IKT og Videnrepræsentationer -
ICT and Knowledge Representations.

1. Systems Development Overview

Cand. Scient. Bygningsinformatik.
Semester 2, 2010.

7.9.2010 (6.9.2010) [3.9.2010]

CONTENT

- System development
- Knowledge representations
- Functional and component building systems
- User involvement in design
- Ontologies
- System Evaluation

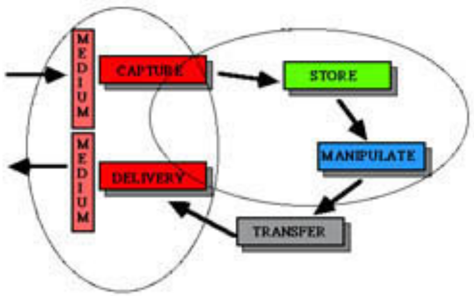
BUILDING INFORMATICS RELATED AREAS

User Environment (UE) design
User needs capture
Requirements specs
Contextual design
Usability/evaluation

Knowledge Management (KM)
Intranet/extranet specifications
ICT and change strategy
Knowledge and experiences
discovery, capture, storage
and transfer
Information QA

Intelligent Buildings (IB)
IB design
Services and systems
Networks
Facility management
Intelligent city

Computer Supported Collaborative Working (CSCW)
Virtual workspaces
Sync/async communication
Distributed collaboration
Storytelling

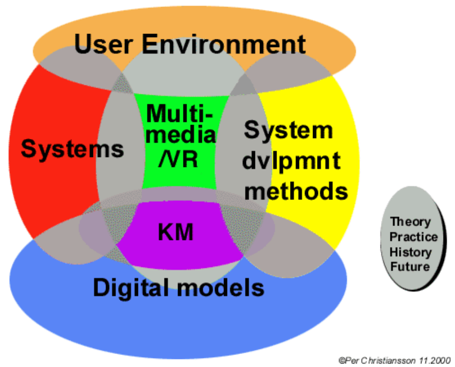


Building simulations
Building systems simulations
Building systems integration

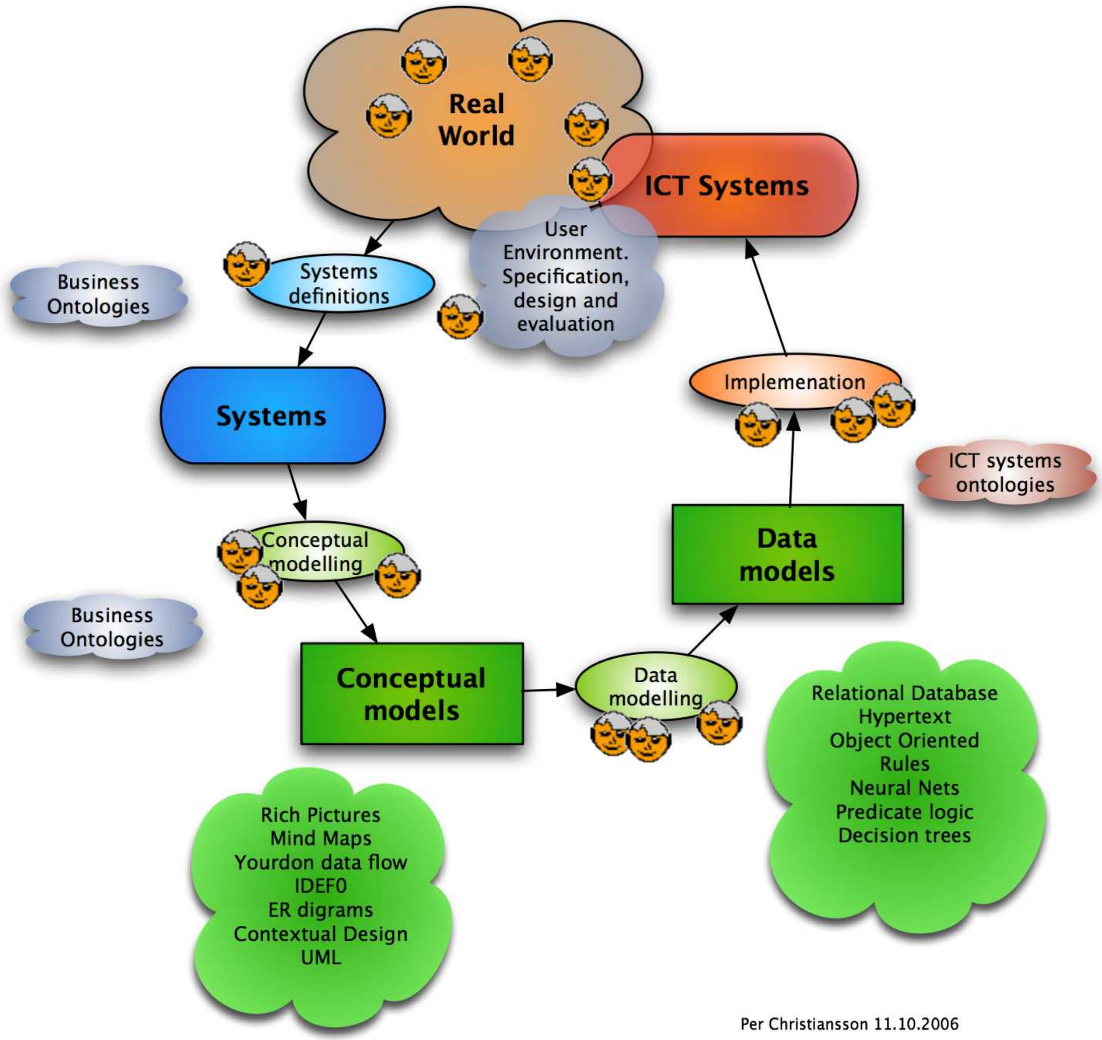
Virtual Buildings (VB)
CAD
Product and process
models and modelling
Classification
Conceptual modelling
3D geometric modelling

Human Computer Interaction/ Multimedia (HCI/MM)
HCI design
Multimodal interfaces
MM formats
Computer graphics
Virtual Reality

Knowledge Representations (KR)
Relational databases
Object Oriented
Logic
HyperText
XML
Semantic Web



