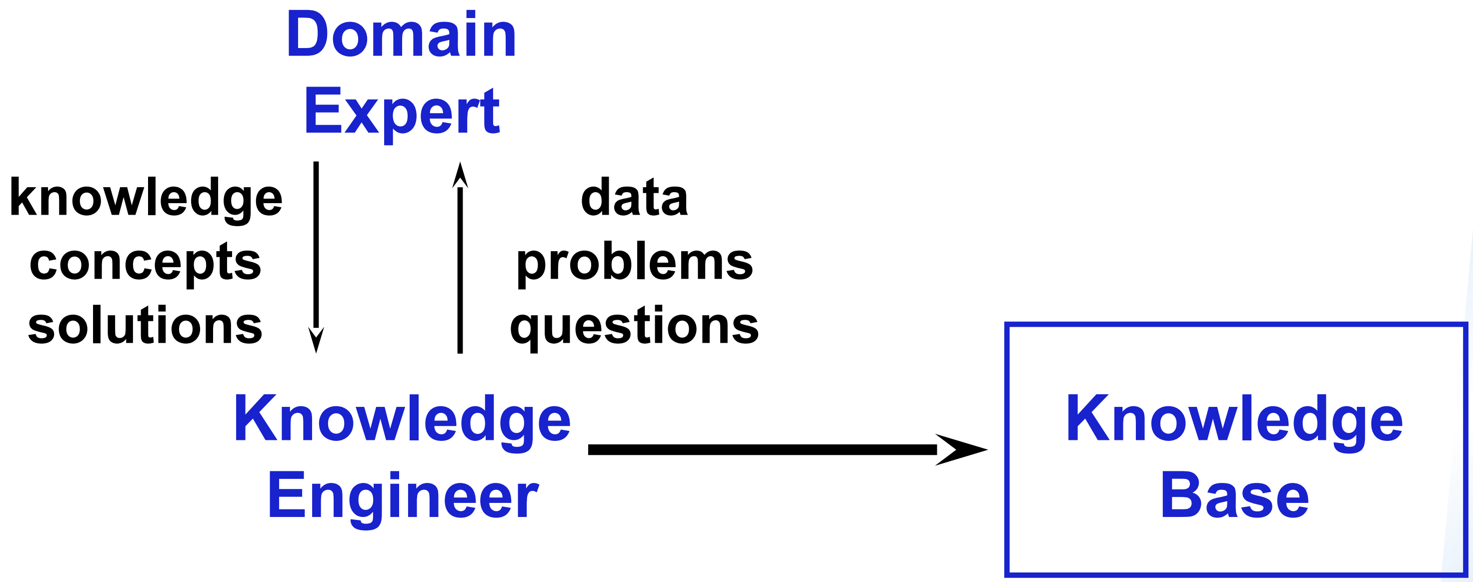
3) using a computer programming language that supports artificial intelligence applications development, e.g., LISP and PROLOG.

4) using other computer programming languages, e.g., SMALLTALK and C++.

tacit knowledge VS explicit knowledge: Knowledge acquisition perspective

- By definition, explicit knowledge is already captured in an understandable form.
- We need to elicit tacit knowledge and then capture it in a form that makes it easily manageable.
  - Most *commonly used* knowledge elicitation process is a manual one—interviews
  - In addition, *partially automated methods* exist for facilitating the capture of domain-specific knowledge

Knowledge Acquisition (1)



Discussion

What is your understanding of a human expert? How would you select one?

- An expert is someone who has spent a number of years or thousands of hours diagnosing or troubleshooting a particularly difficult problem and is known to be good at solving it.
  - This goes from the seasoned and licensed auto mechanic to the open heart surgeon.
- A human expert is a knowledgeable person(s) who provide(s) the knowledge for the knowledge management system. S/he need not know anything about Knowledge-Based Systems – and generally does not.
- …

Discussion (cont'd)

What is your understanding of a human expert? How would you select one? (cont'd)

- S/he must be willing to cooperate in having his knowledge transferred to such a system: a) has time; b) does not feel threatened.
- Desirable traits:
  - competence, articulate, self-confidence, availability, open-minded, personable and enthusiastic … etc.

