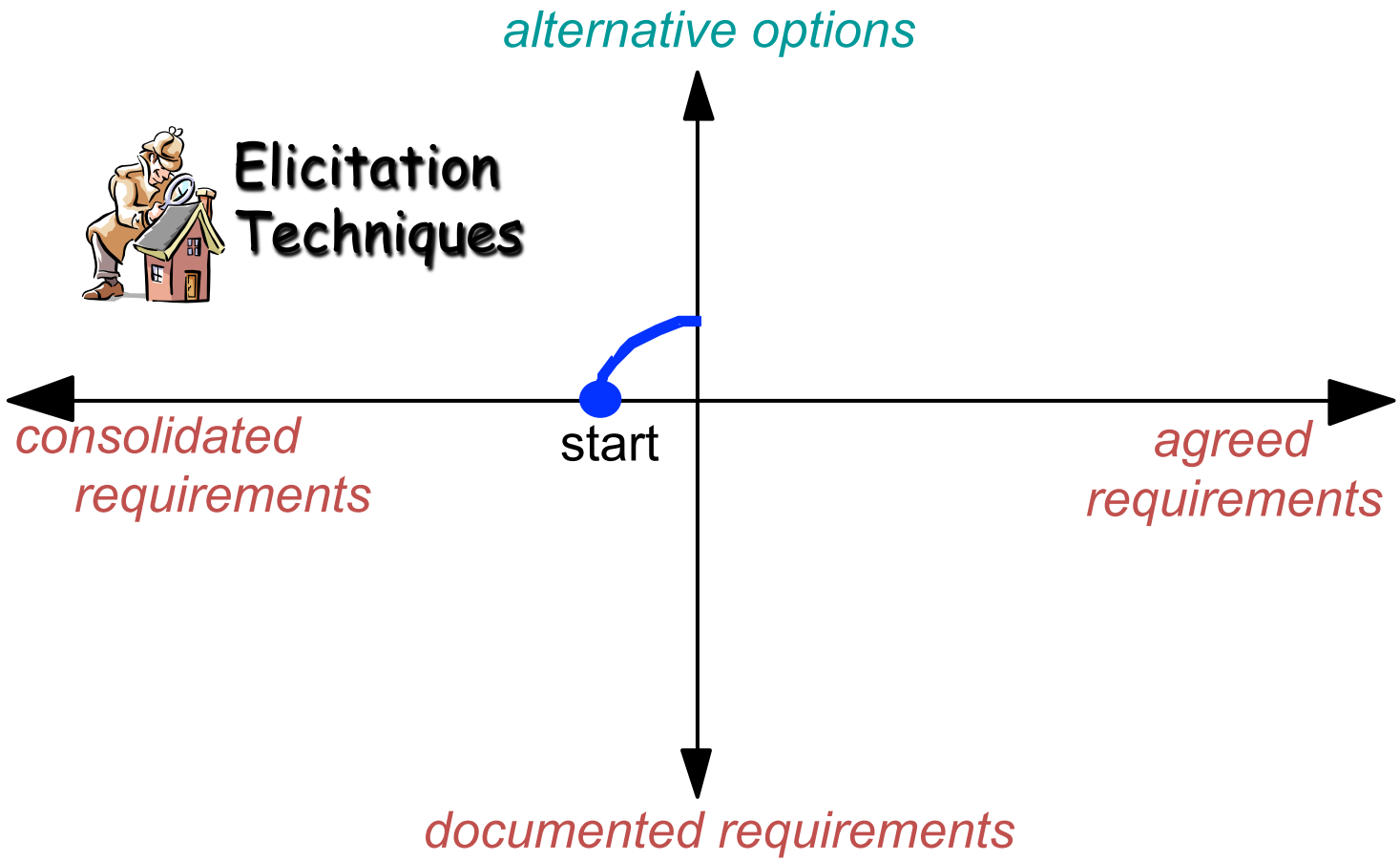
Requirements Engineering - Domain Understanding and Requirements Elicitation -

Slides adapted from official slides of the book “Requirements Engineering”, A. van Lamsweerde


alternative options
Elicitation Techniques
consolidated requirements
start
agreed requirements
documented requirements