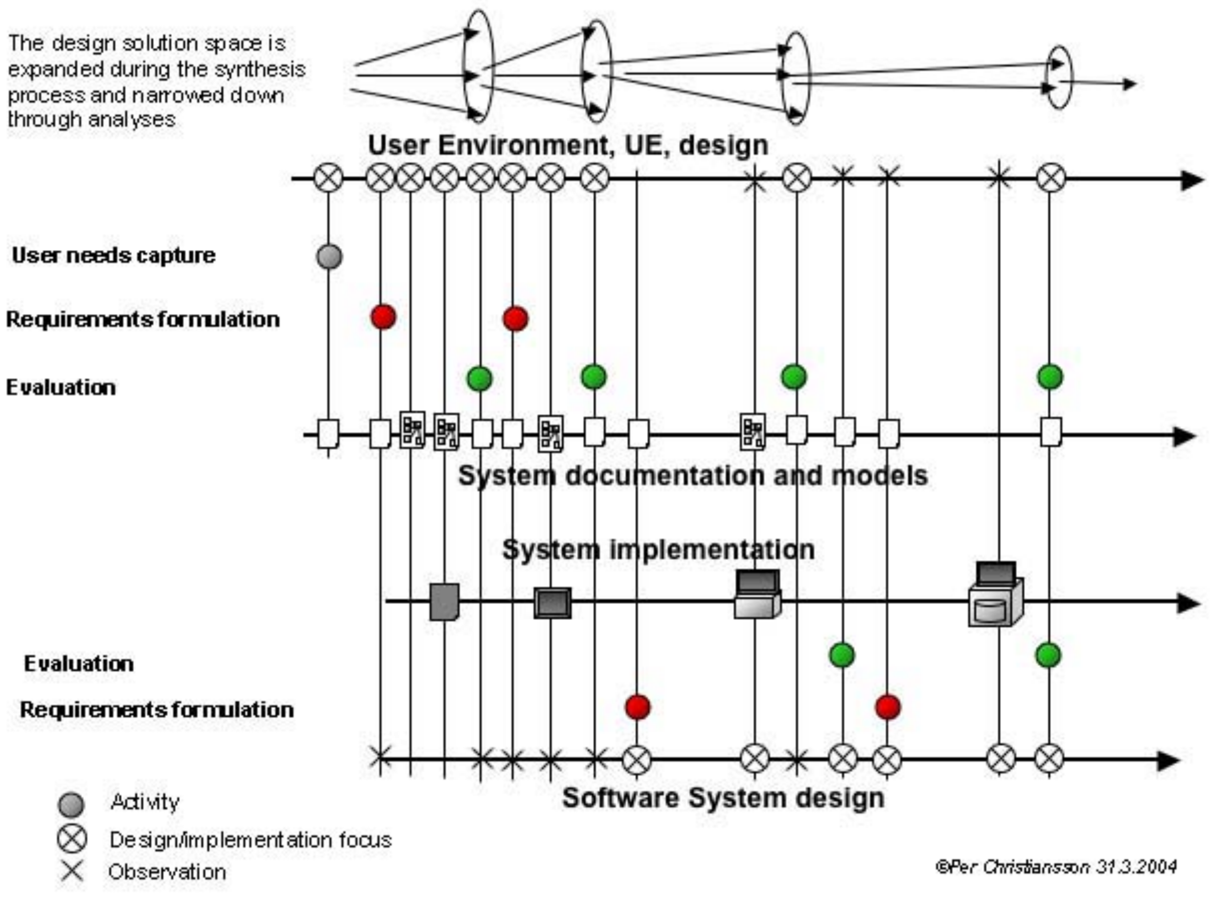
SYSTEM DEVELOPMENT



In the real world we identify activities, things, processes, context, and persons.

The real world can be described as (interrelated) systems (no de-facto structure is available today) to accomplish different functions e.g. a comfort system to provide personal living and working quality, personal transport system, load carrying building system, escape system, and communication systems (collaboration, knowledge transfer, mediation, virtual meeting).
(Christiansson & Svidt & Sørensen, 2009).

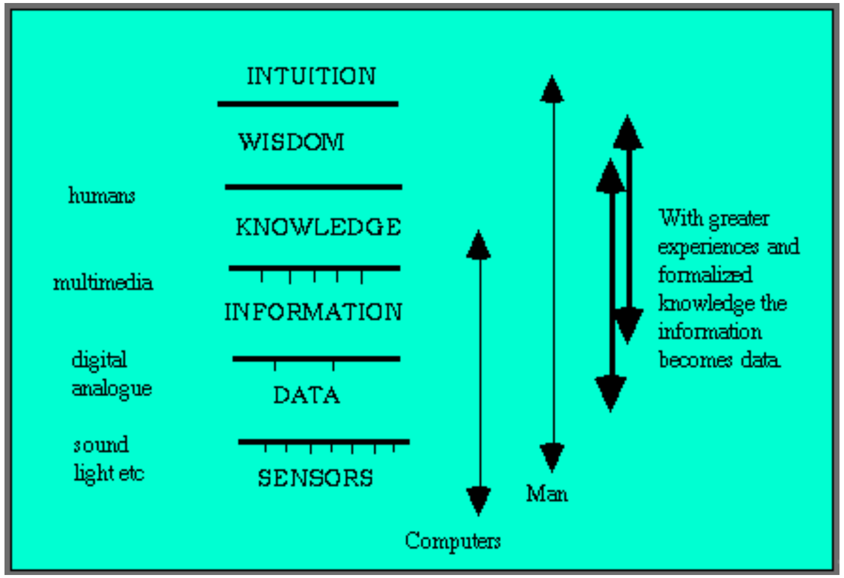
SYSTEM DEVELOPMENT



The early design process focuses on user environment, UE, design/implementation and the later phases on software development and implementation.

The UE design including user needs capture and user requirements formulations can be supported by contextual design methodology. Different evaluation paradigms can be used as design/implementation progresses.

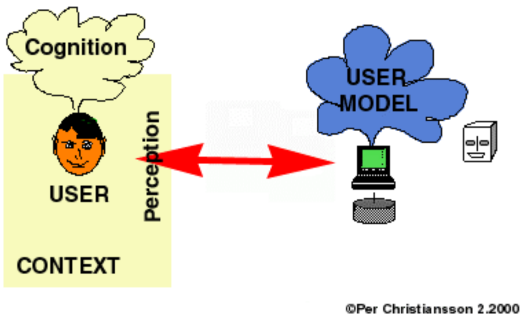
FROM DATA TO INTUITION



The abstraction hierarchy of knowledge. Knowledge has a limited duration in time. (The world is not flat any longer).

From Christiansson, P, "The Formalization process in Global Knowledge Handling". Research Directions for Artificial Intelligence in Design. (eds) J.S. Gero and F. Sudweeks. Key Centre of Design Computing, University of Sydney.(the Fourth Workshop on Research Directions for Artificial Intelligence in Design. University of Twente. Enschede, The Netherlands. January 6 1995). (pp 23-34).(Invited position paper)

PERCPETION - COGNITION



The computer contains a more or less explicitly described user model.

Merriam Webster:

- *Cognition* = the act or process of knowing including both awareness and judgement; from co- + gnoscere to come to know

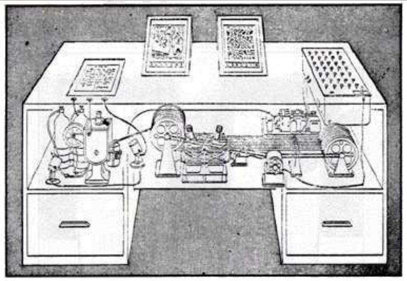
* *Perception* = act of perceiving; awareness of the elements of environment through physical sensation
  (Percieving = to become aware of through the senses)

Gardners Multiple Intelligences

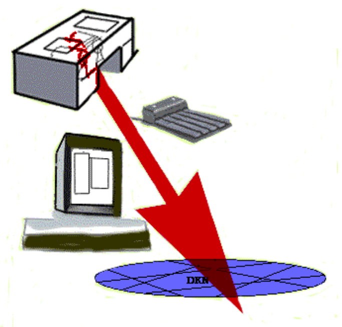
http://www.howardgardner.com/MI/mi.html
http://www.funderstanding.com/content/multiple-intelligences

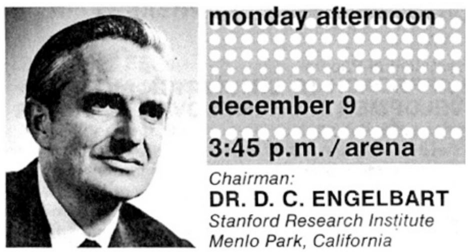
1. Verbal-Linguistic--The ability to use words and language
2. Logical-Mathematical--The capacity for inductive and deductive thinking and reasoning, as well as the use of numbers and the recognition of abstract patterns
3. Visual-Spatial--The ability to visualize objects and spatial dimensions, and create internal images and pictures
4. Body-Kinesthetic--The wisdom of the body and the ability to control physical motion
5. Musical-Rhythmic--The ability to recognize tonal patterns and sounds, as well as a sensitivity to rhythms and beats
6. Interpersonal--The capacity for person-to-person communications and relationships
7. Intrapersonal--The spiritual, inner states of being, self-reflection, and awareness

MULTIMEDIA - HYPERMEDIA - HYPERTEXT



Memex in the form of a desk would instantly bring files and material on any subject to the operator's fingertips. Slanting translucent viewing screens magnify supermicrofilm filed by code numbers. At left is a mechanism which automatically photographs longhand notes, pictures and letters, then files them in the desk for future reference (*LIFE 19*(11), p. 123).