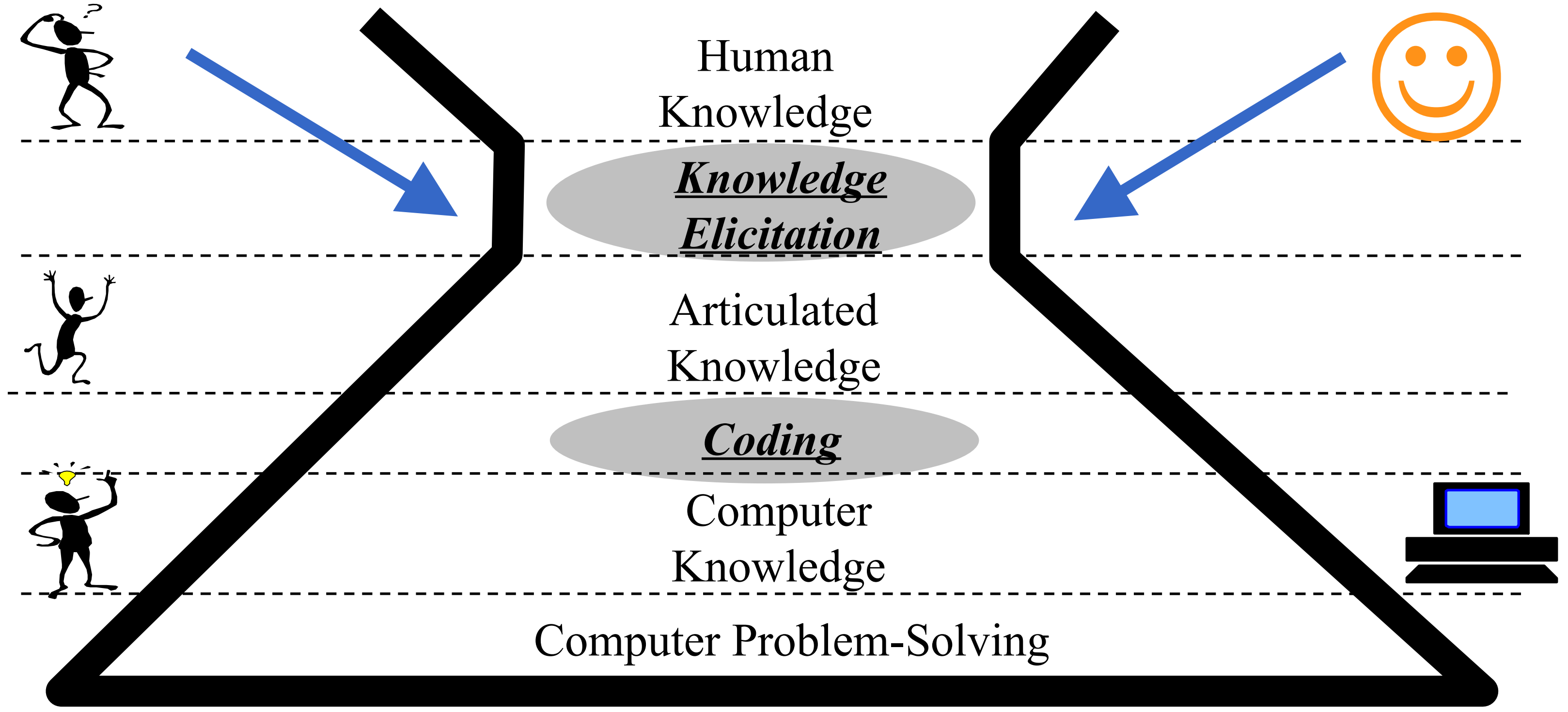
Knowledge Acquisition VS Knowledge Elicitation

- *Knowledge acquisition* is the process of extracting knowledge from whatever source including document, manuals, case studies, etc.

