Requirements Engineering:

Requirements Elicitation
(Basics of RE, Elicitation I, Elicitation II)

Steve Easterbrook
University of Toronto

Lecture 2: Basics of RE

Last Week:
INTRO
Syllabus
Course Goals
Literature on RE

This Week:
Basics of RE
RE in the lifecycle
Types of problem domain
Processes, methods & techniques
A little systems theory

Next Week:
Elicitation (I)
Traditional approaches
Interviews & Questionnaires
Prototyping

Importance of RE: background

- Problems
  - Increased reliance on software
    - E.g. cars, dishwashers, cell phones, web services, …
  - Software now the biggest cost element for mission critical systems
    - E.g. Boeing 777
  - Wastage on failed projects
    - E.g. 1997 GAO report: $145 billion over 6 years on software that was never delivered
  - High consequences of failure
    - E.g. Ariane 5: $500 million payload
    - E.g. Intel Pentium bug: $475 million
- Key factors:
  - Certification costs
    - E.g. Boeing 777: >40% of software budget spent on testing
  - Re-work from defect removal
    - E.g. Motorola: 60-80% of software budget (was) spent on re-work
  - Changing Requirements
    - E.g. California DMV system

Solutions??

- No "Silver Bullet" *(Brooks)*
  - Software is complex for its size
  - Software is invisible and abstract
  - No fabrication step; hence software is "modifiable"(!)
- But: Early modeling and analysis is important
  - Defects are cheaper to remove the earlier they are found (Boehm)
  - Requirements defects are more likely to be safety-related (Lutz)
    - E.g. Voyager and Galileo
- Early modeling and analysis is not enough
  - Need to communicate requirements to everyone
  - Need to seek agreement from all stakeholders
  - Need to understand the context for the system
  - Need to understand the context for the development process
  - Need to keep up to date as the requirements evolve

Refs: *Brooks 1987, Boehm 1981, Lutz 1993*

Basic Definitions

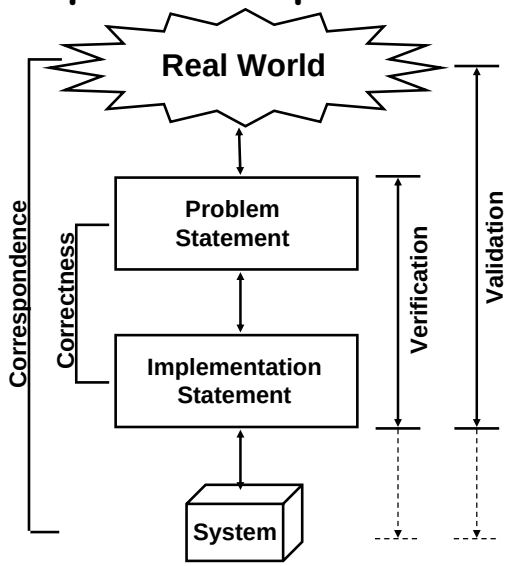
- **What is a Requirement?**
  - Something that someone needs to solve a problem or achieve an objective:
    - *"A condition or capability that must be met or possessed by a system or system component to satisfy a contract, standard, specification, or other formally imposed document. The set of all requirements forms the basis for subsequent development of the system or system component".* [IEEE Std]
- **What is Requirements Engineering?**

"...Requirements Engineering is the branch of systems engineering concerned with real-world goals for, services provided by, and constraints on software systems. Requirements Engineering is also concerned with the relationship of these factors to precise specifications of system behaviour and to their evolution over time and across system families..." [Zave94]

"... RE is concerned with identifying the purpose of a software system, and the contexts in which it will be used. Hence, RE acts as the bridge between the real world needs of users, customers, and other constituencies affected by a software system, and the capabilities and opportunities afforded by software-intensive technologies." [RE'01 CfP]

The essential requirements process

- **Understand the problem**
  - elicitation, requirements acquisition, etc.
- **Formally describe the problem**
  - specification, modelling, etc.
- **Attain agreement on the nature of the problem**
  - validation, conflict resolution, negotiation
  - requirements management - maintain the agreement!