Hypertext defined.
1945. Vannevar Bush (more) and the Memex. Advisor to president Roosevelt

1968. Douglas 'Doug' Engelbart, 'mouse', multiple windows, distributed collaboration (teleconferencing), wordprocessing with commands and use of 'mouse' . In his Augment Project tried to introduced tools that should not disturb 'high-level' thinking (compare to near real access to models in Virtual Reality emvironments).
[http://sloan.stanford.edu/mousesite/1968Demo.html]

1964 Ted Nelson talks about 'hypertext'. The Xanadu project aiming at linking all text produced with backward references, versioning and linked access, multiple forms of content (not only text), classification of text (author,..). Autodesk made a prototype around 1990.

1967. Nicholas Negroponte forms the Architecture Machine Group in the Architecture Department at MIT and later gets his own building, Media Lab, at MIT. Example at Architecture (around 1977) was the interactive wall where you could point and move things (video screen back wall projection).

Xerox PARC in Palo Alto presents a personal computer, Alto, later with XeroxStar operating system with windows, mouse and icons. Apple Lisa takes up the ideas 1983. (Steven Jobs regrets he did not stay 20 minutes more at Xerox to hear about TCP/IP. Instaed Apple developped AppleTalk).

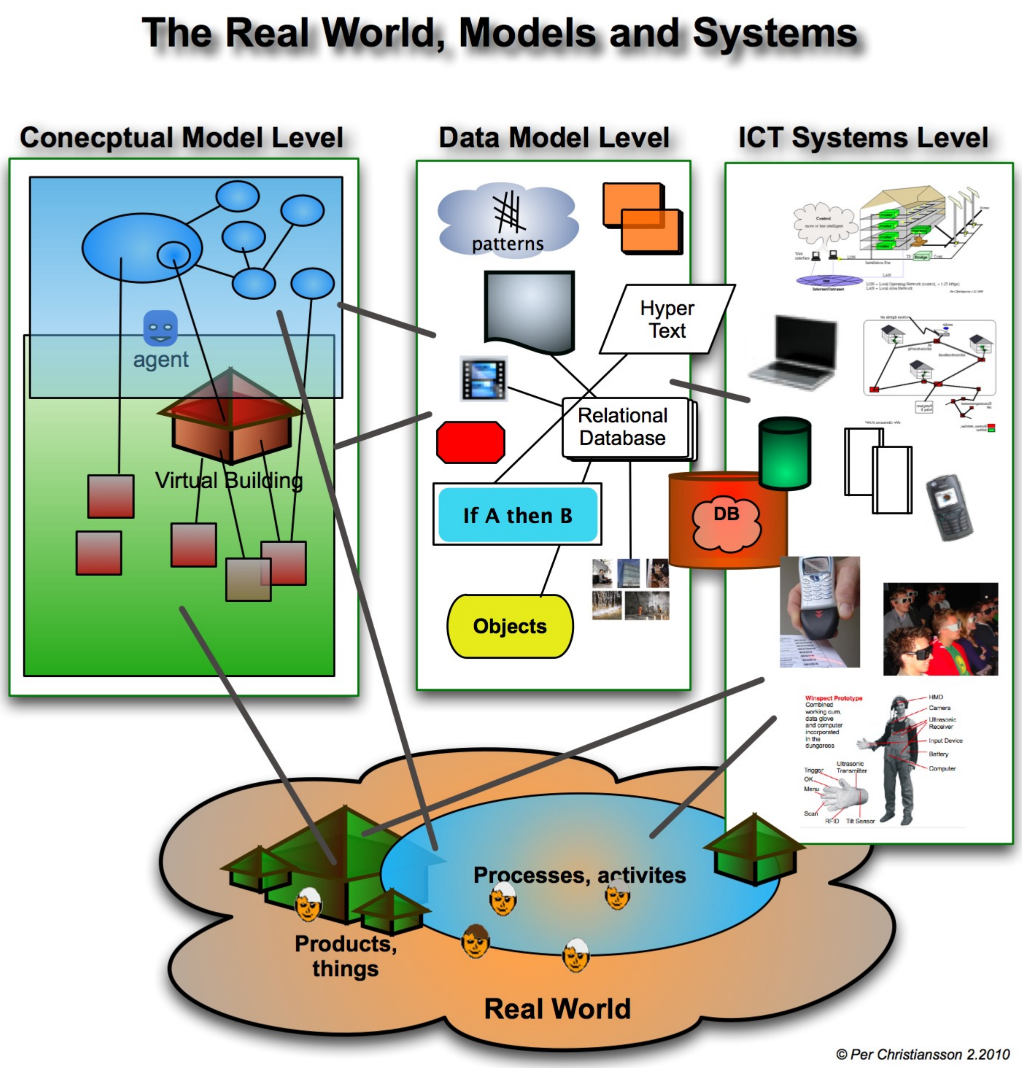
1978 the Aspen Movie Map interactive laser disk (a type of interactive movie) (MIT Media Lab) [http://www.media.mit.edu/]

1983 MIT educational computer network, the Athena project.

1987, Apple releases HyperCard. [http://en.wikipedia.org/wiki/HyperCard]

1993, World Wide Web

MODELS OF THE REAL WORLD



The HOLISTIC view
The holistic view.

We use different kinds of ICT support in the building process and the built environment.

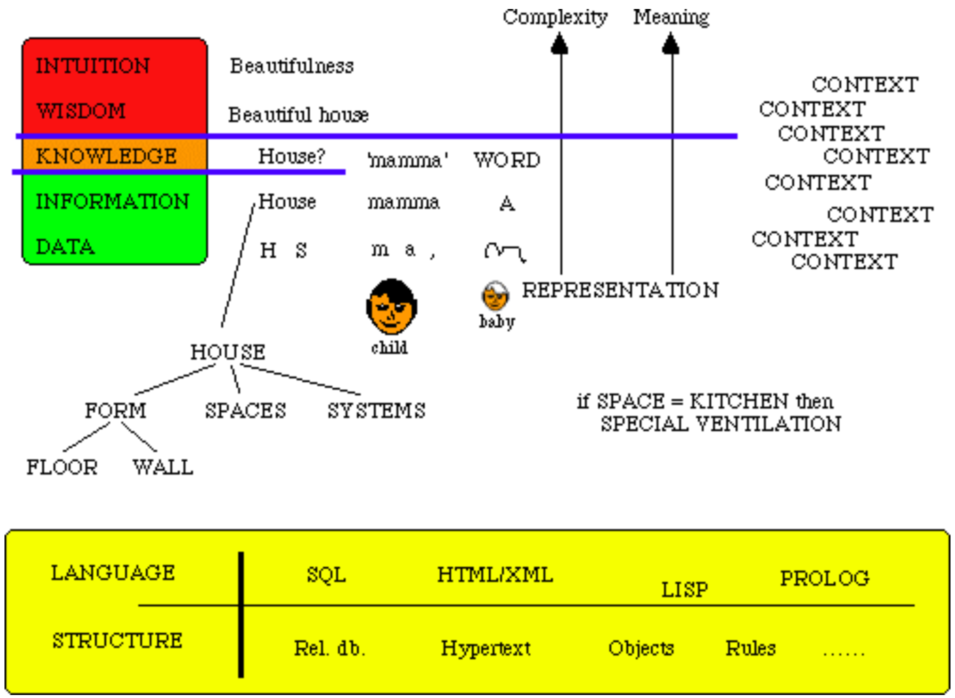
The ICT systems support different functionalities in the building process and built environment.