Domain analysis & requirements elicitation: outline

- **Identifying stakeholders** & interacting with them
- **Artefact-driven** elicitation techniques
  - Background study
  - Data collection, questionnaires
  - Repertory grids, card sorts for concept acquisition
  - Scenarios, storyboards for problem world exploration
  - Prototypes, mock-ups for early feedback
  - Knowledge reuse: domain-independent, domain-specific
- **Stakeholder-driven** elicitation techniques
  - Interviews
  - Observation and ethnographic studies
  - Group sessions

Stakeholder Selection

- understanding the problem world requires **representative stakeholders**
- Relevant aspects for the selection:
  - Position in the organization
  - Role in making decisions about the system-to-be
  - Level of domain expertise
  - Exposure to the perceived problem
  - Influence in system acceptance
  - Personal objectives and conflicts of interest

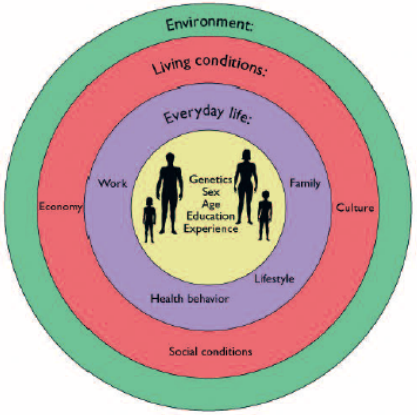
General Obstacles to Knowledge Acquisition

Distributed and conflicting knowledge sources

Obstacles to Good Communications

Tacit knowledge and hidden needs

Difficult access to sources



Sociopolitical factors

Unstable conditions

Interactions with Stakeholders

- Communication skills
  - Right terminology
  - Get the key points
  - Trustworthy relationships

- Knowledge Reformulation
  - Review meetings where the relevant knowledge about the problem world, acquired from multiple sources, is presented in an integrated and structured way

ARTEFACT-DRIVEN ELICITATION TECHNIQUES

Background study

- Collect, read, synthesize documents about...
  - the **organization**: organizational charts, business plans, financial reports, meeting minutes, etc
  - the **domain**: books, surveys, articles, regulations, reports on similar systems in the same domain
  - the **system-as-is**: documented workflows, procedures, business rules; exchanged documents; defect/complaint reports, change requests, etc.

What are the issues of background studies?

CONS

Many voluminous documents must be read!!

- Key information has to be extracted from a mass of irrelevant details
- Exploit *meta-knowledge* to discriminate useful from irrelevant information

PROS

It produces basic information useful for interacting with stakeholders

Questionnaires

- Submit a list of questions to selected stakeholders, each with a list of possible answers (+ brief context if needed)
  - **Multiple choice** questions
  - **Weighting** questions: list of statements to be weighted
    - qualitatively ('high' , 'low", ...),
      - How many options? e.g., see Liker's scale
    - quantitatively (percentages)
    - to express perceived importance, preference, risk...
- Effective for acquiring subjective info quickly, cheaply, remotely from many people
- Helpful for preparing better focused interviews

Questionnaires should be carefully prepared

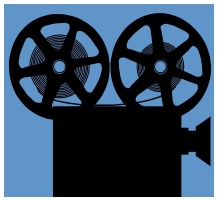
- What are the main issues?

Questionnaires should be carefully prepared

- Subject to ...
  - multiple **biases**: recipients, respondents, questions, answers
  - unreliable info: misinterpretation of questions, of answers, inconsistent answers, ...

=> Guidelines for questionnaire design/validation:

- Select a **representative, statistically significant sample of people**; provide motivation for responding
- Check **coverage of questions**, of possible answers
- Make sure questions, answers, formulations are **unbiased** & **unambiguous**
- Add implicitly redundant questions to **detect inconsistent answers**
- Have your questionnaire checked by a third party

