

Design-oriented Approaches

(version 0.9, 1/4/05)

Code: res-design-design

Daniel K. Schneider, TECFA, University of Geneva

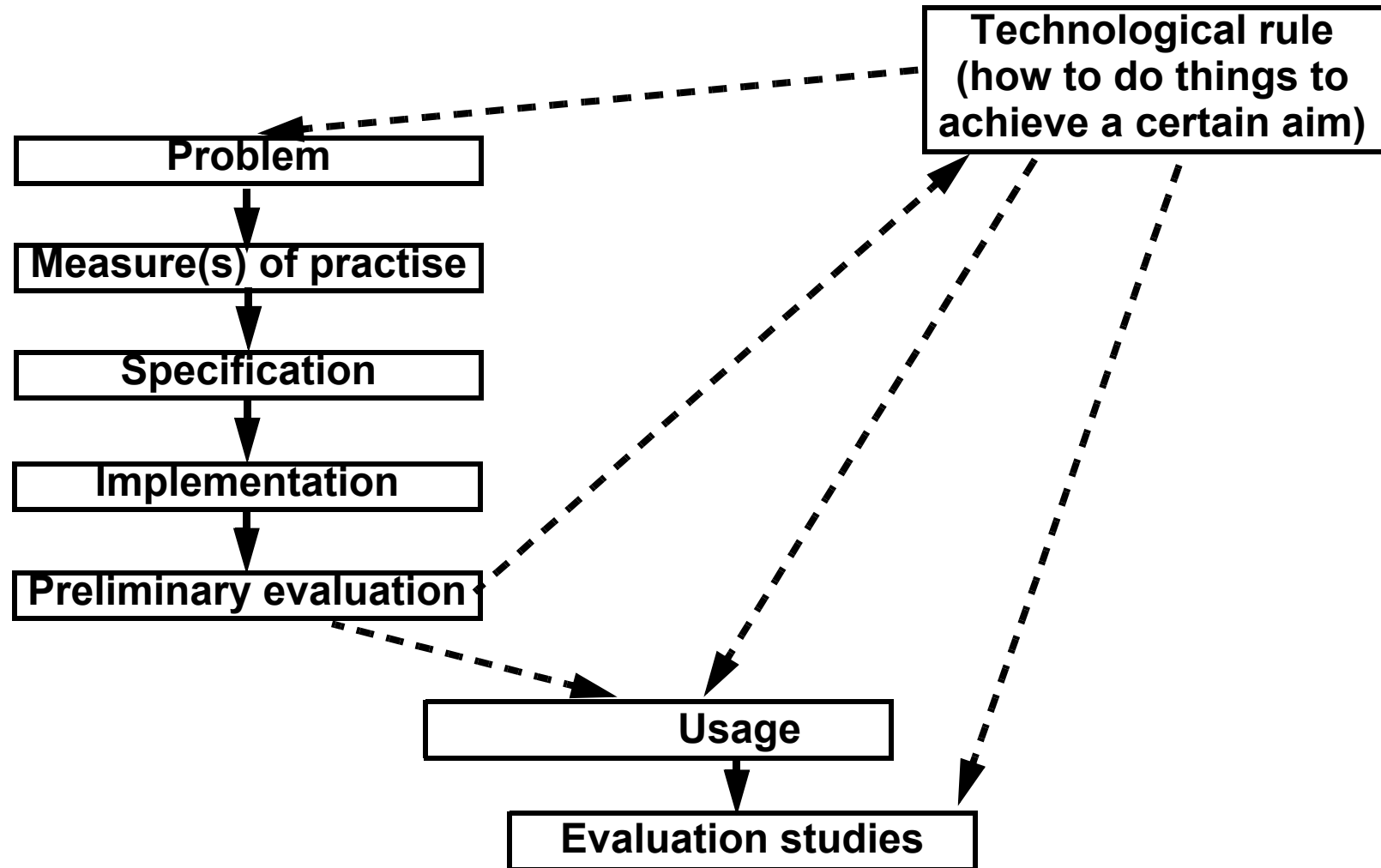


Menu

1. Key elements of a design-oriented approach:	2
2. The design process	7
3. Evaluation	12
4. Examples	16

1. Key elements of a design-oriented approach:

1.1 The global picture



- investigate at least one of the dotted lines
- Technological rule as (theory on how to things) can be input, output, or both

1.2 Ingredients of design research

(Pertti Järvinen, 2004)

Technological rules

- tell you how to do things and are dependant on other theories (and beliefs)
- Bunge (quoted by Järvinen:99): "A technological rule: an instruction is defined as a chunk of general knowledge, linking an intervention or artifact with a desired outcome or performance in a certain field of application".