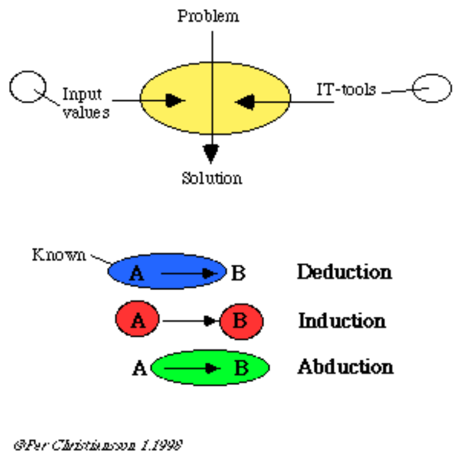
KNOWLEDGE REPRESENTATIONS

Five different roles of a knowledge representation (Dragan et al., 2006).

- *substitute* for things from the real world inside an intelligent entity
- *making a set of ontological commitments* about the real world and a selection
- a *fragmentary theory of intelligent reasoning*
- a *medium for efficient computation*

KNOWLEDGE REPRESENATION. STRUCTURE - ACCESS

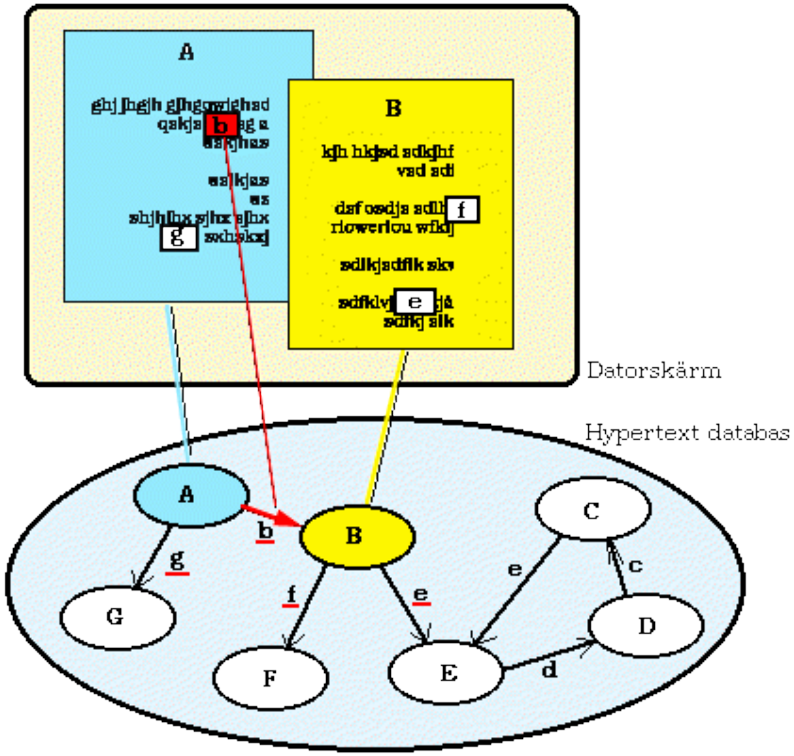




Knowledge is stored in humans heads and mimiced in computers

Combinations possible

KNOWLEDGE REPRESENATION. HYPERTEXT.



From Christiansson P, 1989, "KBS-MEDIA projektet vid Lunds Universitet". ORDO, LNTH 1/89. (11 pp) after Conklin J, 1987. "Hypertext: An Introduction and survey". IEEE Computer, September 1987. pp 17-41.).

KNOWLEDGE REPRESENATION. LOGIC.

language(X,prolog).
X = logic

The logic programming language Prolog was created around 1970 and is based on *predicate logic*. It can be used for symbolic computation applications such as mathematical logic, problem solving, artificial intelligence applications and databases. In the examples below we have used JB Prolog.

In Prolog we write rules in the form,
Conclusion <- Conditions.

:pc_family.pro

```
/* Example 'Family' written in Prolog. */
/* Per Christiansson. October 2000 */


/* The context dependent facts part  */

father(per,linus).
father(per,jens).
mother(ulla,linus).
mother(ulla,jens).

father(linus,putte).

father(lennart,per).
father(lennart,lena).
father(lennart,monica).
father(lennart,lars).
mother(margot,per).
mother(margot,lena).
mother(margot,monica).
mother(margot,lars).

father(leif,sanna).
father(leif,frida).
father(leif,klara).
father(leif,olle).
mother(lena,sanna).
mother(lena,frida).
mother(lena,klara).
mother(lena,olle).

father(lars,max).
father(lars,jon).
father(carina,max).
father(carina,jon).

father(kurt,ulla).
father(kurt,thomas).
father(kurt,claes).
mother(inga,ulla).
mother(inga,thomas).
mother(inga,claes).

father(thomas,lisa).
mother(agneta,lisa).
```

```
father(thomas,lisa).
mother(agneta,lisa).


/* The general context independent rules  */

syskon(X,Y) :- father(U,X), father(U,Y), mother(V,X), mother(V,Y), X\==Y, U
parent(X,Y) :- father(X,Y); mother(X,Y).


cousin(X,Y):- parent(U,X), parent(V,Y), syskon(U,V), X\==Y.
fatherfather(X,Y):- father(X,U), father (U,Y).
fathermother(X,Y) :- mother(X,U), father (U,Y).
motherfather(X,Y) :- father(X,U), mother(U,Y).
mothermother(X,Y) :- mother(X,U), mother(U,Y).

parents(X,Y,Z) :- father(X,Z), mother(Y,Z).


ancestor(X,Y) :- parent(X,Z), parent(Z,Y).
ancestor(X,Y) :- parent(X,Z), ancestor(Z,Y).
```

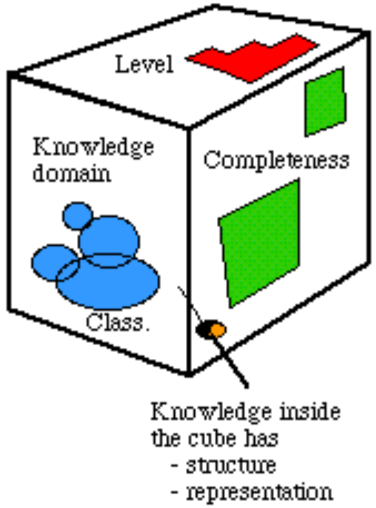
Con

```
> ancestor(X,linus).
 X = lennart
;
 X = kurt
;
 X = margot
;
 X = inga
;
> ;
```

Console

```
JB-Prolog 2.1#2, 01/24/94
>
':pc_family.pro' consulted, 33 msec.
> cousin(linus,Z).
 Z = max
;
 Z = jon
;
 Z = sanna
;
 Z = frida
;
 Z = klara
;
 Z = olle
;
 Z = lisa
;
>
```