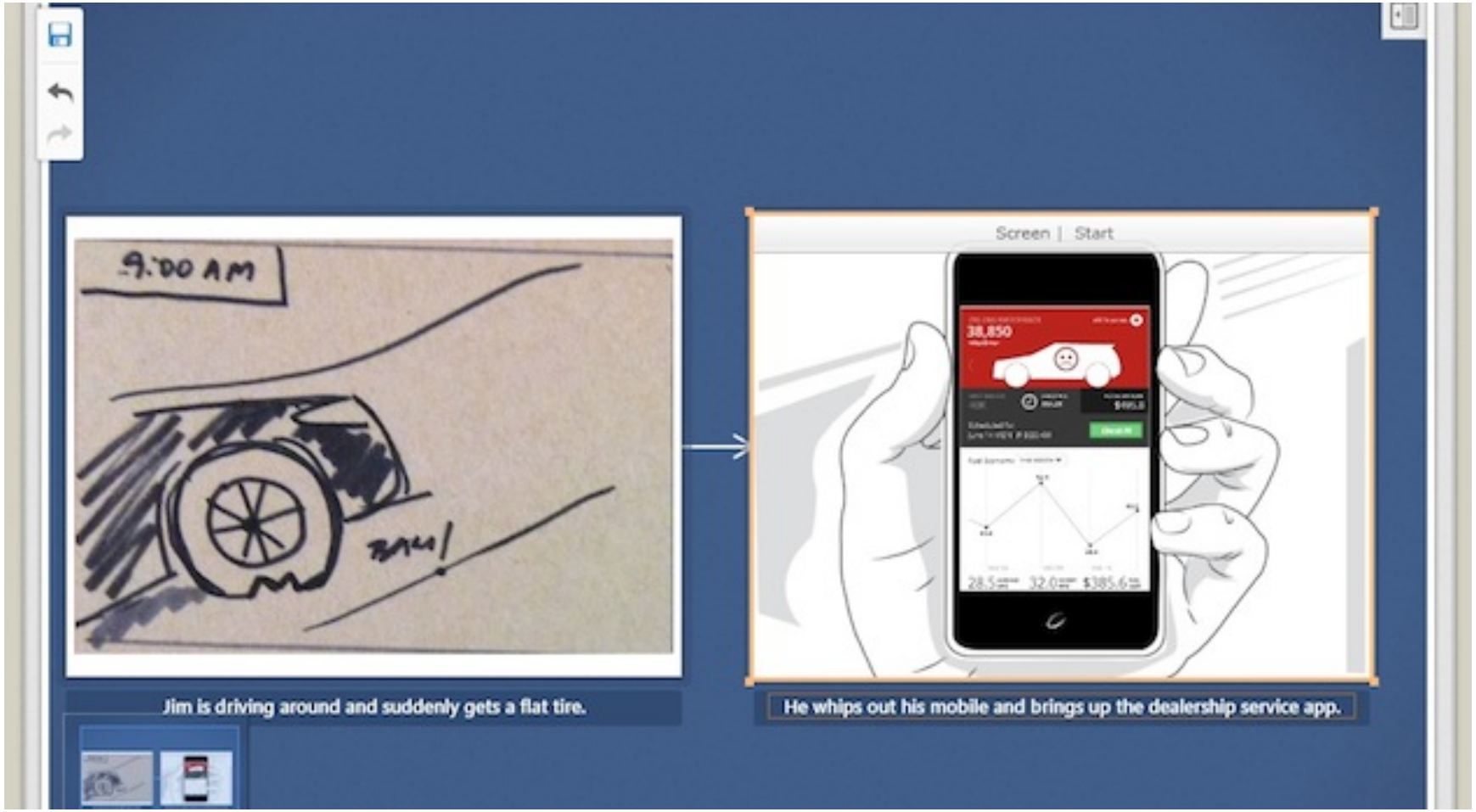


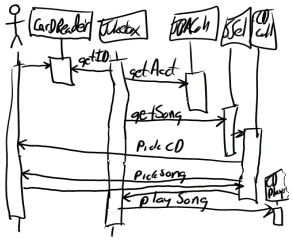
Storyboards

- Acquire or validate info from concrete examples through narratives ...
  - about the system-*as-is*
  - about the system-*to-be*
- **Storyboard**: tells a story by a sequence of snapshots
  - Snapshot = sentence, sketch, slide, picture, etc.
  - Possibly structured with annotations:
    - WHO are the players, WHAT happens to them, WHY this happens, WHAT IF this does / does *not* happen, etc
  - **Passive** mode (for validation): stakeholders are told the story
  - **Active** mode (for joint exploration): stakeholders contribute