Types of outcomes (artifacts, interventions):

- Constructs (or concept) form the "**language**" of a domain
- Models are sets of propositions expressing **relationships** among constructs
- Methods are a **set of steps** to perform a task (guidelines, algorithms)
- Instantiations are **realizations** of an artifact in its environment

Types of research:

- Build: Demonstrate feasibility of an artifact or intervention
- Evaluate: Development of criteria, and assessment of both artifact building and artifact usage

What does this mean ?

- There are 4*2 ways to lead interesting design research.
- Usually, it's the not the program you build that is interesting, but something behind (constructs, models, methods) or around (usage).

1.3 Instructional design rules

Exemple 1-1: The MISA/MOT/ADISA technical rule

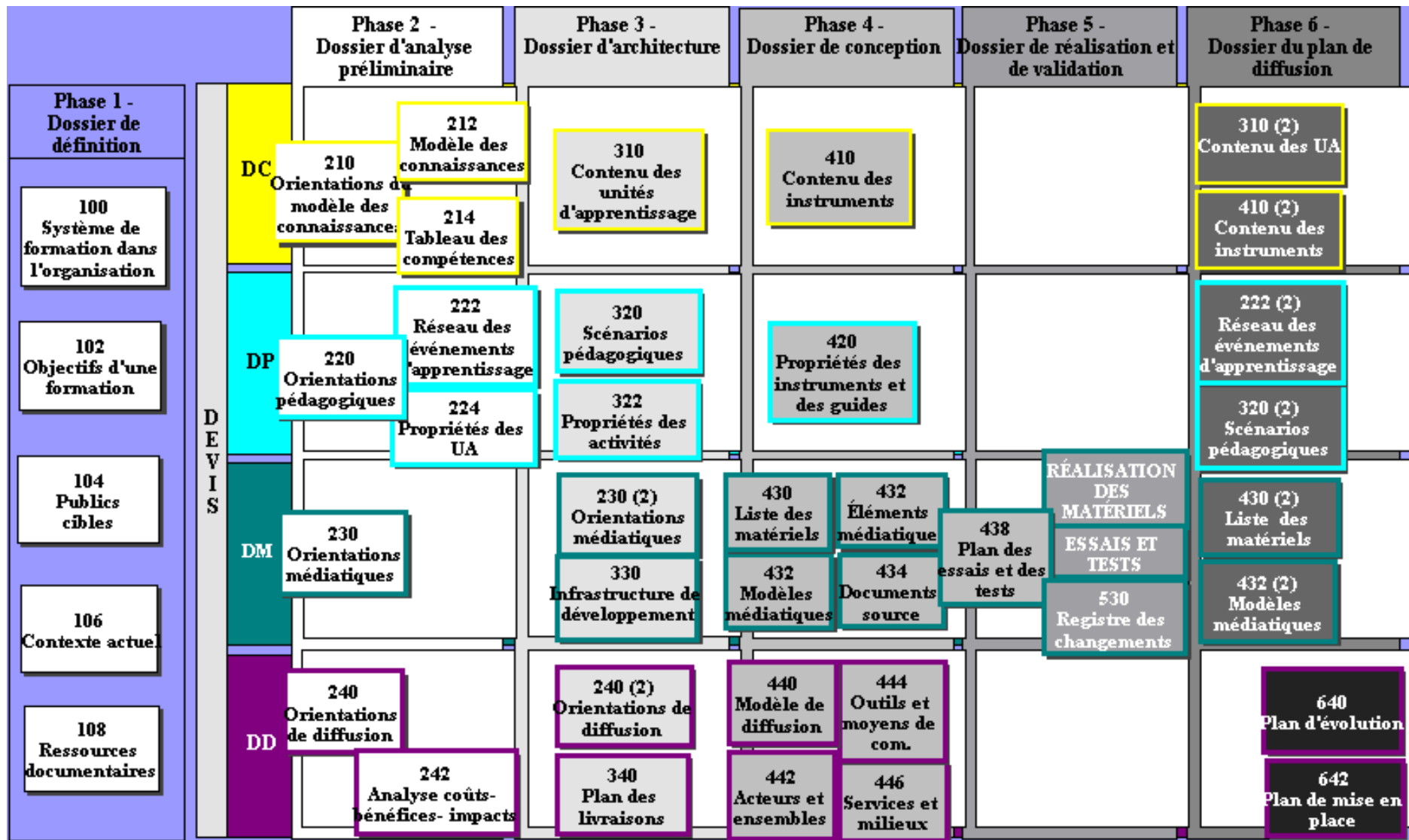
E.g. MISA/MOT/ADISA: Course designer works on "4 models"

1. **Knowledge** and **Skill** Representation
DC: Design of Content (know-that **and** know-how)
2. Application of **Teaching Methods** and Approaches
DP: Design of Pedagogical specifications
3. Specification of **Learning Materials**
DM: Design of Materials
4. **Delivery** Planning
DD: Design of Delivery

Using such a method (see next slide) is worth the effort:

- if you plan do it right (e.g. buy the MOT editor)
- if you focus on a whole course instead of difficult problems
- if you plan to train yourself in instructional design

url: <http://www.cogigraph.com>



Too much for you ?