KNOWLEDGE REPRESENTATIONS



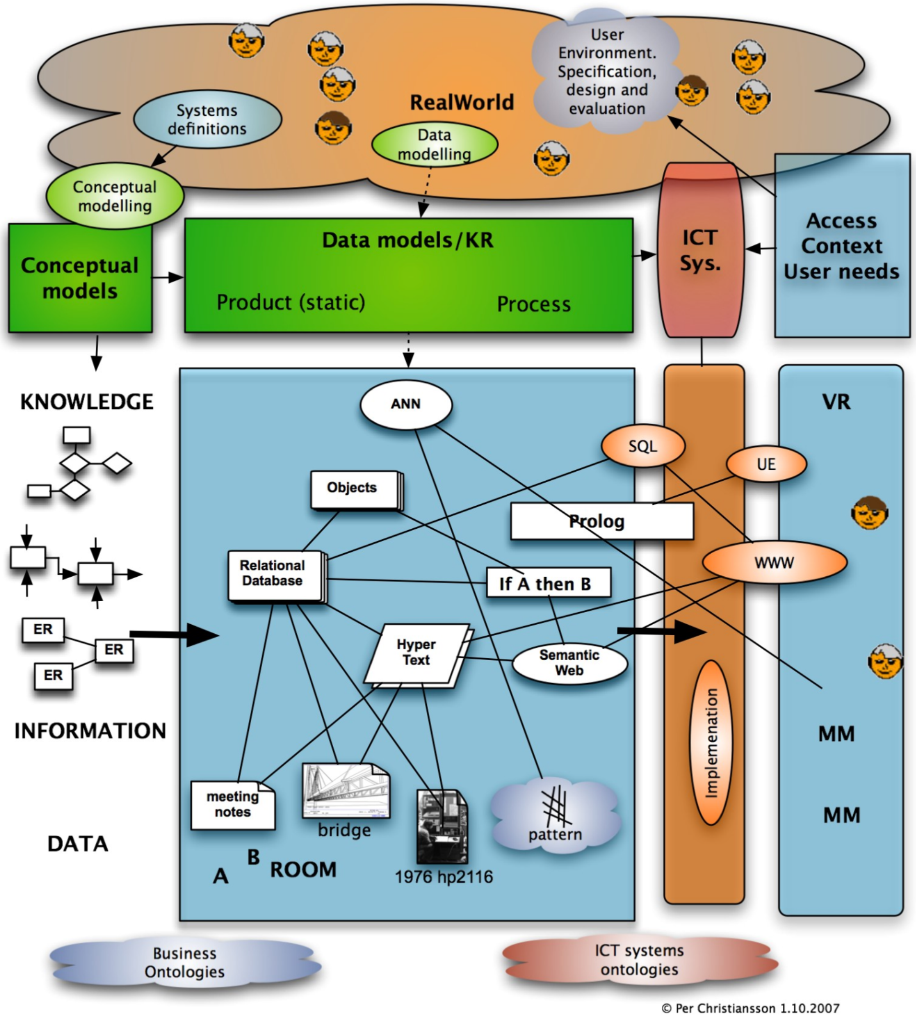
Level (deepness)	Completeness	Domain(s)
Shallow (some explnantion)	Random coverage	Scientific areas
Some deepnes (alt. answers)	Sufficient	Context dependent
Depth (with analyses)	Very complete	

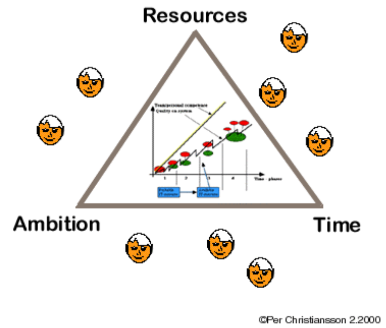
©Per Christiansson, 8.1996

Knowledge accessed from a node may be marked according to level, completeness and domain

From Christiansson P, 1997, "Experiences from developing a Buiding Maintenance Knowedge Node". CIB Proceedings W78 Workshop Cairns 9-11 July 1997.

SYSTEM DEVELOPMENT

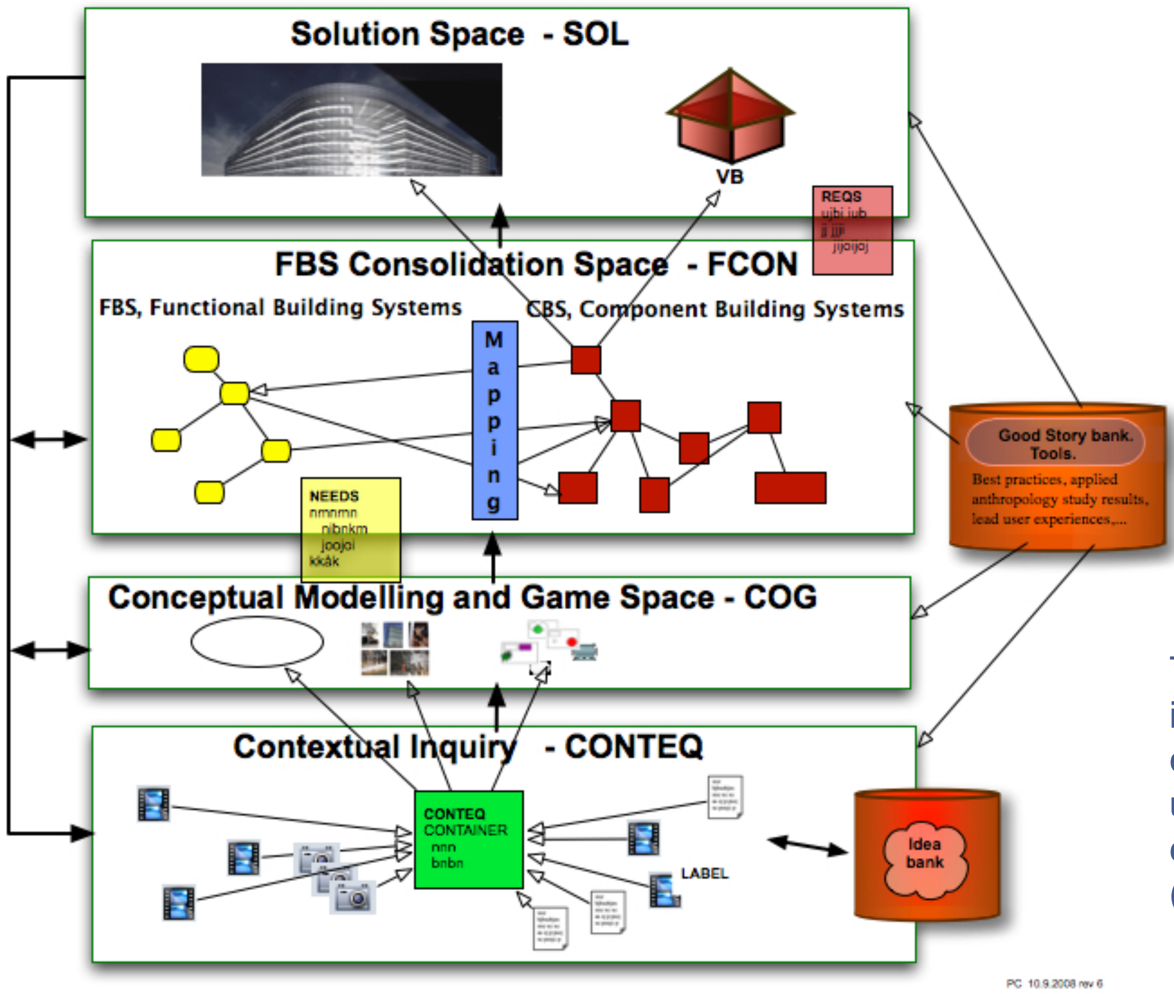




Always achieve a good balance between Time, Ambition and Resources.

From the real world to implemented systems in use

VIC-MET DESIGN SPACES



The method supports user involvement in every phase of the construction process and with a unique setup depending on design context.
(Christiansson et al. 2009a).

VIC-MET DESIGN SPACES

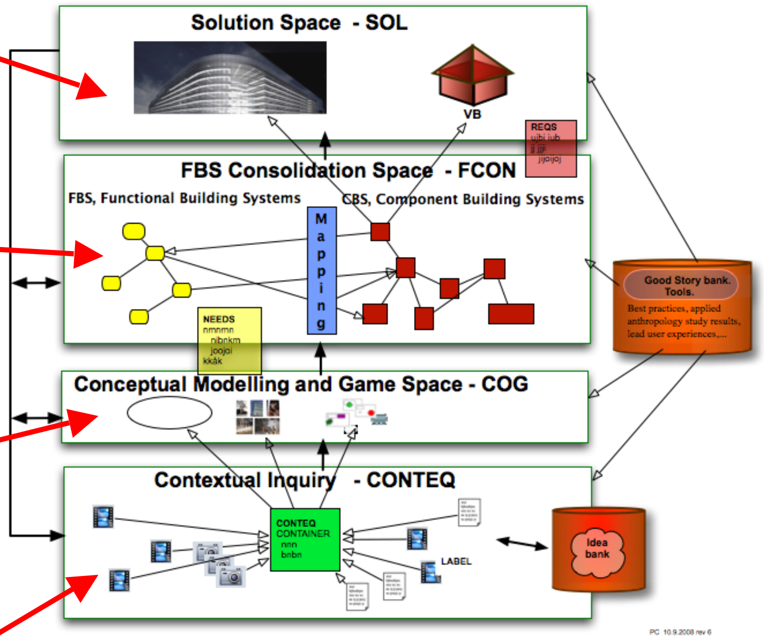
Activities in the Solution space (SOL)
- 3D virtual building modeling of (alternative) solutions.
- End user evaluation of solutions.
- Documentation of end user feed-back.
- Allocate tools from the ICT tools bank.
- Choose solution(s) or return to the FCON, COG or CONTEQ space.