Comics: Storyboard

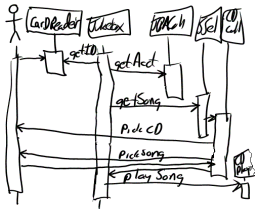
Requirements: Storyboard





Scenarios

- Illustrate typical sequences of interaction among system components to meet an implicit objective (typically who + what + how)
- Could be used as specification of acceptance test cases
- Represented by text or diagram



Types of scenario

- **Positive** scenario = one behavior the system should cover (example)
- **Negative** scenario = one behavior the system should exclude (counter-example), e.g.
  1. A participant returns a list of constraints covering all dates within the given date range
  2. The scheduler forwards this message to all participants asking them for alternative constraints within extended date range
- **Normal** scenario = everything proceeds as expected
- **Abnormal** scenario = a desired interaction sequence in exception situation (still positive)

  e.g. meeting initiator not authorized, participant constraints not valid

Scenario example: meeting scheduling

1. The **initiator** *asks* the **scheduler** for planning a meeting within some date range. The request includes a list of desired participants.
2. The **scheduler** checks that the initiator is entitled to do so and that the request is valid. It *confirms* to the **initiator** that the requested meeting is initiated.
3. The **scheduler** *asks* all **participant**s in the submitted list to send their date and location constraints back within the prescribed date range.
4. When a **participant** *returns* her constraints, the **scheduler** validates them (e.g., with respect to the prescribed date range). **It** *confirms* to the **participant** that the constraints have been safely received.
5. Once all valid constraints are *received*, the **scheduler** determines a meeting date and location that fit them.
6. The **scheduler** *notifies* the scheduled meeting date and location to the **initiator** and to all invited **participant**s

What type of scenario is this?
Positive/Negative, Normal/Abnormal?

Scenario example: meeting scheduling

…

4. A **participant** *returns* a list of constraints covering all dates within the given date range
5. The **scheduler** *forwards* this message to all **participants** asking them for alternative constraints within extended date range

…

4. The **scheduler** does *not receive* the participant constraints in due time

…

What type of scenarios are these?
Positive/Negative, Normal/Abnormal?

Pros of Scenarios

- Concrete examples/counter-examples
- Appealing to stakeholders
- Usable as acceptance test cases

Likely the most useful/ practical elicitation method!!