Exemple 1-2: Gagné's 9 steps of instruction for learning

- a. **Gain attention** e.g. present a good problem, a new situation, use a multimedia advertisement.
- b. **Describe the goal**: e.g. state what students will be able to accomplish and how they will be able to use the knowledge, give a demonstration if appropriate.
- c. **Stimulate recall of prior knowledge** e.g. remind the student of prior knowledge relevant to the current lesson (facts, rules, procedures or skills). Show how knowledge is connected, provide the student with a framework that helps learning and remembering. Tests can be included.
- d. **Present the material** to be learned e.g. text, graphics, simulations, figures, pictures, sound, etc. Chunk information (avoid memory overload, recall information).
- e. Provide **guidance for learning** e.g. presentation of content is different from instructions on how to learn. Use of different channel (e.g. side-boxes)
- f. Elicit **performance "practice"**, let the learner do something with the newly acquired behavior, practice skills or apply knowledge. At least use MCQ's.
- g. Provide **informative feedback**, show correctness of the trainee's response, analyze learner's behavior, maybe present a good (step-by-step) solution of the problem
- h. **Assess** performance test, if the lesson has been learned. Also give sometimes general progress information

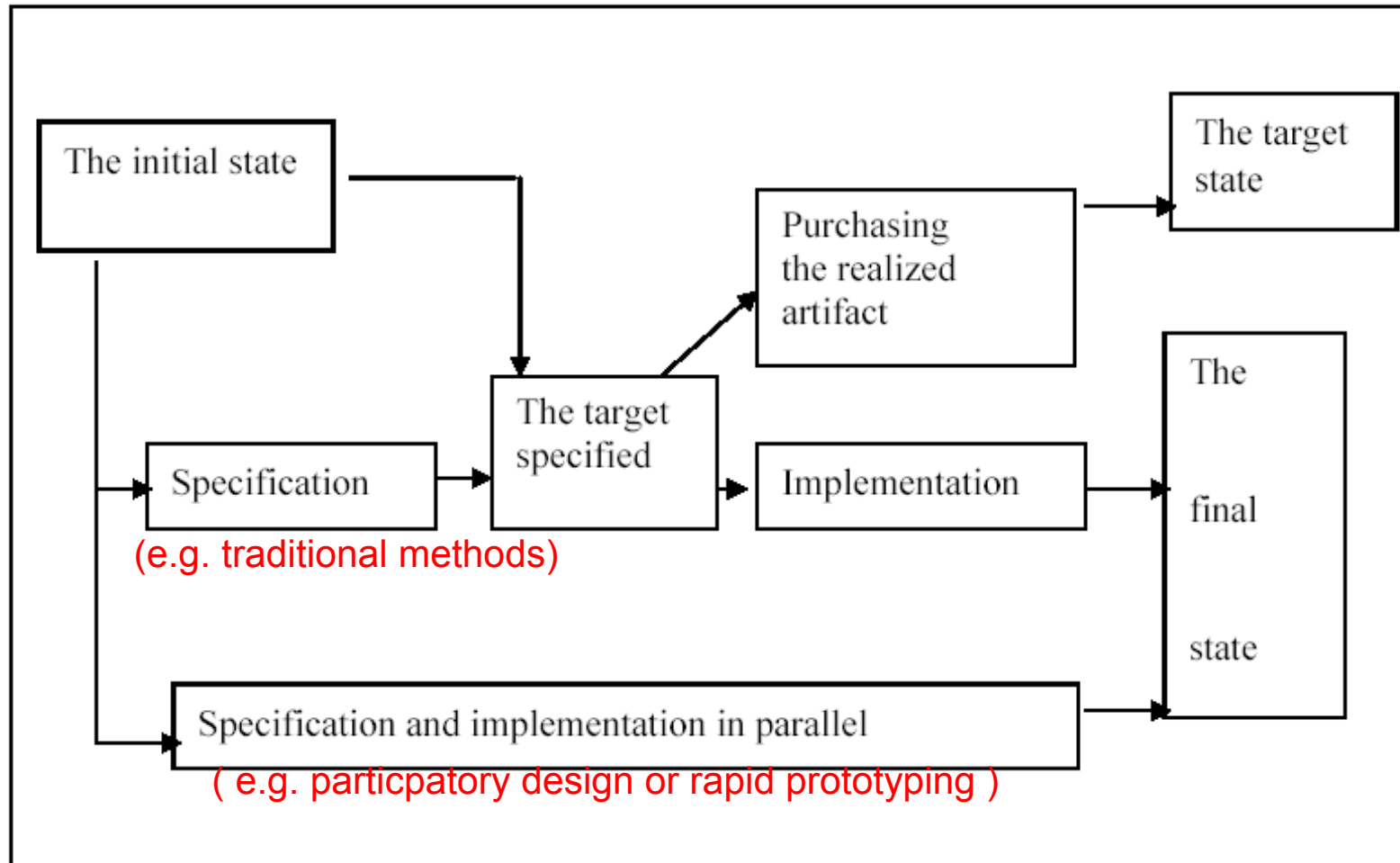
Enhance retention and **transfer**: inform the learner about similar problem situations, provide additional practice. Put the learner in a transfer situation. Maybe let the learner review the lesson.