Source: *Adapted from Loucopoulos & Karakostas, 1995, p20 and Blum, 1992*

Types of RE project

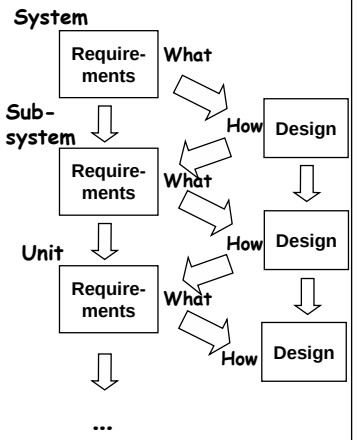
- **Source of Requirements:**
  - Customer-driven
    - involve a specific customer who needs a system to solve a specific problem
  - Market-driven
    - involve a developer who needs to develop a system to be sold in the market
  - Hybrid
    - developed for a specific customer, but want to market the software eventually
- **Nature of the Product**
  - One-off ('bespoke') vs. Packaged ('shrink wrapped')
  - Single system vs. Product Family ('product line')
  - New system vs. Upgrade to existing system
- **These questions affect the role of Requirements:**
  - as a statement of the problem to be solved
  - as a contract between customer and developers
  - for communication between designer, customer and end-users
  - to support system evolution
  - to support design validation

Software Types

- **Information Systems**
  - software to support organizational work
  - includes files/databases as well as applications
  - More than 70% of all software falls in this category, written in languages such as COBOL, RPG and 4GLs.
    - Examples: Payroll, Employee Records, Accounts payable/receivable, Customer records, Transaction records
- **Embedded Systems**
  - software that drives some sort of a hardware process
    - Examples: industrial plant, an elevator system, or a credit card machine.
- **Generic Services**
  - systems that provide some form of generic service
    - Examples: many internet applications, e.g. search engines, stock quote services, credit card processing, etc.
  - Such systems will be developed using a variety of languages and middleware, including Java, C++, CORBA, HTML/XML etc.

What vs. How

- Traditionally, Requirements should specify 'what' without specifying 'how'
  - But this is not always easy to distinguish:
    - What does a car do?
    - What does a web browser do?
    - What does an operating system do?
  - The 'how' at one level of abstraction forms the 'what' for the next level
- Jackson's work provides a clearer distinction
  - 'What' refers to a system's purpose
    - it is external to the system
    - it is a property of the *application domain*
  - 'How' refers to a system's structure and behavior
    - it is internal to the system
    - it is a property of the *machine domain*



Source: Adapted from Jackson, 1995, p207

What are requirements about?

- Some distinctions:
  - Domain Properties are things in the application domain that are true whether or not we ever build the proposed system
  - Requirements are things in the application domain that we wish to be made true by delivering the proposed system
  - A specification is a description of the behaviours the program must have in order to meet the requirements
- Two verification criteria:
  - The Program running on a particular Computer satisfies the Specification
  - The Specification, in the context of the given Domain properties, satisfies the Requirements
- Two validation criteria:
  - Did we discover (and understand) all the important Requirements?
  - Did we discover (and understand) all the relevant Domain properties?

Source: Adapted from Jackson, 1995, p170-171

Validation Example

- Requirement R:
  - "Reverse thrust shall only be enabled when the aircraft is moving on the runway"
- Domain Properties D:
  - Wheel pulses on if and only if wheels turning
  - Wheels turning if and only if moving on runway
- Specification S:
  - Reverse thrust enabled if and only if wheel pulses on
- S + D imply R
  - But what if the domain model is wrong?

Source: Adapted from Jackson, 1995, p172

Another Example

- Requirement R:
  - "The database shall only be accessible by authorized personnel"
- Domain Properties D:
  - Authorized personnel have passwords
  - Passwords are never shared with non-authorized personnel
- Specification S:
  - Access to the database shall only be granted after the user types an authorized password
- S + D imply R
  - But what if the domain assumptions are wrong?

Source: Adapted from Jackson, 1995, p172

RE is all about Description

- A designation
  - singles out a phenomena of interest
    - tells you how to recognize it and gives it a name
  - A designation is always informal,
    - it maps from the fuzzy phenomena to formal language
- A definition
  - gives a formal definition of a term that may be used in other descriptions
    - Note: definitions can be more or less useful, but never right or wrong.
- A refutable description
  - states some property of a domain that could in principle be refuted
    - Might not be practical to refute it, but refutation should be conceivable
  - Refutability depends on an appeal to the designated phenomena of the domain being described
- A rough sketch
  - is a tentative description that is being developed