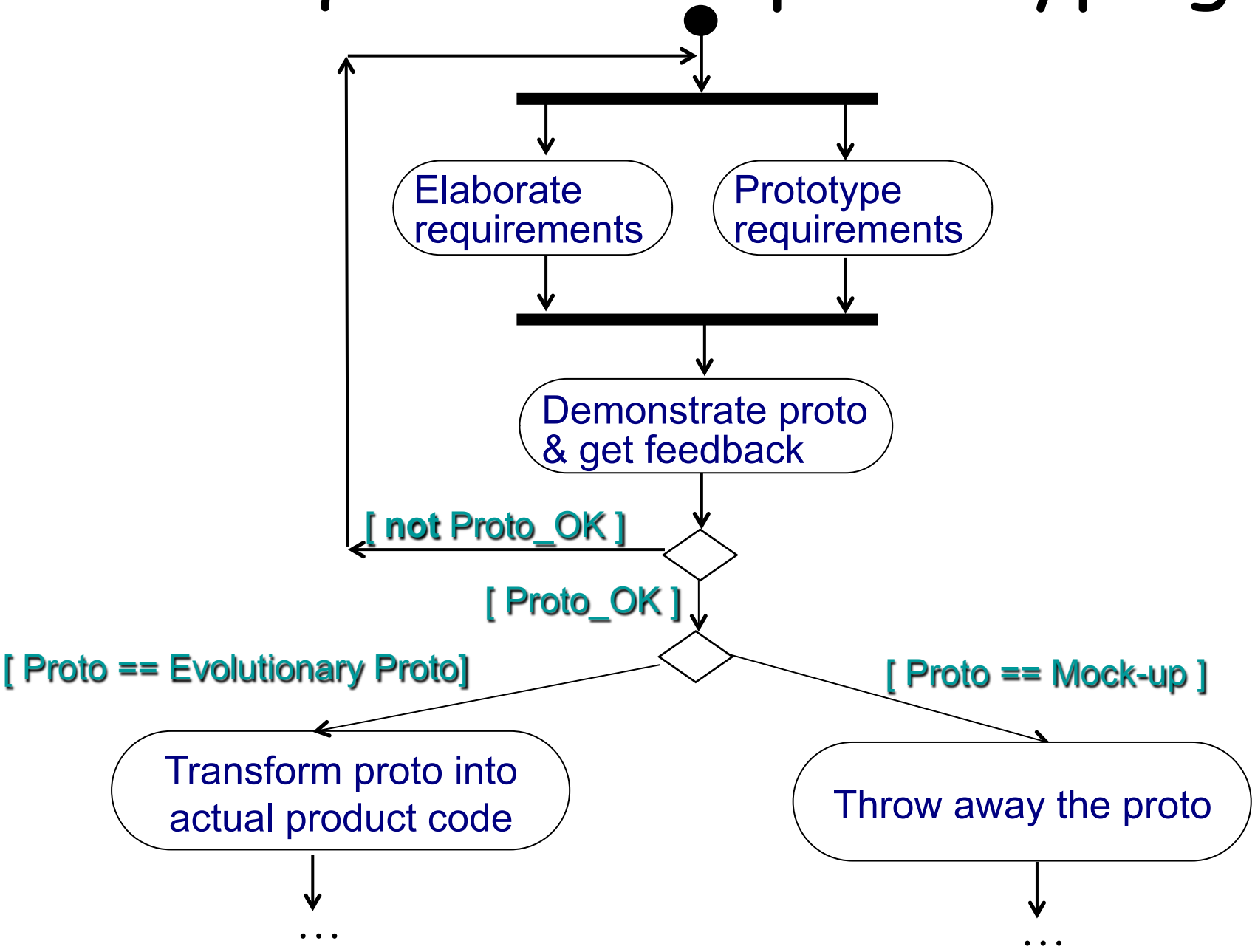
Cons of Scenarios

- Inherently partial
- Combinatorial explosion
- Potential overspecification: unnecessary sequencing
- May contain irrelevant details
- Incompatible granularities from different stakeholders

Prototypes & mock-ups

- **Goal**: check req adequacy from direct user feedback, by showing reduced **sketch of software-to-be in action**
  - focus on unclear, hard-to-formulate reqs to elicit further
- **Prototype** = quick implementation of some aspects
  - **Functional** proto: focus on specific functional reqs
  - **User interface** proto: focus on usability by showing input-output forms, dialog patterns

Requirements prototyping



Pros & Cons

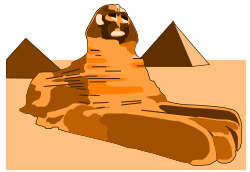
☺ Concrete flavor of what the software will look like

=> clarify reqs, elicit hidden ones, improve adequacy, understand implications, user training, ...