1.4 Design rules from computer science

(not here sorry, e.g. have a look at various UML-based cases/rules)

2. The design process

Alternatives: (Pertti Järvinen, 2004: 103)



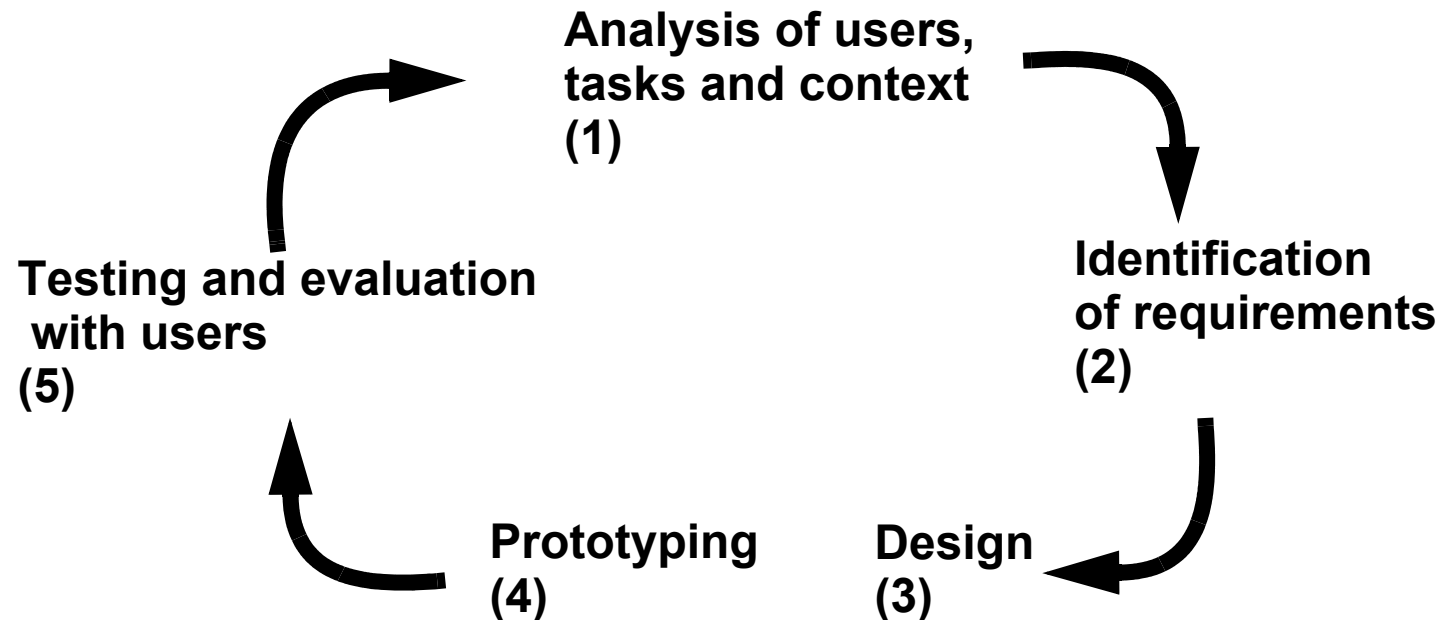
- Annotations in red by me (DKS)