Source: *Adapted from Jackson, 1995, p58-59*

Examples

- Designations:
  - Parent(x,p) denotes that p is the genetic parent of x
  - Female(x) denotes that x is biologically female
  - …
- Definitions:
  - mother(x,m) ≙ Parent(x,m) and Female(m)
  - sister(x,y) ≙ Female(x) and mother(x,m) and mother(y,m) and father(x,f) and father (y,f)
- Refutable Description:
  - For all m and x, Parent(x, p) implies not(Parent(m, p))
- A rough sketch
  - "Everyone's related somehow"

Source: *Adapted from Jackson, 1995, p58-59*

Requirements are optative

- Traditionally, requirements contain the word 'shall'
  - (and contractually, 'will' means it's optional!)
  - The distinction in English is subtle:
    - "I shall drown. No one will save me"
    - "I will drown. No one shall save me"
- Mood (of a verb):
  - Indicative: asserts a fact ("you sing")
  - Interrogative: asks a question ("are you singing")
  - Imperative: conveys a command ("Sing!")
  - Subjunctive: states a possibility ("I might sing")
  - Optative: expresses a wish ("may you sing")
- For requirements engineering:
  - use the indicative mood for domain properties
  - use the optative mood for requirements
  - Never mix moods in the same description.
  - Anyway, mood changes as development progresses! :-)

Source: *Adapted from Jackson, 1995, p125-127*

Phenomena

- A little Philosophy:
  - Phenomenology
    - the study of the things that appear to exist when you observe the world
  - Ontology
    - the study of what really does exist (independently from any observer)
  - Epistemology
    - the study of what people are capable of knowing (or what they believe)
  - Weltanschauung
    - a world view that defines the set of phenomena that an observer is willing (likely) to observe ('viewpoint')
- Each method has its own Weltanschauung
  - Examples:
    - *OO* sees the world as objects with internal state that respond to stimuli
    - *SA* sees the world as processes that transform data
    - Natural language also defines a viewpoint
  - Each method restricts the set of phenomena you can describe
    - …and therefore what you can model
  - Choose a method that emphasizes the appropriate kinds of phenomena

Source: *Adapted from Jackson, 1995, p143, and Blum 1996, chapter 2*

What is a system?

- **Definition of a System:**
  - Some part of reality that can be observed to interact with its environment
    - Separated from its environment by a boundary
    - A system receives inputs from the environment & send outputs to the environment
    - A system usually have subsystems
    - Systems that endure have a control mechanism
    - Systems have interesting emergent properties
  - Examples:
    - cars, cities, houseplants, rocks, spacecraft, buildings, weather,...
    - operating systems, DBMS, the internet, an organization
  - Non-examples (there aren't many!):
    - numbers, truth values, letters.
  - *A closed system* doesn't interact with its environment (there aren't many!)
- **Systems might have no physical existence**
  - Only manifestations are symbolic/analogical representations of the system
  - Such systems are social constructs: they exist because we agree on ways to observe them

Source: *Adapted from Wieringa, 1996, p10*

Hard vs. Soft Systems

Hard Systems:

- **The system is**
  - precise,
  - well-defined
  - quantifiable
- **No disagreement about:**
  - Where the boundary is
  - What the interfaces are
  - The internal structure
  - Control mechanisms
  - The purpose (??)
- **Examples**
  - ?

Soft Systems:

- **The system…**
  - …is hard to define precisely
  - …is an abstract idea
  - …depends on your perspective
- **Not easy to get agreement**
  - The system doesn't "really" exist
  - Calling something a system helps us to understand it
  - Identifying the boundaries, interfaces, controls, helps us to predict behaviour
  - The "system" is a theory of how some part of the world operates
- **Examples:**
  - All human activity systems
  - (what else?)

Lecture 3: Requirements Elicitation I

Last Week:
Approaches to RE
Processes, methods & techniques
Importance of Description
Role of modeling

This Week:
Elicitation (I)
Traditional approaches
Interviews & Questionnaires
Scenarios, Goals and Use-Cases