- *Knowledge elicitation* is a type of the knowledge acquisition where the **only** knowledge source is the domain expert.
  - It is face-to-face discussion between the subject matter experts (SMEs) who possesses the domain knowledge, and KEs who ask questions, observe the expert solving problems, and **determine what knowledge is used**.

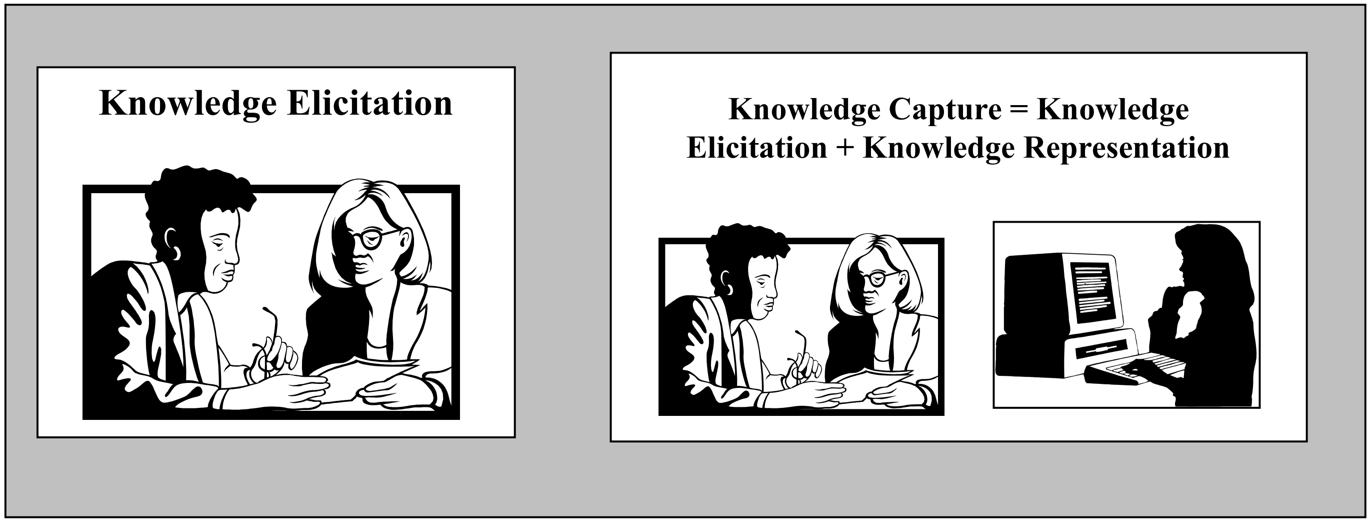
Knowledge Elicitation

- This is the primary *bottleneck* in developing a system
- One of the major difficulties is to explicitly identify and capture knowledge relevant to the intended application



Knowledge capture

- *Knowledge capture* is the combination of knowledge elicitation and representation of the knowledge in a machine-readable form.
- Iterative (ongoing) process during the KBS development—incremental development.
- Continually refines and 'purifies' knowledge base.
- The most crucial prerequisites to knowledge capture is to find a qualified expert and selecting a problem domain.


Knowledge Elicitation
Knowledge Capture = Knowledge Elicitation + Knowledge Representation

Example of Knowledge Elicitation Document

Elicitation Report	
Date:	Knowledge Engineer:
Session#:	Topic:
Location:	Source:
Start time :	End time :
Type: [ ] Interview [] Protocol analysis [ ] Concept sorting [ ] Other...................	
Session Goals: Session Summary: Rules derived :	

Interview

- Need the willing cooperation of the domain expert
  - Obstacles (status difference, age differences, differences of interest)
- Selecting an expert
  - Desirable traits: competence, articulate, self-confidence, availability, open-minded, personable and enthusiastic … etc.
- Preparation-- *the KE homework!*
  - Digesting the knowledge obtained during the last session.
  - Transfer knowledge into representation (diagrams, pseudo code, initial implementation) for testing ideas

Types & objectives of Interviews

1) Kickoff (initial) interview-*the plan of attack*
   - identifying exact function of the proposed ES
   - identifying the End-users
   - studying the domain background