2.1 The participatory design model

Note: This whole chapter draws a lot from Maria Håkansson, 2003

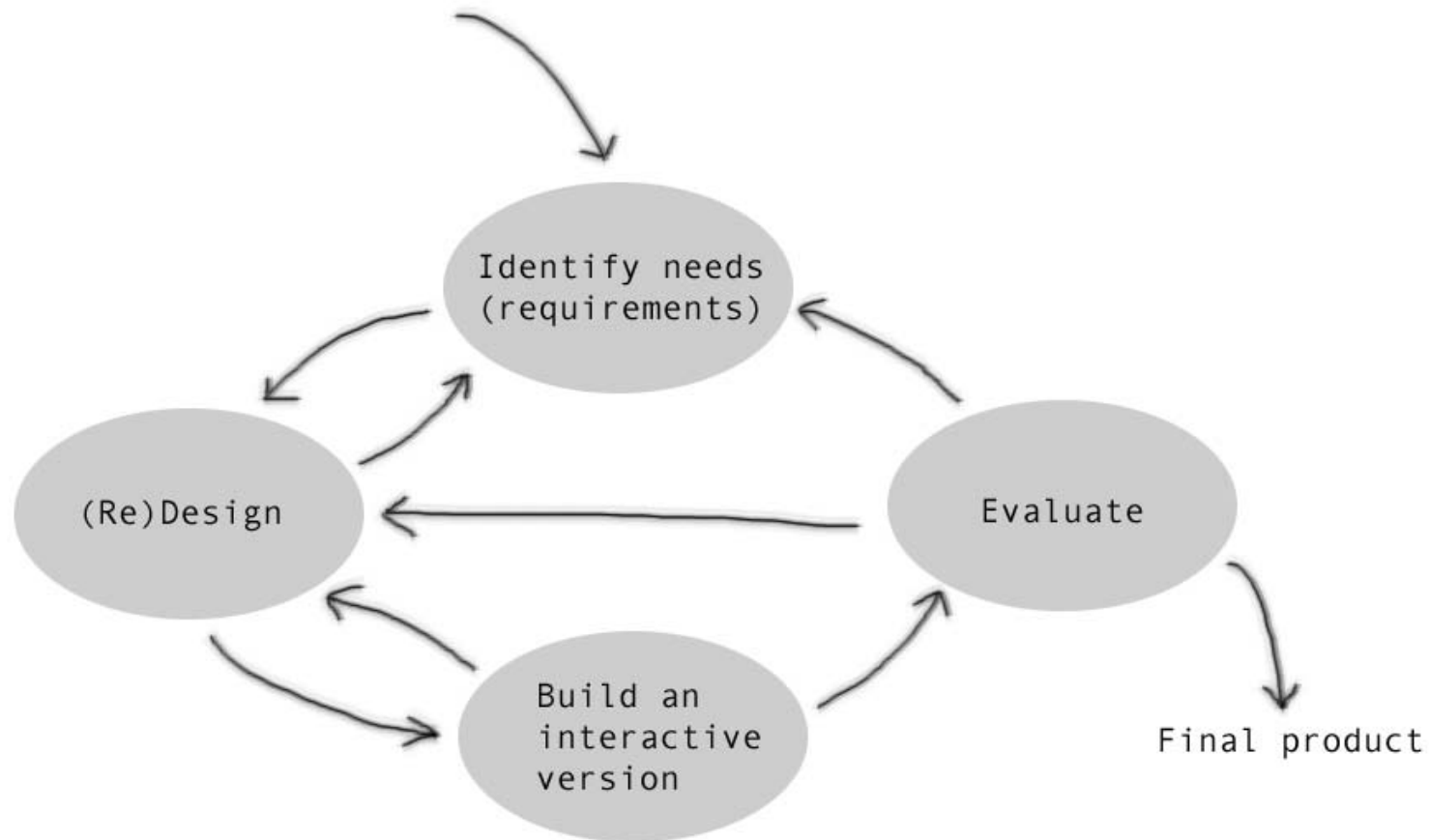
User-centred design:

- ***involves users as much as possible*** so that they can influence it
- integrates knowledge and expertise from other disciplines than just IT
- is highly ***iterative*** so that testing can insure that design meets users' requirements



adapted from Håkansson

A similar model from Preece, Rogers and Sharp (2002), figure also from Hakansson



2.2 Typical user analysis techniques

(Adapted from Håkansson, see also the modules on qualitative data gathering and analysis)

- Questionnaires
 - if user number is high
 - if you know precisely what to ask (e.g. to identify user profiles, to test hypothesis gained from in-depth studies, etc.)
- Semi-structured Interviews
 - to explore new issues
 - to let participants develop argumentation (subjective causalities)
- Focus groups
 - group interview”, collecting multiple viewpoints
- Observations/Ethnography
 - To observe work as it happens in its natural setting (observe task related workflow, interactions)
 - to understanding context (other interactions, conditions)
- Scenarios (for task description)
 - An “informal narrative description”, e.g. write **real** stories that describe in detail how someone will use your software (do not try to present specifications here !)
- Cultural probes
 - Alternative approach to understanding users and their needs, developed by Gaver (1999) ?