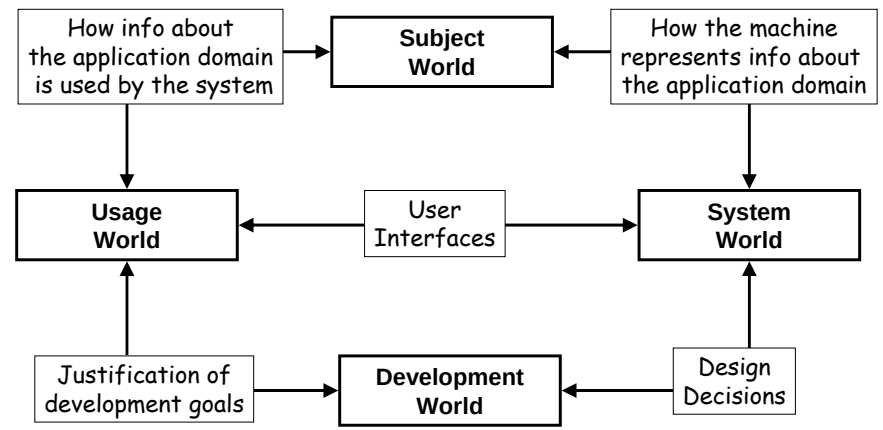
Next Week:
Elicitation (II)
Cognitive approaches
Contextual approaches
Ethnography

Requirements Elicitation

- **Starting point**
  - Some notion that there is a "problem" that needs solving
    - e.g. dissatisfaction with the current state of affairs
    - e.g. a new business opportunity
    - e.g. a potential saving of cost, time, resource usage, etc.
  - A Requirements Engineer is an agent of change
- **The requirements engineer must:**
  - identify the "problem"/"opportunity"
    - Which problem needs to be solved? (identify problem Boundaries)
    - Where is the problem? (understand the Context/Problem Domain)
    - Whose problem is it? (identify Stakeholders)
    - Why does it need solving? (identify the stakeholders' Goals)
    - How might a software system help? (collect some Scenarios)
    - When does it need solving? (identify Development Constraints)
    - What might prevent us solving it? (identify Feasibility and Risk)
  - elicit enough knowledge
    - ...sufficient to analyze requirements for validity, consistency, completeness, etc.
  - i.e. become an expert in the problem domain
    - although ignorance is important too [Berry]

W6H
The journalist's technique:
What?
Where?
Who?
Why?
When?
How?
(Which?)

The four worlds



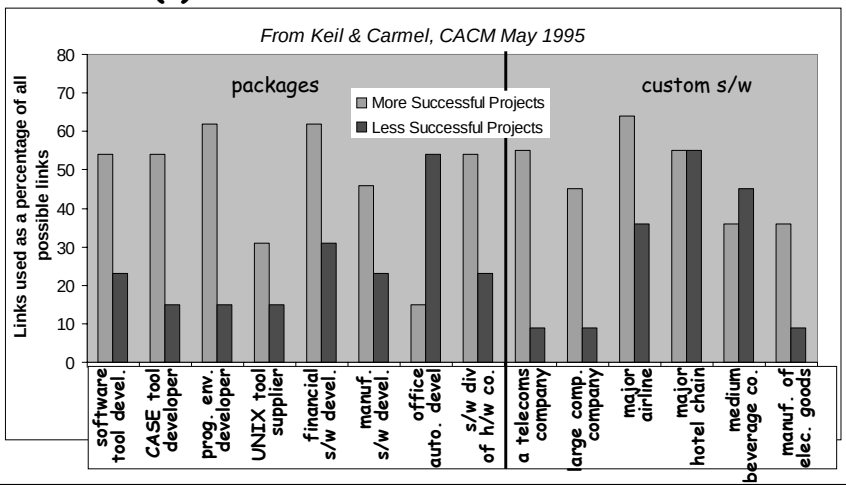
Source: Adapted from Loucopoulos & Karakostas, 1995, p73

Difficulties of Elicitation

- **Thin spread of domain knowledge**
  - The knowledge might be distributed across many sources
    - It is rarely available in an explicit form (I.e. not written down)
  - There will be conflicts between knowledge from different sources
    - People have conflicting goals
    - People have different understandings of the problem
- **Tacit knowledge (The "say-do" problem)**
  - People find it hard to describe knowledge they regularly use
    - Descriptions may be inaccurate rationalizations of expert behaviour
- **Limited Observability**
  - The problem owners might be too busy solving it using the existing system
  - Presence of an observer may change the problem
    - E.g. the Probe Effect and the Hawthorne Effect
- **Bias**
  - People may not be free to tell you what you need to know
    - Political climate & organisational factors matter
  - People may not want to tell you what you need to know
    - The outcome will affect them, so they may try to influence you (hidden agendas)

Importance of links with customer(s)