Activities in the Functional Building Systems Consolidation space (FCON)
- Needs consolidation, weighing and listing.
- Project vision formulation.
- Prioritizing needs.
- Mapping of Functional Building Systems (FBS) and Component Building Systems (CBS).
- Listing of requirements on Component Building systems.
- Component Building System modeling.
- Allocate tools from the ICT tools bank.

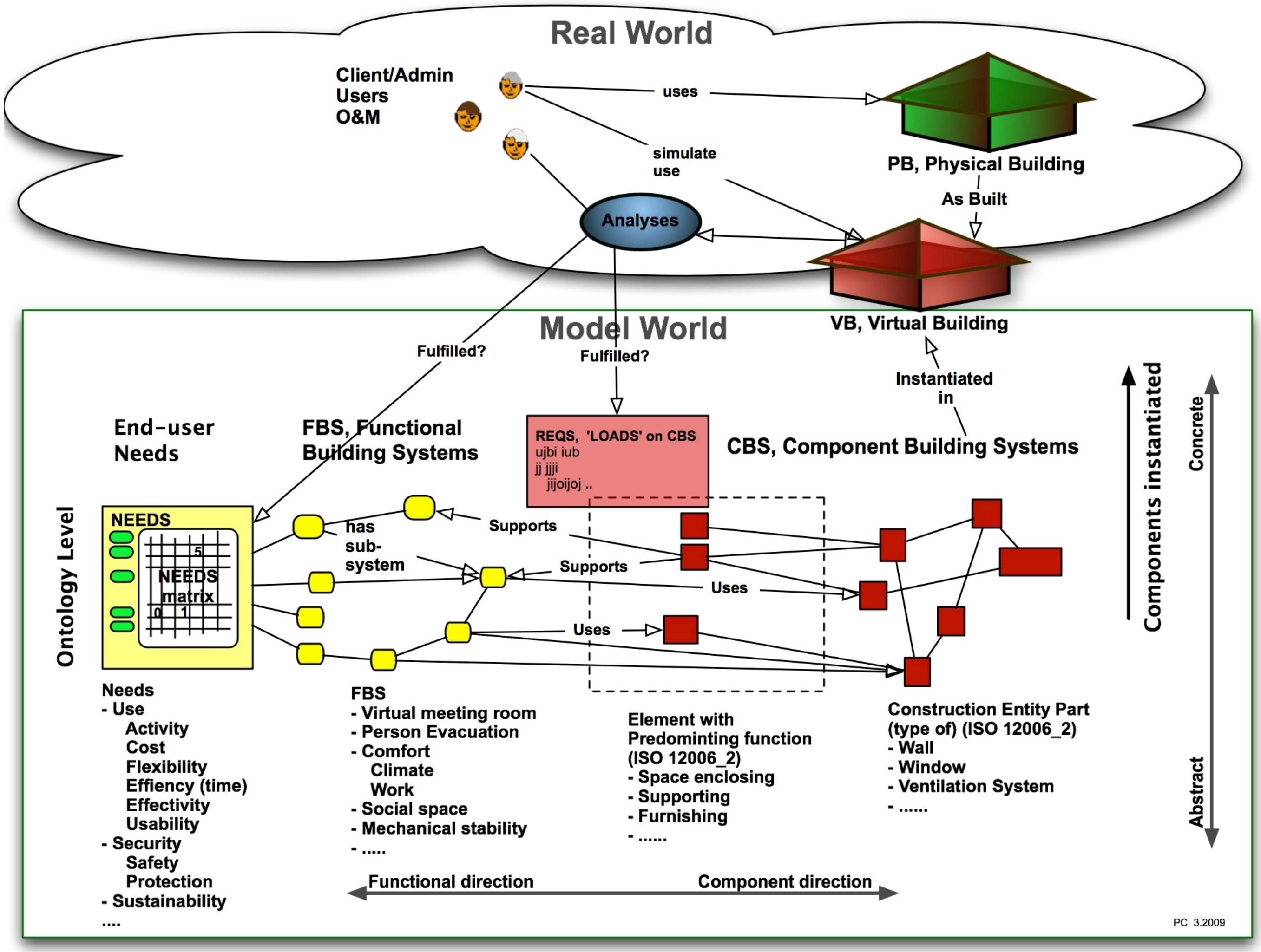
Activities in the Conceptual Modeling and Gaming space (COG)
- Develop conceptual models (e.g. using Contex-tual design methodology).
- Needs listing.
- Common values development.
- Functional Building Systems specification.
- Creative/Innovative design.
- Allocate tools from the ICT tools bank.

Activities in the Contextual Inquiry space (CON-TEQ)
- Formulate Design/Innovation domain.
- Set up design team including proper end-users groups.
- Plan the whole design process.
- Identify/allocate resources such as Idea bank, Best practice, Contextual Inquiry Bank.
- Allocate tools from the ICT Tools Bank.
- Perform contextual inquiry including needs capture.

(Christiansson et al. 2009a).



FUNCTIONAL and COMPONENT BUILDING SYSTEMS

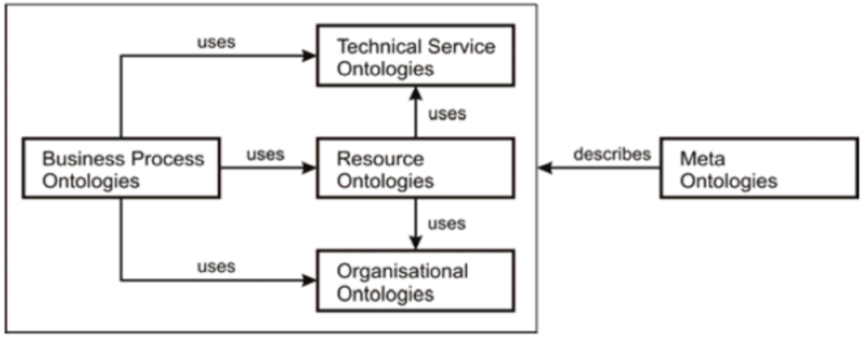


Formalization of the building design process. References are made to (ISO 12006-2, 1001). From (Christiansson & Svidt & Sørensen 2009)

ONTOLOGIES

The probability for a successful system development is highly increased if we can agree on concepts and their relations that is, we have a common linguistic reference frame. We call such a description an ontology.