2.3 Definition of requirements

Different types

- Functional requirements
- Environmental requirements
- Physical, social, organizational, technical
- User requirements
- Usability requirements

2.4 Building prototypes

- Prototypes can be anything !!
- Quote: "From paper-based storyboards to complex pieces of software: 3D paper models, cardboard mock-ups, hyperlinked screen shots, video simulations of a task, metal or plastic versions of the final product" (Håkansson).
- Prototypes are of different nature according to the stage and the evolution of the design process:
 - Useful aid when discussing ideas (e.g. you only need a story-board here)
 - Useful for clarifying vague requirements (e.g. you only need some UI interface mockup)
 - Useful for testing with users (e.g. you only need partial functionality of the implementation)

3. Evaluation

3.1 Evaluation criteria

- Evaluation usually happens according to some "technological rule"

Exemple 3-1: Example: Merrill's criteria for 5 Star Instructional Design's

Not applicable to transmissive ("spray-and-pray" / or exploratory designs ("sink-or swim").

5. Does the courseware relate to real world problems?
 - a.... show learners the task or the problem they will be able to do/solve ?
 - b.are students engaged at **problem or task level** not just operation or action levels?
 - c.... involve a **progression** of problems rather than a single problem?
6. Does the courseware activate prior knowledge or experience?
 - a.do learners have to recall, relate, describe, or apply **knowledge from past experience** (as a foundation for new knowledge) ?
 - b.does the same apply to the present courseware ?
 - c.is there an opportunity to demonstrate previously acquired knowledge or skill ?
7. Does the courseware demonstrate what is to be learned ?
 - a.Are **examples consistent** with the content being taught? E.g. examples and non-examples for concepts, demonstrations for procedures, visualizations for processes, modeling for behavior?
 - b. Are learner **guidance techniques** employed? (1) Learners are directed to relevant information?, (2) Multiple representations are used for the demonstrations?, (3) Multiple demonstrations are explicitly compared?

- c. Is **media** relevant to the content and used to enhance learning?
8. Can learners practice and apply acquired knowledge or skill?
- a. Are the **application (practice)** and the post test **consistent** with the stated or implied **objectives**?
(1) Information-about practice requires learners to recall or recognize information. (2) Parts-of practice requires the learners to locate, name, and/or describe each part. (3) Kinds-of practice requires learners to identify new examples of each kind. (4) How-to practice requires learners to do the procedure. (5) What-happens practice requires learners to predict a consequence of a process given conditions, or to find faulted conditions given an unexpected consequence.
 - b. Does the courseware require learners **to use new knowledge or skill** to solve a varied sequence of problems and do learners receive **corrective feedback** on their performance?
 - c. In most application or practice activities, are learners able to access context sensitive help or guidance when having difficulty with the instructional materials? Is this coaching gradually diminished as the instruction progresses?
9. Are learners encouraged to integrate (transfer) the new knowledge or skill into their everyday life?
- a. Is there an opportunity to publicly demonstrate their new knowledge or skill?
 - b. Is there an opportunity to reflect-on, discuss, and defend new knowledge or skill?
 - c. Is there an opportunity to create, invent, or explore new and personal ways to use new knowledge or skill?

=> This is rather a list of evaluation criteria

let's now look at learning types and instructional methods before we look a design method

3.2 Evaluation methodology

Design evaluation methodology draws all major social science approaches,

e.g. Håkansson cites:

- Heuristics
- Experiments
- Questionnaires
- Interviews
- Observations
- Think-aloud

Therefore:

- have a look at my others slides :)

Exemple 3-2: Nielsons (1993) usability methods

url: <http://fdlwww.kub.nl/~krahmer/evaluation-introduction.ppt>

Table 10 Summary of the usability methods covered in Chapters 5-7.

Method Name	Lifecycle Stage	Users Needed	Main Advantage	Main Disadvantage
Heuristic evaluation	Early design, "inner cycle" of iterative design	None	Finds individual usability problems. Can address expert user issues.	Does not involve real users, so does not find "surprises" relating to their needs.
Performance measures	Competitive analysis, final testing	At least 10	Hard numbers. Results easy to compare.	Does not find individual usability problems.
Thinking aloud	Iterative design, formative evaluation	3-5	Pinpoints user misconceptions. Cheap test.	Unnatural for users. Hard for expert users to verbalize.
Observation	Task analysis, follow-up studies	3 or more	Ecological validity; reveals users' real tasks. Suggests functions and features.	Appointments hard to set up. No experimenter control.
Questionnaires	Task analysis, follow-up studies	At least 30	Finds subjective user preferences. Easy to repeat.	Pilot work needed (to prevent misunderstandings).
Interviews	Task analysis	5	Flexible, in-depth attitude and experience probing.	Time consuming. Hard to analyze and compare.
Focus groups	Task analysis, user involvement	6-9 per group	Spontaneous reactions and group dynamics.	Hard to analyze. Low validity
Logging actual use	Final testing, follow-up studies	At least 20	Finds highly used (or unused) features. Can run continuously.	Analysis programs needed for huge mass of data. Violation of users' privacy.
User feedback	Follow-up studies	Hundreds	Tracks changes in user requirements and views.	Special organization needed to handle replies.