- **Successful projects tend to have more links with customer(s)**



Source: Adapted from Keil and Carmel, 1995, p37

Elicitation Techniques

- **Traditional Approaches**
  - Introspection
  - Existing Documents
  - Data Analysis
  - Interviews
    - Open-ended
    - Structured
  - Surveys / Questionnaires
  - Group elicitation
    - Focus Groups
    - Brainstorming
    - JAD/RAD workshops
  - Prototyping
- **Representation-based approaches**
  - Goal-based
  - Scenario-Based
  - Use Cases

This week

Next week

- **Contextual (social) approaches**
  - Ethnographic techniques
    - Participant Observation
    - Enthnomethodology
  - Discourse Analysis
    - Conversation Analysis
    - Speech Act Analysis
  - Participatory Design
  - Sociotechnical Methods
    - Soft Systems Analysis
- **Cognitive approaches**
  - Task analysis
  - Protocol analysis
  - Knowledge Acquisition Techniques
    - Card Sorting
    - Laddering
    - Repertory Grids
    - Proximity Scaling Techniques

Questionnaires

- **Advantages**
  - Can quickly collect info from large numbers of people
  - Can be administered remotely
  - Can collect attitudes, beliefs, characteristics
- **Disadvantages**
  - Simplistic (presupposed) categories provide very little context
    - No room for users to convey their real needs
- **Watch for:**
  - Bias in sample selection
  - Bias in self-selecting respondents
  - Small sample size (lack of statistical significance)
  - Leading questions (*"have you stopped beating your wife?"*)
  - Appropriation (*"What is this a picture of?"*)
  - Ambiguous questions (I.e. not everyone is answering the same question)
  - *Questionnaires MUST be prototyped and tested*

Source: *Adapted from Goguen and Linde, 1993, p154.*

Interviews

- **Types:**
  - Structured - agenda of fairly open questions
  - Open-ended - no pre-set agenda
- **Advantages**
  - Rich collection of information
- **Disadvantages**
  - Large amount of qualitative data can be hard to analyze
  - Hard to compare different respondents
  - Interviewing is a difficult skill to master
- **Watch for**
  - Unanswerable questions ("how do you tie your shoelaces?")
  - Tacit knowledge (and post-hoc rationalizations)
  - Removal from context
  - Interviewer's attitude may cause bias (e.g. variable attentiveness)

Source: *Adapted from Goguen and Linde, 1993, p154.*

Group Elicitation Techniques

- **Types:**
  - Joint/Rapid Application Development (JAD/RAD) Workshops
  - Focus Groups
  - Brainstorming
- **Advantages**
  - More natural interaction between people than formal interview
  - Can gauge reaction to stimulus materials (e.g. mock-ups, storyboards, etc)
- **Disadvantages**
  - May create unnatural groups (uncomfortable for participants)
  - Danger of *Groupthink*
  - May only provide superficial responses to technical questions
  - Requires a highly trained facilitator
- **Watch for**
  - sample bias
  - dominance and submission

"Hard Data" Collection

- **Identify Collections of Hard Data**
  - Facts and figures, financial information,…
  - Reports used for decision making,…
  - Survey results, marketing data,…
- **Sampling**
  - Sampling used to select representative set from a population
    - Purposive Sampling - choose the parts you think are relevant without worrying about statistical issues
    - Simple Random Sampling - choose every kth element
    - Stratified Random Sampling - identify strata and sample each
    - Clustered Random Sampling - choose a representative subpopulation and sample it
  - Sample Size is important
    - balance between cost of data collection/analysis and required significance

Use Cases

- **What is a use case?**
  - Each different way that an actor interacts with a system is a use case
    - "a description of a *sequence of actions* that a system performs that yields an *observable result of value to a particular actor* " [Booch]
    - All the use cases need to be enumerated (or the requirements will not be complete)
  - A description of a set of possible scenarios, with a common purpose
  - Typically written in natural language
  - No internal description of the system; just the interaction.
- **Combining use cases**
  - extends/uses
- **Advantages & Disadvantages**
  - detailed characterization of all possible interaction with the system
  - helps in drawing system boundary, and scoping the requirements
  - Use cases do not capture domain knowledge!!
  - Don't confuse use cases with a precise specification!