Within knowledge engineering the term has been widely discussed in the 1990's. Tom Gruber's definition is the best known (Gruber, 1993): "*An ontology is an explicit specification of a conceptualization*." For use within IT in construction the similar but more detailed definition by (DLI Glossary, 1998) is also a good definition: "*An ontology is an explicit formal specification of how to represent the objects, concepts, and other entities that are assumed to exist in some area of interest and the relationships that hold among them.* Example below:

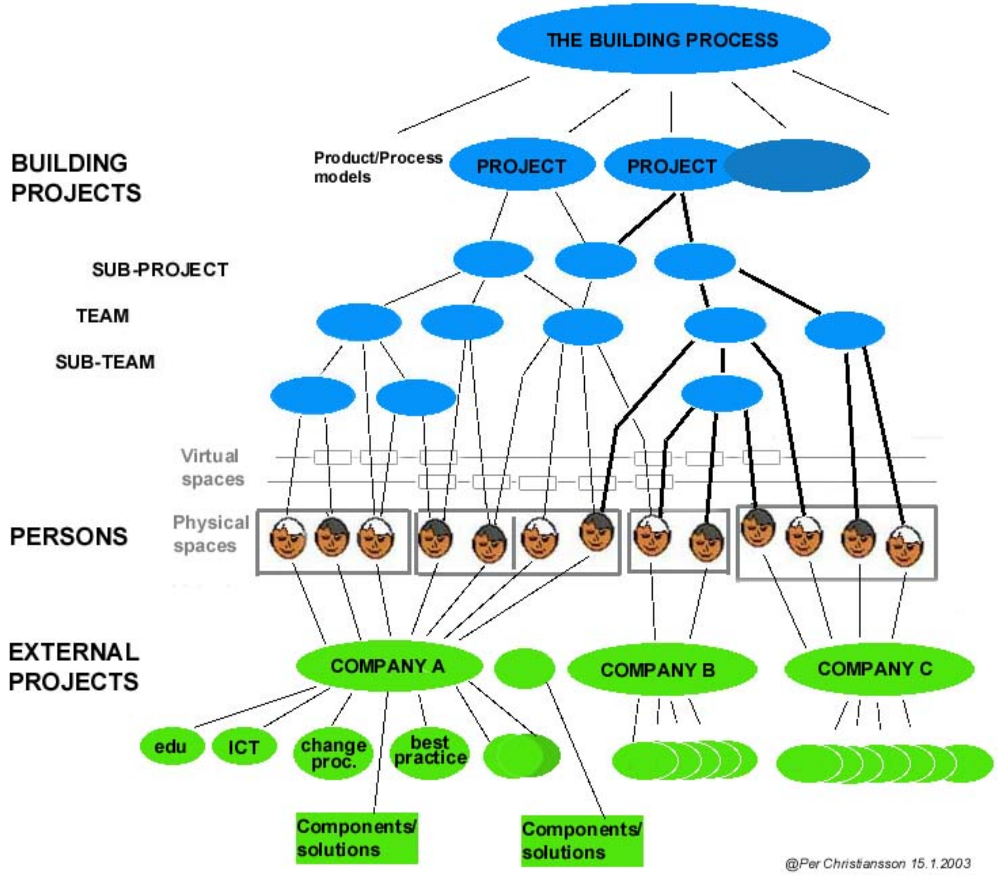


Organisational
- Context
- working task
- Roles, competences
- User needs (person, team, organisation, society)
- Co-opertation (type, form, language)
- Actitvit

Resources
- Models (Virtual Building, processes, project external, representations, level)
- Functional Building Systems (FBS)
- Component Building Systems (CBS)
- Building classifications
- Communication standards and formats
- Tools (modelling, cooperation, analyses, learning,...)
- Building system components (virtual, physical)

Technical Service
- I/O devices
- Communication channels
- Web services

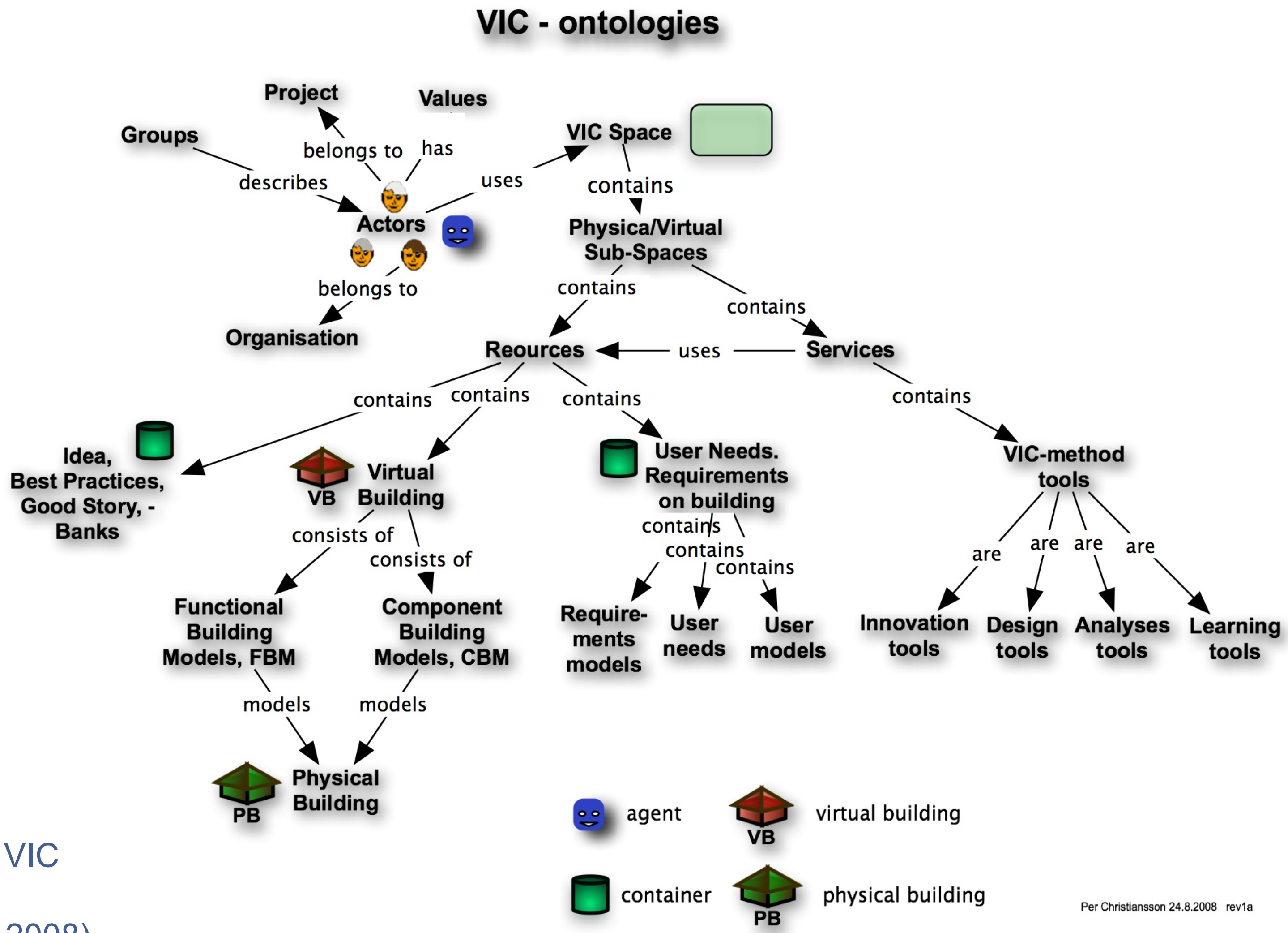
KNOWLEDGE REPRESENTATIONS - ORGANISATIONAL VIEW



Organisational view on internal and external building project actors, activities and attached information containers.

From Christiansson, P, 2003, "Next Generation Knowledge Management Systems for the Construction Industry". Auckland, New Zealand, April 23-25, 2003. CIB W78 Proceedings 'Construction IT Bridging the Distance", ISBN 0-908689-71-3. CIB Publication 284. (494 pages). (pp. 80-87).