4. Examples

Exemple 4-1: V. Synteta's master thesis

Title: EVA_pm: Design and Development of a Scaffolding Environment For Students Projects

Objectives (quotations from the thesis,

- This study is also an intervention for improving PBL efficiency. It entails the development of a Scaffolding Learning Environment (SLE1) that is trying to learn from the lessons of the past and leverage from stresses on new technologies like XML2 and the World Wide Web making a lightweight and easily portable environment.
- Most of the research for improving PBL efficiency tries to remediate specific weaknesses of PBL, but doesn't propose a complete system that supports a substantial student project through all it's phases and for all contexts.
- Our key goal was to develop a constructivist environment and a method for scaffolding students' projects (assignments) from their management up to the writing of their final report.

So, the objectives of this SLE are:

- to help students develop scientific inquiry and knowledge integration skills, to focus on important and investigate key issues;
- to support them directing investigations;
- to make students better manage the time and respect the time constraints;
- to overcome possible writer's block, or even better to avoid it;
- to help students acquire knowledge on project design and research skills;
- to improve team management and collaboration (especially collaborative editing of student groups);
- to make students reflect on their work;

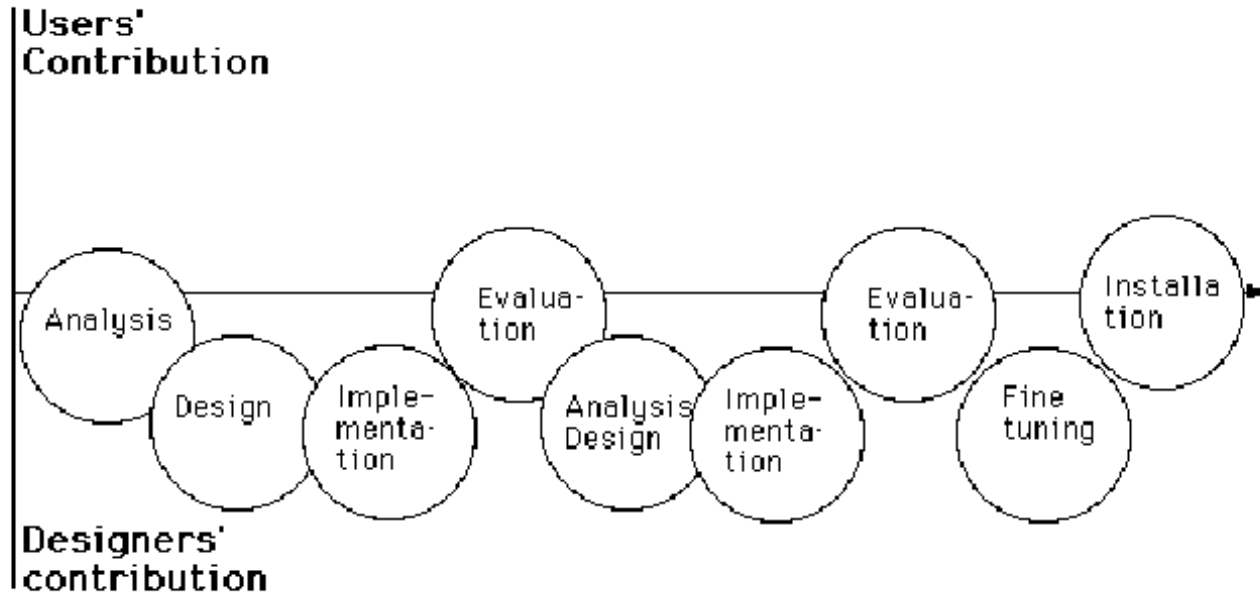
- to support the tutor's role in a PBL approach;
- to facilitate monitoring and evaluation for the tutor;
- to help the tutor verify whether knowledge is being acquired;
- to motivate the peers, and eventually to distribute the results to bigger audiences.