☹ Can be misleading, set expectations too high

- Missing functionalities, ignore non-functional reqs…

☹ 'Quick-and-dirty' code, hard to reuse for sw development

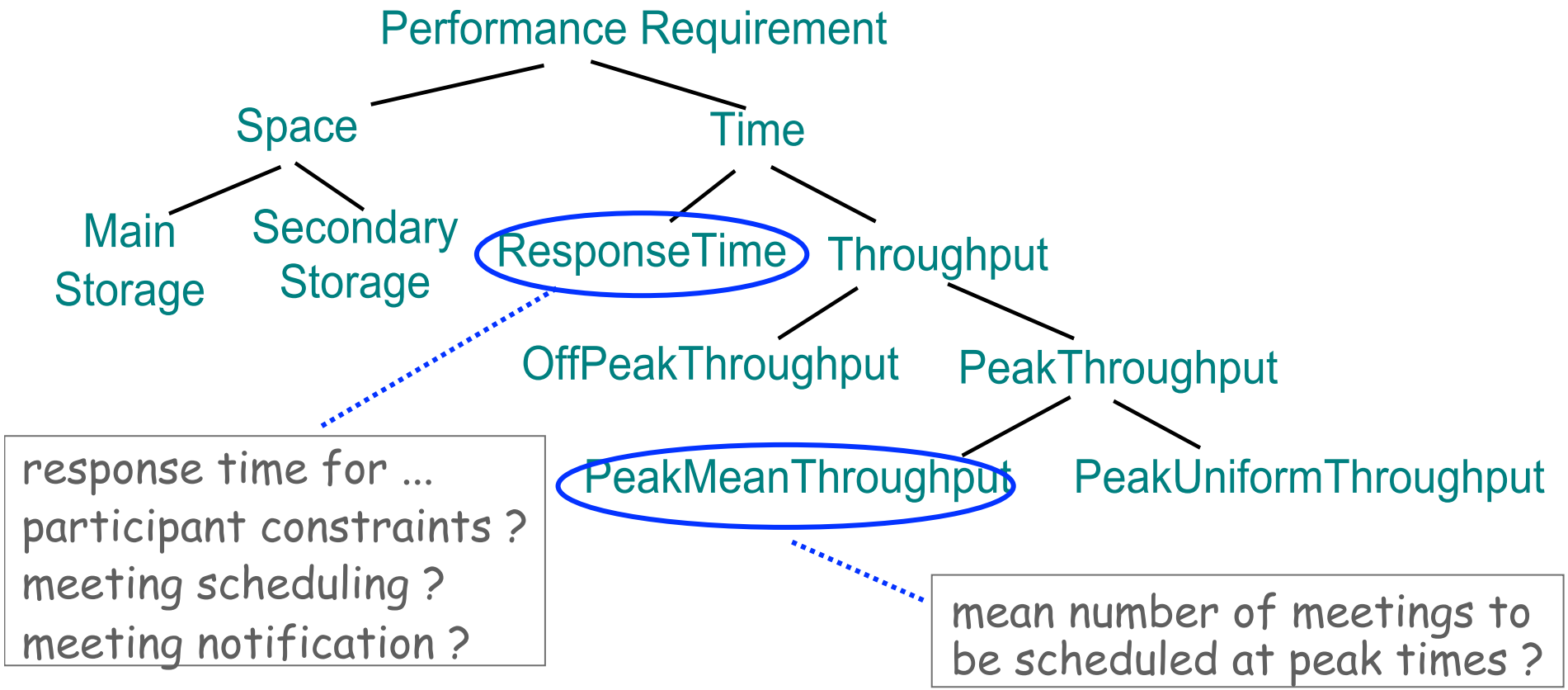


Knowledge reuse

- **Goal**: speed up elicitation by reuse of knowledge from experience with related systems
- General reuse process
  1. **RETRIEVE** relevant knowledge from other systems
  2. **TRANSPOSE** it to the target system
  3. **VALIDATE** the result, **ADAPT** it if necessary & **INTEGRATE** it with the system knowledge already acquired
- Knowledge could be **domain independent** or **domain dependent**