2) General knowledge-gathering (unstructured) interview sessions
   - not planned sessions; used early in the lifecycle
   - used in the first knowledge acquisition stages for identifying and understanding the problem

3) Specific problem-solving, knowledge-gathering (Structured) interview sessions
   - planned sessions (goals & questions)
   - time scheduled
   - used in all phases of the lifecycle to clarify or extend information received via other techniques

Kickoff (initial) interview-the plan of attack

- Objective:
  - establish good rapport (relationship) with the expert
- Agenda (max 1 hour)

1. Introduction and light conversation
2. Explanation of what the objectives of the elicitation process are–incremental development
3. Discussion of the importance of the project
4. Discussion of what is expected of the expert as well as what the expert can expect from the KE
5. Identification of what reading materials the expert recommends for the KE to review to become familiar with the domain
6. Scheduling of the next meeting

General knowledge-gathering interview sessions

- Objectives for KE:
  - learn general principles about the domain from the expert
    1. Better understand the subject matter (domain)
    2. Better understand the expert's opinions and viewpoints on the domain
- Uses open-ended questions: the expert talks freely and the KE gains insight into the expert's tacit KB.
  - require discussion
  - cannot be answered simply with a yes, no, simple term, or number
- Characteristics:
  - Knowledge gathered probably will not be explicitly expressed
  - Relieves some of the burden from the expert, by not requiring a continual definition of every term used
  - Wide-ranging, emphasizing breadth of coverage

Specific problem-solving, knowledge-gathering interview sessions (1)

- Objective
  - explore how the expert solves specific problems or answers questions in the domain
- Selecting a subarea that can serve as the first chunk of knowledge to be extracted from the expert and represented explicitly.
- Subareas should involve issues or areas that are:
  1. Well understood by the particular expert interviewed
  2. Reasonably well understood by the KE
  3. Of sufficient breadth and depth to truly represent the difficulties of this domain
  4. Small enough to require 2 to 3 months of development effort without trivializing the scope of this domain's problems

Specific problem-solving, knowledge-gathering interview sessions (2)

- Characteristics:
  - Highly directed, emphasizing depth instead of breadth of coverage
    - Particular goals and questions in mind, usually derived out of previous sessions.
    - Knowledge Engineer needs strong advance understanding of topic.
  - Knowledge gathered probably will be explicitly expressed using the system’s knowledge representation language
- KE uses close-ended questions:
  - are quite specific questions
  - Can be answered simply with a yes, no, simple term, or number

Discussion (20 min)

- Select a topic for an expert systems project, assume you have done:
  - Kickoff (initial) interview-*the plan of attack*
  - General knowledge-gathering (unstructured) interview sessions
  - Specific problem-solving, knowledge-gathering (Structured) interview sessions
- Give examples of the results of each of which.
- This will be presented after 20 min to your colleagues and will be a subject of discussions

Knowledge elicitation techniques

- Output-input-middle method
- Observational elicitation method
- Role reversal method

Output-input-middle method

- Knowledge elicitation follows the sequence: **Output-Input-Middle method**

Output

- Identify the answers or solutions to the problem under discussion (goals)
- KE should focus on understanding subtle differences between goals

Input

- Identify the sources of information that the expert uses to deduce the solution/answer
- KE should make sure how these inputs are identified, determined, or generated is known and understood

Middle?

- Determine the links between the inputs and outputs
- These connections represent the core of the expert’s knowledge
- Some inputs may not be required initially, but may be requested later after the initial inputs are interpreted
- Intermediate goals/hypotheses may be required to complete the connections

Discussion - Give Example (15 minutes)

- Fruit classification Knowledge Base
- Output:
  - Fruit type: banana, watermelon, honeydew, cantaloupe, apricot, orange, cherry, peach, apple, or, plum
- Input:
  - Fruit characteristics: Diameter, shape, seed count, and color
- Middle: This is the knowledge that directly or indirectly matches the inputs to the outputs. The middle represents the rules that will allow the user of the knowledge to determine the fruit type.