Research questions: see above

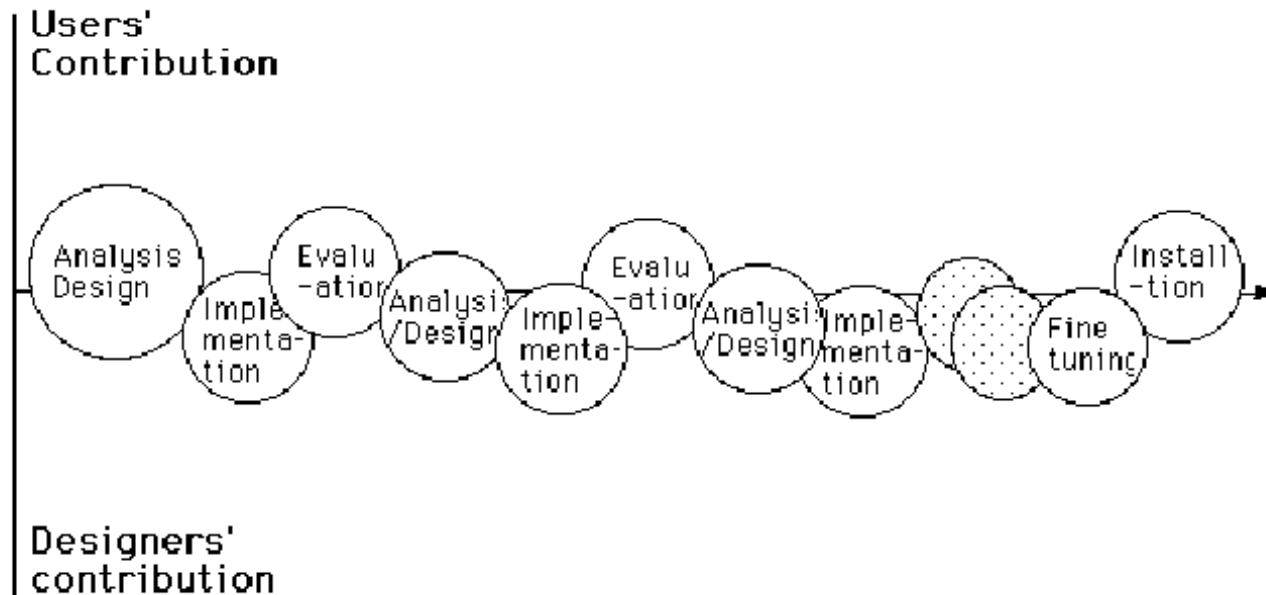
Method

- Field exploration:
 - A very important part of this research was to conceive a grammar that would model the work of an academic project. There are different sources of information that have been used to achieve this goal. (....)
- Survey of needs with a questionnaire:
 - In order to gather precious information from the key persons involved in projects, like professors and their assistants, a questionnaire was articulated in such a way that would provoke a productive discussion, leading to comments and suggestions that would improve this research. The idea was to give the questionnaire to a small sample of the unit and stop the survey when the same answers came up again.
- The development method
 - ... that has been adopted corresponds to participatory design and specifically to **cooperative prototyping**.
 - Both "prototyping" and "user involvement" (or "user centered design") are concepts that have frequently been suggested to address central problems within system development in recent years. The problems faced in many projects reduce to the fact that the systems being developed do not meet the needs of users and their organizations. [...]

Traditional Information System prototyping approaches (after Grønbaek, 1991).



Cooperative prototype approach used in this study (after Grønbaek, 1991)



Exemple 4-2: F. Radeff's master thesis

Title: Le portail personnalisable comme interface d'optimisation de ressources académiques

Research questions

La question principale de ce travail est : pourquoi les bibliothèques universitaires ne proposent-elles pas de portails personnalisables, ces derniers constituant vraisemblablement une bonne solution pour l'optimisation des ressources des bibliothèques à l'ère numérique.

Cette question peut s'articuler en 3 sous-questions :

- pourquoi pas plus de portails personnalisables ?
- pourquoi les gens ne personnalisent-ils pas ?
- la personnalisation correspond-elle à un besoin ?

Method, not clearly articulated, e.g.:

- La méthodologie retenue est une revue de la littérature, afin d'explicitier les concepts, de dégager les grands modèles et d'examiner les dispositifs existants, accompagnée d'un monitoring sur l'**implémentation partielle d'un prototype de dispositif** de portail personnalisable.
- L'implémentation partielle n'ayant pu être menée à terme, le monitoring a été supprimé et j'ai élargi le champ de la revue de littérature. L'analyse est donc qualitative, les données quantitatives initialement prévues n'ayant pu être collectées.
- Le matériel recueilli lors du prototype MyBCU ainsi que l'expérience acquise en 2001-2002 comme webmaster [...] m'ont néanmoins permis de tenter de répondre aux questions initiales.