Source: *Adapted from Rumbaugh 1997, p123-124*

Scenarios

- **Scenarios**
  - Specific sequence of interaction between actor and system
  - Tend to be short (e.g between 3 and 7 steps)
  - May be:
    - positive (i.e. required behavior)
    - negative (i.e an undesirable interaction)
  - May be indicative or optative
- **Advantages**
  - Very natural: stakeholders tend to use them spontaneously
  - Short scenarios very good for quickly illustrating specific interactions
- **Disadvantages**
  - Lack of structure: need use cases or task models to provide higher level view

Source: *Adapted from Dardenne, 1993.*

Goal-based Approaches

- **Approach**
  - Focus on why systems are constructed
  - Express the 'why' as a set of stakeholder goals
  - Use goal refinement to arrive at specific requirements
  - Goal analysis
    - document, organize and classify goals
  - Goal evolution
    - refine, elaborate, and operationalize goals
  - Goal hierarchies show refinement and obstacle relationships between goals
- **Advantages**
  - Reasonably intuitive
  - Explicit declaration of goals provides sound basis for conflict resolution
- **Disadvantages**
  - Hard to cope with evolution of goals
  - Can regress forever up (or down) the goal hierarchy

Source: *Adapted from Anton, 1996.*

Lecture 4: Requirements Elicitation II

Last Week:
Elicitation (I)
Traditional approaches
Interviews & Questionnaires
Scenarios, Goals and Use-Cases

This Week:
Elicitation (II)
Cognitive approaches
Contextual approaches
Ethnography as an RE technique

Next Week:
Modeling and Analysis (I)
Modeling Goals
Modeling Organisations
Modeling Non-Functional Reqs

Knowledge Elicitation Techniques in RE

- **Background**
  - Knowledge elicitation is concerned with discovering 'expert' knowledge
  - Grew out of Expert Systems work in the 80's
  - Originally focussed on deriving expert's "rules" for Rule-based Systems
  - More recently, focussed on "problem solving methods"
- **But KE is hard**
  - Separation of domain knowledge from performance knowledge
  - Modeling problems
    - Brittleness
    - Assumption of rationality
  - Representational Problem
    - epistemological inadequacy
    - expressiveness vs. acquirability
  - Expert Bias

Example Techniques
- Eliciting domain knowledge
  - Card Sorting
  - Laddering
  - Proximity Scaling Techniques
- Eliciting performance knowledge
  - Protocol Analysis
- Using Multiple Experts
  - Delphi Technique
  - Focus Groups
  - Repertory Grids
- Automated Techniques
  - Machine Learning

Why is KE so hard?

- **Experts are not used to describing what they do.**
  - Three stage model of learning:
    1) cognitive - verbal rehearsal of tasks;
    2) associative - reinforcement through repetition, verbal mediation disappears
    3) autonomous - compiled, no conscious awareness of performance.
  - Procedural and declarative are different mechanisms
    - Declarative knowledge becomes procedural with repeated application - experts lose awareness of what they know and cannot introspect reliably
    - Experts have little or no introspective access to higher order cognitive processes
- **Representational Problems**
  - Experts don't have the language to describe their knowledge
    - No spoken language offers the necessary precision
    - Knowledge Engineer and Expert must work together to create a suitable language
  - Different knowledge representations are good for different things
    - Epistemological adequacy: does the formalism express expert's knowledge well?
- **Brittleness**
  - Knowledge is created, not extracted.
    - Knowledge models are abstractions of reality and hence are unavoidably selective
    - Brittleness is caused by the simplifying assumptions - instead of adding more knowledge, a better (more comprehensive) model is needed.

Knowledge Elicitation Techniques

- **Protocol Analysis**
  - based on vocalising behaviour
    - Think aloud vs. retrospective protocols
  - Advantages
    - Direct verbalisation of cognitive activities
    - Embedded in the work context
    - Good at revealing interaction problems with existing systems
  - Disadvantages
    - Essentially based on introspection, hence unreliable
    - No social dimension
- **Proximity Scaling Techniques**
  - Given some domain objects, derive a set of dimensions for classifying them:
    step 1: pairwise proximity assessment among domain elements
    step 2: automated analysis to build multi-dimensional space to classify the objects
  - Advantages
    - help to elicit mental models, where complex multivariate data is concerned
    - good for eliciting tacit knowledge
  - Disadvantages
    - Requires an agreed on set of objects
    - Only models classification knowledge (no performance knowledge)