Middle (rules/Knowledge) Fruit Classification

- Shape & Color determine fruit
  - IF Shape = long and Color = green or yellow THEN Fruit = banana
- Shape & Diameter determine fruit class
  - IF Shape = round or oblong and Diameter > 4 inches THEN Fruitclass = vine
- Seed Count determine Seed Class
    - IF Seedcount = 1 THEN Seedclass = stonefruit
- Fruit Class & Color determine fruit
    - IF Fruitclass = vine and Color = green THEN Fruit = watermelon
- Fruit Class & Surface & Color determine fruit
    - IF Fruitclass = vine and Surface = smooth and Color = yellow THEN Fruit = honeydew
- Fruit Class & Color & Seed Class determine fruit
    - IF Fruitclass = tree and Color = orange and Seedclass = stonefruit THEN Fruit = apricot

Some weakness of Basic Interview

- The Q&A interview is not always the most efficient means of eliciting knowledge from an expert
- In some domains, considerable expertise is documented in instruction manuals or books
  - eg, maintenance manuals for automobile diagnosis
- Sometimes even cooperative experts have difficulty articulating their expertise
    - *Articulate: express your ideas or feelings in words*
- Other elicitation techniques can be used when appropriate
  - Observational elicitation
  - Role reversal
  - Process:
    - Expert is given initial information about the problem.
    - Expert is asked to solve the problem “out loud”.
      - Observational study : expert “walk-through” of solution process pitfalls and heuristics.

Observational elicitation method

- Asking expert to report on or demonstrate his decision making process for specific problem
- KE observes the expert at work and trying to understand and duplicate the expert’s problem-solving methods.
- Provides an indication of the process the expert uses to solve a particular (real) problem
- Observation types (next slide):
  - Quiet on-site observation
  - On-site observation with discussion
  - Role reversal method

Quiet on-site observation

- KE can not question (interrupt) the expert while they work (solving a real problem)– sometimes called protocol analysis
- Pros
  - Experts' train of thought is not continually interrupted by questions, so they can proceed at their most effective and realistic form
- Cons:
  - Lack of interaction leaves the KE wondering about the solution approaches taken by the expert—not good at for obtaining details about the process
  - If expert is asked to talk out loud as they work, can make experts self-conscious causing them to alter it or to create a verbalization that is much more or less complex than what they are actually doing
- Should be used:
  - to get a feel for the total magnitude of the problem-solving process
  - to verify (or reject) that a hypothesized approach is in use
- Should not be used:
  - to obtain details about the process
- Q&A session should follow

On-site observation with discussion

- KE may interact with the experts while they work (solving the real problem)
- Pros
  - Permits KE to better probe the process observed
- Cons
  - The expert may become distracted by the questions and not follow the normal procedure
    - develop a line of reasoning that does not reflect his/her actual approach
- Should be used:
  - when the observed task does not significantly challenge the expert's problem-solving abilities (eg, is fairly routine)
- Should not be used:
  - when the expert needs to struggle to reach a solution
  - Symptoms:
    - uneasiness,
    - hesitation in decision-making: Some experts do not like to be observed
    - uncooperative or refusing to create a solution in front of the KE: Fear of 'giving away' expertise is a concern
- Q&A session should follow

Role reversal method (KE <-> DE)

- KE attempts to become the expert (pseudoexpert)
- The pseudoexpert attempts to solve a problem in the presence of the true expert (role-playing)
- The true expert questions the pseudoexperts about what they are doing and why
  - Like the observation process, but the with roles reversed
- May be used when:
  - KE already has a significant understanding of the problem-solving process
  - KE wishes to verify correctness of understanding
- Can clarify, modify, and provide significant new knowledge not previously uncovered by the KE

Team Interviewing

Explain briefly how knowledge is elicited via teams?