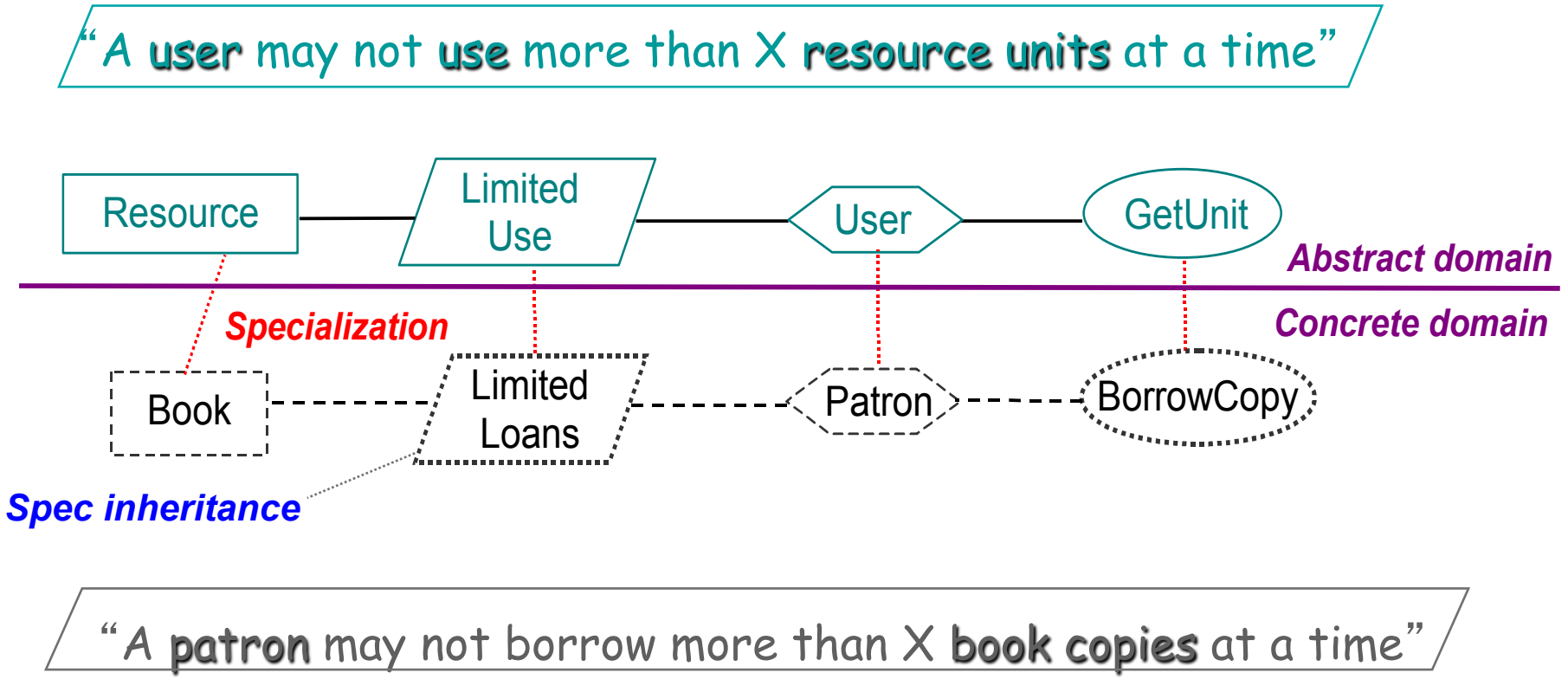
Reuse of domain-independent knowledge: requirements taxonomies

- For each leaf node in available req taxonomies:
  "Is there any system-specific req instance from this class?"
- More specific taxonomy => more focused search



Reuse of domain-specific knowledge

- Items acquired as **specializations** of abstract items to target system: feature inheritance + system-specific renaming



Pros & Cons

- ☺ Expert analysts naturally **reuse from past experience**
- ☺ Significant guidance and **reduction of elicitation efforts**
- ☺ Inheritance of structure & quality of abstract domain spec
- ☺ Effective for **completing** requirements with overlooked aspects

- ☹ Effective only if abstract domain **sufficiently "close"** and accurate
- ☹ Defining abstract domains for significant **reusability is hard**
- ☹ Validation & **integration efforts**
- ☹ Near-matches may require tricky adaptations

Other artefacts-based methods

- **Card sort**: ask stakeholders to partition a set of cards
  - Each card captures a concept textually or graphically
  - Cards grouped into subsets based on stakeholder's criteria
  - **Goal**: acquire further info about concepts already elicited
  - For each subset, ask the implicit shared property used for grouping
  - Iterate with same cards for new groupings/properties

- **Data collection**: marketing data, usage statistics, performance figures, costs, ...
  - **Goal**: Gather undocumented facts & figures
  - by designed experiments *or* selection of representative data sets from available sources (use of statistical sampling techniques)
  - May complement background study

STAKEHOLDER-DRIVEN ELICITATION TECHNIQUES

Interviews

- Primary technique for knowledge elicitation
  1. Select stakeholder specifically for info to be acquired
  2. Organize meeting with interviewee, ask questions, record answers
  3. Write report from interview transcripts
  4. Submit report to interviewee for validation & refinement

- Single interview may involve multiple stakeholders
  - ☺ saves times
  - ☹ weaker contact; individuals less involved, speak less freely