Source: *Adapted from Hudlicka, 1996.*

more KE techniques

- **Card Sorting**
  - For a given set of domain objects, written on cards:
    - Expert sorts the cards into groups...
    - ...then says what the criterion was for sorting, and what the groups were.
  - Advantages
    - simple, amenable to automation
    - elicits classification knowledge
  - Problems
    - suitable entities need to be identified with suitable semantic spread across domain.
    - No performance knowledge
- **Laddering**
  - Uses a set of probes (types of question) to acquire structure and content of stakeholders' knowledge.
    - Interview the expert.
    - Use questions to move up and down a conceptual hierarchy
  - Advantages
    - deals with hierarchical knowledge, including poly-hierarchies (e.g., goal trees, "is-a" taxonomies).
    - knowledge is represented in standardised format
    - can elicit structural knowledge
    - suitable for automation.
  - Disadvantages
    - assumes hierarchically arranged knowledge.

Expert Bias

- **What is bias?**
  - Bias only exists in relation to some reference point
    - can there ever be "no bias"? (reflects reality or truth)
  - We cannot perceive reality directly:
    - It is interpreted through a filter of mental models
    - …mediated by our senses and neural pathways.
  - All decision making is based partly on personal value systems.
- **Types of bias:**
  - Motivational bias
    - the expert makes accommodations to please the interviewer or some other audience
  - Cognitive bias
    - the expert does not follow objective rules or standards

Sources of Bias

- Social pressure
  response to verbal and non-verbal cues from an interviewer
- Group think
  response to reactions of other experts
- Impression management
  response to imagined reactions of managers, clients, etc.
- Wishful thinking
  response to hopes or possible gains.
- Misinterpretation
  Analyst selectively interprets to support what she currently believes.
- Misrepresentation
  expert cannot accurately fit a response into the requested response mode
- anchoring
  contradictory data is ignored once an initial solution is available
- inconsistency
  assumptions made earlier are forgotten
- availability
  some data are easier to recall than others
- underestimation of uncertainty
  tendency to underestimate by a factor of 2 or 3.

KA from Multiple Experts

- **Delphi technique**
  - Used where contact between experts is difficult:
    - Each expert submits their judgement
    - All judgements are circulated anonymously to all experts
    - Each expert then submits a revised judgement
    - Iterate until judgements converge
- **Focus Groups**
  - A technique derived from marketing:
    - Assemble experts together and discuss the problem
    - Discussion may be structured (e.g. debate) or unstructured
- **Repertory Grids** (based on Kelly's Personal Construct Theory)
  - Used to detect terminological differences
    - Get the experts to agree a set of entities
    - Each expert provides attributes and values
    - For each attribute in expert A's grid, find the closest match in expert B's grid. (i.e. are there attributes which have the same discriminatory function?)
    - Experts then rate the entities using each other's attributes

The Ethnomethodolgist's View

- **Requirements elicitation is a social activity:**
  - Because it involves people-to-people communication (through discussions, observation, etc.)
  - Because it involves negotiation in bringing about consensus when there is disagreement.
  - Because it affects and changes human activity systems
- **The domain of application is often a social world**
  - Need techniques that uncover the order of the social world
    - social order might not be immediately obvious or describable
    - social order cannot be assumed to have an *a priori* structure
  - Social order can only be understood through immersion
    - social order is constructed by the participants' actions
    - need to witness the unfolding of social phenomena
    - cannot just collect data using pre-given categories
  - Need to consider
    - How meanings develop and evolve within context
    - The methods people use to make sense of the world around them

Ethnomethodology

- **Basis**
  - Social world is ordered
  - The social order may not be immediately obvious, nor describable from common sense
  - The social order cannot be assumed to have an a priori structure
  - I.e. social order emerges only when an observer immerses herself in it.
  - Emphasizes the importance of natural setting
- **Categories**
  - Most conventional approaches assume preexisting categories
    - This may mislead the observer (appropriation)
  - Ethnography attempts to use the subjects' own categories
  - Related to postmodern deconstruction: "there is no grand narrative"
- **Measurement**
  - No scientific objectivity, so use the subjects' own measurement theory

Source: Adapted from Goguen and Linde, 1993, p158.

Participant Observation

- **Approach**
  - Observer spends time with the subjects, joining in, long enough to become a member of the group ('longitudinal studies')
- **Advantages**
  - Contextualized;
  - Reveals details that other methods cannot
- **Disadvantages**
  - Extremely time consuming!
  - Resulting 'rich picture' is hard to analyze
  - Cannot say much about the results of proposed changes
- **Watch for**
  - going native!

An RE Methods Classification (after Lyotard)