1. One KE and multiple experts (one-on-many)
2. Multiple KEs and one expert (many-on-one)
3. Multiple KEs and multiple experts (many-on-many)

One-on-many Interviews: One KE and multiple experts

- Good when
  - Common when several experts work closely together
  - Each expert may be specialized in slightly different areas (complement each other)
  - Cooperation toward a common goal
- Cons
  - Sometimes the experts do not get along; can undermine team's productivity
  - Can be redundant especially in general knowledge-gathering sessions, which is wasteful of experts' time
  - Inexperienced KEs may be overwhelmed by multiple experts
  - Even experienced KEs may be exhausted quickly, since the KE must maintain concentration while each expert can drift in and out of "high gear"

Many-on-one interviews: Multiple KEs and one expert

- Good when
  - Multiple observers; multiple sets of eyes and ears are better than one to a clearer picture
  - A senior KE explaining tasks to junior KEs (implementers)
  - Training a junior KE on KA
- Cons
  - The single expert often feels overwhelmed by the multiple KEs–may become more defensive –cooperating in KA may exhaust the expert before any one of the KEs
  - Little chance for synergism, since no one else present has the expert’s level of domain understanding
  - implementation details may confuse the expert

Many-on-many interviews: Multiple KEs and multiple experts

- Good when
  - Discussing political issues
  - may realize the benefits of both one-on-many and many-on-one interviews –synergism between experts as well as multiple observer perspectives
- Obstacles
  - Harder to accomplish the task
  - High redundancy is wasteful of experts' and KEs' time

Handling Problem Experts

- The wimp expert
  - Wimp experts fear the loss of their job or status within their organization.
  - They believe that the KBS is being developed to perform their functions
- The KE must convince the expert the KBS is being developed to free him for other more pressing tasks
  - the advantage of human expert over artificial expert

Handling Problem Experts (Cont.)

- The cynical expert
  - Cynical experts hate their job, despite their boss and detests their organization.
  - They do not want to leave for some reason, e.g. Salary, seniority, benefits.
- The KE must
  - Carefully examining all provided knowledge
  - Showing sympathy for the expert
  - Appealing to the expert’s professionalism.

Handling Problem Experts (Cont.)

- The high priest of the domain expert
  - KE wastes his time, daring to thing that he can be replaced with a machine.
  - KBS can never approach his level of competence
  - He is never available for meeting and unwilling to do any tasks for the project between meetings.
- The KE must his confidence through the development of a competent prototype.

Handling Problem Experts (Cont.)

- The paternalistic expert
  - Paternalistic expert and KE is a kind of a *teacher-student* relationship.
  - He tends to talk too much, telling stories revolving around his achievements.
  - He can be very beneficial if he handled very well
- The KE must
  - Diplomatically tries to minimize the expert's discourse, attempting to turn him into useful information.
  - A sense of humor can achieve the desired effect

Handling Problem Experts (Cont.)

- The uncommunicative expert
  - Uncommunicative experts speak short sentences and do not provide elaboration on their answers.
  - He tends to talk too much, telling stories revolving around his achievements.
  - He can be very beneficial if he handled very well
- The KE must
  - Examine answers with great attention.
  - Study the domain deeply

Handling Problem Experts (Cont.)

- The uncaring expert
  - Uncaring experts never disagree with anything with the KE because it takes too much time to explain.
  - This may be due to a lack of interest to the project.
- The KE must
  - Attempt to verify expert's knowledge to a limited extent.
  - Use a recording devices (if he agrees) may force him to think more about the answers given.

Handling Problem Experts (Cont.)

- The pseudo-AI expert
  - Pseudo-AI experts has read an article or two about KBS and thinks he knows it all.
  - He is dangerous when he tries to get involved in the details of the system which is the domain of the KE.
- The KE must
  - Attempt to gently let him know who is and who is not the KE.

Conclusions

- The student should be familiar with:
  - How to conduct a one-on-one interview with an expert to elicit her knowledge.
  - Alternative techniques for knowledge elicitation and when it is appropriate to use them.
  - Tools that can facilitate the knowledge elicitation process from an expert.
  - Techniques to automate the knowledge capture process from electronic databases.

Chapter 10

Knowledge Elicitation – Converting Tacit Knowledge to Explicit