ONTOLOGIES



The first version of a VIC meta ontology (Christiansson et al., 2008)

SYSTEM EVALUATION

Usability: the extent to which a product can be used by specific users to achieve specified goals with effectiveness, efficiency and satisfaction in a specified context of use." (from ISO 9241-11, Guidance on Usability, 1998)

- *effectiveness* (the system can be used to fulfill required tasks)
- *efficiency* (how much work/time is required to fulfill the tasks)
- *user friendliness* (how well do the system adopt to the user expectations on the system. Merriam Webster Collegiate Dictionary - easy to learn, use, understand, or deal with)

(Faulkner, 2000) defines

usability engineering (the whole process of producing usable products from requirements gathering to installation) (Other authors such as Holtzblatt, Nielsen refer more to the evaluation and re-design process)

usability engineer (the person who facilitate the process. A person with both software engineering and HCI competences)

usability evaluation (the evaluation process using any methods available to usability engineer and usability evaluators) to these we shall add the software engineering process

USABILITY - EVALUATION

Iterative design, with its repeating cycle of design and testing, is the only validated methodology in existence that will consistently produce successful results. If you don't have user-testing as an integral part of your design process you are going to throw buckets of money down the drain". Bruce Tognazzini, http://www.asktog.com. (14 years at Apple Computer, he founded the Apple Human Interface Group).

Tog points out five good reasons for investing in user testing, see also

1. Problems are fixed before the product is shipped, not after.
2. The team can concentrate on real problems, not imaginary ones.
3. Engineers code instead of debating
4. Time to market is sharply reduced
5. Finally, upon first release, your sales department has a rock-solid design it can sell without having to pepper their pitches with how it will all actually work in release 1.2 or 2.0

EVALUATION PARADIGMS

Table 11.1 Characteristics of different evaluation paradigms

Evaluation paradigms	"Quick and dirty"	Usability testing	Field studies	Predictive
Role of users	Natural behavior.	To carry out set tasks.	Natural behavior.	Users generally not involved.
Who controls	Evaluators take minimum control.	Evaluators strongly in control.	Evaluators try to develop relationships with users.	Expert evaluators.
Location	Natural environment or laboratory.	Laboratory.	Natural environment.	Laboratory-oriented but often happens on customer's premises.
When used	Any time you want to get feedback about a design quickly. Techniques from other evaluation paradigms can be used–e.g., experts review software.	With a prototype or product.	Most often used early in design to check that users' needs are being met or to assess problems or design opportunities.	Expert reviews (often done by consultants) with a prototype, but can occur at any time. Models are used to assess specific aspects of a potential design.
Type of data	Usually qualitative, informal descriptions.	Quantitative. Sometimes statistically validated. Users' opinions collected by questionnaire or interview.	Qualitative descriptions often accompanied with sketches, scenarios, quotes, other artifacts.	List of problems from expert reviews. Quantitative figures from model, e.g., how long it takes to perform a task using two designs.
Fed back into design by . . .	Sketches, quotes, descriptive report.	Report of performance measures, errors etc. Findings provide a benchmark for future versions.	Descriptions that include quotes, sketches, anecdotes, and sometimes time logs.	Reviewers provide a list of problems, often with suggested solutions. Times calculated from models are given to designers.
Philosophy	User-centered, highly practical approach.	Applied approach based on experimentation, i.e., usability engineering.	May be objective observation or ethnographic.	Practical heuristics and practitioner expertise underpin expert reviews. Theory underpins models.

Characteristics of different evaluation paradigms. Page 344 (Preece et.al., 2002)

and page 594 (Preece et.al., 2007) Table 12.1 Characteristics of different evaluation approaches (without "Quick and dirty")

USABILITY TESTING

From the usability lab at institute 8 Aalborg University, where a craftsman is evaluating a handheld mobile system for time registration and resource management. The left most screen shows the camera mounted to the mobile phone and the right most actions on the server that the mobile phone communicates with. (The 'IT at the Building Site' project, http://it.bt.aau.dk/it/projects/index.html). See also (Christiansson P, Svidt K, 2006).

EVALUATION TECHNIQUES

Relation between evaluation paradigm and techniques. Page 347 (Preece et.al., 2002)

and page 596 (Preece et.al., 2007) Table 12.1 The relationship between evaluation approaches and methods (without "Quick and dirty")

Table 11.2 The relationship between evaluation paradigms and techniques.

Techniques	Evaluation paradigms: "Quick and dirty"	Usability testing	Field studies	Predictive
Observing users	Important for seeing how users behave in their natural environments.	Video and interaction logging, which can be analyzed to identify errors, investigate routes through the software, or calculate performance time.	Observation is the central part of any field study. In ethnographic studies evaluators immerse themselves in the environment. In other types of studies the evaluator looks on objectively.	N/A
Asking users	Discussions with users and potential users individually, in groups or focus groups.	User satisfaction questionnaires are administered to collect users' opinions. Interviews may also be used to get more details.	The evaluator may interview or discuss what she sees with participants. Ethnographic interviews are used in ethnographic studies.	N/A
Asking experts	To provide critiques (called "crit reports") of the usability of a prototype.	N/A	N/A	Experts use heuristics early in design to predict the efficacy of an interface.
User testing	N/A	Testing typical users on typical tasks in a controlled laboratory-like setting is the cornerstone of usability testing.	N/A	N/A
Modeling users' task performance	N/A	N/A	N/A	Models are used to predict the efficacy of an interface or compare performance times between versions.

DESIGN HEURSTICS

8 Golden rules in HCI design

* *Strive for consistency*
  sequens of actions in similar situations, identical terminologi in menus, prompts etc.,
* *Enable frequent users to use shortcuts*
* *Offer informative feedback*
  for every user actions there should be system feedback.
* *Design dialogs to yield closure*
  sequences of actions should be organized into groups with a beginning, middle, and end.
* *Offer error prevention and simple error handling*
  to prevent serious errors (e.g prefer menu selection to form filling), minimize retyping forms,..
* *Permit easy reversal of actions*
* *Support internal locus control*
  operators should be in charge (no surprises, difficulty to obtain information,..)
* *Reduce short-term memory load*
  human short-term memory handles 7 +-2 chunks.

Shneiderman, 1998, "Designing the User Interface". Addison-Wesley Longman Inc., Reading Massachusetts. (638 pp). See also http://www.aw.com/DTUI/. Nielsen J., Mack R., 1994, "Usability Inspection Methods". John Wiley & Sons, Inc. New York. (413 pp.)

More to read:
Xerox, 1995, 'X Heuristic Evalualtion - A System Checklist'. Usability Analysis & Design, Xerox Corporation. (22 pp.)
[education/repofrts/he_cklst.pdf]

1. *Visibility of system status*
   Are users kept informed about what is going on? Is appropriate feedback provided within reasonable time about a user's action?
2. *Match between system and the real world*
   Is the language used at the interface simple? Are the words, phrases and concepts used familiar to the user?
3. *User control and freedom*
   Are there ways of allowing users to easily escape from places they unexpectedly find themselves in?
4. *Consistency and standards*
   Are the ways of performing similar actions consistent?
5. *Help users recognize, diagnose, and recover from errors*
   Are error messages helpful? Do they use plain language to describe the nature of the problem and suggest a way of solving it?
6. *Error prevention*
   Is it easy to make errors? If so where and why?
7. *Recognition rather than recall*
   Are objects, actions and options always visible?
8. *Flexibility and efficiency to use*
   Have accelerators (i.e shortcuts) been provided that allow more experienced users to carry out tasks more quickly?
9. *Aesthetic and minimalist design*
   Is any unnecessary and irrelevant information provided?
10. *Help and documentation*
    Is help information provided that can be easily searched and easily followed?

- Nielsen J., Mack R., 1994, "Usability Inspection Methods". John Wiley & Sons, Inc. New York. (413 pp.)
- Jacob Nielsen, Heuristic evaluation. http://www.useit.com/papers/heuristic/

DECIDE framework

DECIDE: framework to guide evaluation (Preece, 2007. page 626)
(Preece, 2002. page 348)

- Determine the overall *goals* (that the evaluation addresses)
- Explore the specific *questions* to be answered
- Choose the evaluation paradigm (*approach*) and techniques (*methods*) to answer the questions
- Identify the *practical issues* that must be addressed, such as selecting participants
- Decide how to deal with the *ethical issues*
- Evaluate, interpret, and present the *data*

END

http://it.civil.aau.dk