Types of interview

- **Structured** interview: predetermined set of questions
  - specific to purpose of interview
  - some open-ended, others with pre-determined answer set

  => more focussed discussion, no rambling among topics

- **Unstructured** interview: no predetermined set of questions
  - free discussion about system-as-is, perceived problems, proposed solutions

  => exploration of possibly overlooked issues

=> Effective interviews should mix both modes ...
  - start with structured parts
  - shift to unstructured parts as felt necessary

Guidelines for effective interviews

- Come prepared, to focus on right issue at right time
  - Avoid questions obvious to the interviewee (e.g., backgound study first)
  - predesign a sequence of questions for this interviewee
- Centre the interview on the interviewee's work & concerns
- Keep control over the interview
- Make the interviewee feel comfortable
  - *Start:* break ice, provide motivation, ask easy questions
  - Consider the person too, not only the role
  - Do always appear as a trustworthy partner
  - Ask Why questions
- Avoid certain types of questions
  - opinionated or biased
  - affirmative
  - obvious or impossible answer for this interviewee
- Edit & structure interview transcripts while still fresh in mind
  - including personal reactions, attitudes, etc
- Keep interviewee in the loop
  - co-review interview transcript for validation & refinement

Do we need alternative ways of interacting with stakeholders?

Explain how to tie shoelaces

Observation & ethnographic studies

Main principle: sometime understanding a task is often easier by observing people performing it rather than by verbal or textual explanation

Explain how to tie shoelaces

Types of Observation Studies

- **Passive** observation: no interference with task performers
  - Watch from outside, record (notes, video), edit transcripts, interpret
  - **Protocol analysis**: task performers concurrently explain it
  - **Ethnographic studies**: over long periods of time, try to discover emergent properties of social group involved

- **Active** observation: you get involved in the task, even become a team member

pros & cons

☺ May reveal
- **tacit knowledge** that would not emerge otherwise
- hidden problems through tricky ways of doing things
- culture-specific aspects to be taken into account

☺ Contextualization of acquired info

☹ Slow & expensive: to be done over long periods of time, at different times, under different workload conditions

☹ Potentially inaccurate: people behave differently when observed

☹ Focus on system-*as-is*

Group sessions

- More perception, judgment, invention from interactions within group of diverse people
- Elicitation takes place in series of group workshops (a few days each) + follow-up actions
- Audiovisuals, wall charts to foster discussion, record outcome

Types of Group Sessions

- **Structured** group sessions:
  - Each participant has a clearly defined **role**: leader, moderator, manager, user, developer, …
  - Contributes to req elaboration according to his/her role, towards reaching synergies
  - Generally focused on high-level reqs
- **Unstructured** group sessions (brainstorming):
  - Participants have a less clearly defined role
  - Two separate stages ...

    1. **Idea generation** to address a problem: as many ideas as possible from each participant without censorship/criticism

    2. **Idea evaluation**: by all participants together according to agreed criteria (e.g. value, cost, feasibility) to prioritize ideas

Group sessions: pros & cons

- ☺ Potentially
  - wider exploration of issues & ideas
  - more inventive ways of addressing problems
- ☺ Synergies => agreed conflict resolutions

- ☹ Group composition is critical
  - time consuming for key, busy people
  - heavily relying on leader expertise & skills
  - group dynamics, dominant persons => biases, inadequacies
- ☹ Risk of
  - missing focus & structure => rambling discussions, little concrete outcome, waste of time
  - superficial coverage of more technical issues

Domain analysis & requirements elicitation: summary

- Identifying the right stakeholders, interacting the right way

- Artefact-driven elicitation techniques
  - Background study as a prerequisite
  - Data collection, questionnaires for preparing interviews
  - Repertory grids, card sorts for concept characterization
  - Scenarios, storyboards for concrete exploration
  - Prototypes, mock-ups for early feedback & adequacy check
  - Knowledge reuse brings a lot: domain-independent, domain-specific

- Stakeholder-driven elicitation techniques
  - Interviews are essential - structured, unstructured, cf. guidelines
  - Observation, ethnographic studies for hidden knowledge
  - Group sessions for broader, more inventive acquisition & agreement