4 Analyzing the Requirements

Requirements Definition Process (Cooper)

From A. Cooper, About Face 2.0

- Defining the requirements
  - Step 1: Creating problem and vision statements
  - Step 2: Brainstorming
  - Step 3: Identifying ***persona*** expectations
  - Step 4: Constructing context ***scenarios***
  - Step 5: Identifying needs
    - » Data needs
    - » Functional needs
    - » Contextual needs
- Scenarios
  - Are extremely helpful to understand the real needs of users
  - are an excellent starting point for design activities

Scenario Development

- Important methods
  - **General scenario**
    - » Fictional story featuring the product to be developed and explaining implications on users experience
    - » Similar to describing conceptual models, may be concept video
  - **“Day in the life” scenario**
    - » Creating a fictional user
    - » Describing a day in her life augmented with the product to be developed
  - **Situation scenarios**
    - » Fictional story concentrating on a specific situation, e.g. an emergency case)
- Forms of presentation
  - Writing
  - Video
  - Acting/playing it – connected to paper prototypes

“Day in the Life” Scenario

- Describe the usage of a product in the context of a day
  - In particular for products that are used more than once a day, e.g. mobile services, helps to identify practicalities
- Based on the information gathered invent a day
  - Working day or holiday
  - Make a plan what the persons is going to do on this day
  - Make it a normal day but include real life tension and trade-off (e.g. getting kids to school and having a meeting shortly after that)
  - Don’t let the day to be perfect (e.g. you may forget a document at home)
  - Don’t make the day a nightmare (e.g. do not anticipate the user’s airplane is going to crash)
- Describe a day of the fictional user in detail
  - Concentrate on the relation between the users actions and tasks and the product introduced.
  - Basically asking: “How does the product change the life?”

Example: Day in a Life Scenario

www.IYOUIT.eu

A day with IYOUIT…

Read this story to follow Jason and his family to see how IYOUIT simplifies their life.

For a while now Jason is working for this Web2.0 start-up with offices in Paris and Munich. He lives in Paris together with his wife Kelly and his son Bob. For project and business appointments he goes to Munich on a regular basis.

Today is one of those days.

1. After an early flight he arrives at Munich airport, which he visited many times in the past. As he frequently comes here, the airport area had been automatically detected as a frequent place of stay by IYOUIT.

2. In IYOUIT, Places can be assigned a name and meaning. Jason is referring to this place as an airport named "MUC". This piece of information automatically becomes part of his personal context as he goes through MUC together with other IYOUIT context items that can be acquired (Locations, Places, Experiences, Photos, Sounds, Observations, Products, Weather and Tags).

3. Through the IYOUIT Buddy list personal context items are shared with others as you go. This way Kelly, with a particular buddy view on family members, can observe Jason's arrival in Munich on her mobile phone. It's good to know that he safely arrived!

Example: Day in a Life Scenario

www.IYOUIT.eu

A day with IYOUIT…

4. Jason takes a photo of the nice and modern Munich airport in the rising sun while getting on a taxi. He edits as title "Arrival" and sets his IYOUIT mood to "Relaxed".

5. IYOUIT automatically uploads this picture to Jason's Flickr account and adds all available context information. From the picture and the annotated context it's easy to see that he not only safely arrived, he is also in time, everything is fine and the weather is all right.

6. In fact, Jason is early today and the taxi ride towards his office is ever-so smooth. Jason therefore sends a message to his local project team to tell that he is coming a little earlier. He can swiftly do so via the group SMS function of the IYOUIT buddy list, which automatically groups members according to their current context.

7. His team leader at the Munich office receives this SMS and also sees from the IYOUIT buddy list that all required meetings participants are already present. He re-schedules the meeting to start 30min earlier and again uses the buddy group messaging function to notify all parties.

8. The area of the Munich office is again known to IYOUIT as a frequently visited place of Jason. Upon arrival, a couple of personal settings are automatically switched based on rules that were previously defined by Jason and bound to the office context: the profile of his phone switches to "Silent", the connectivity is handed over to the office WLAN and his presence is defined to be "Busy".

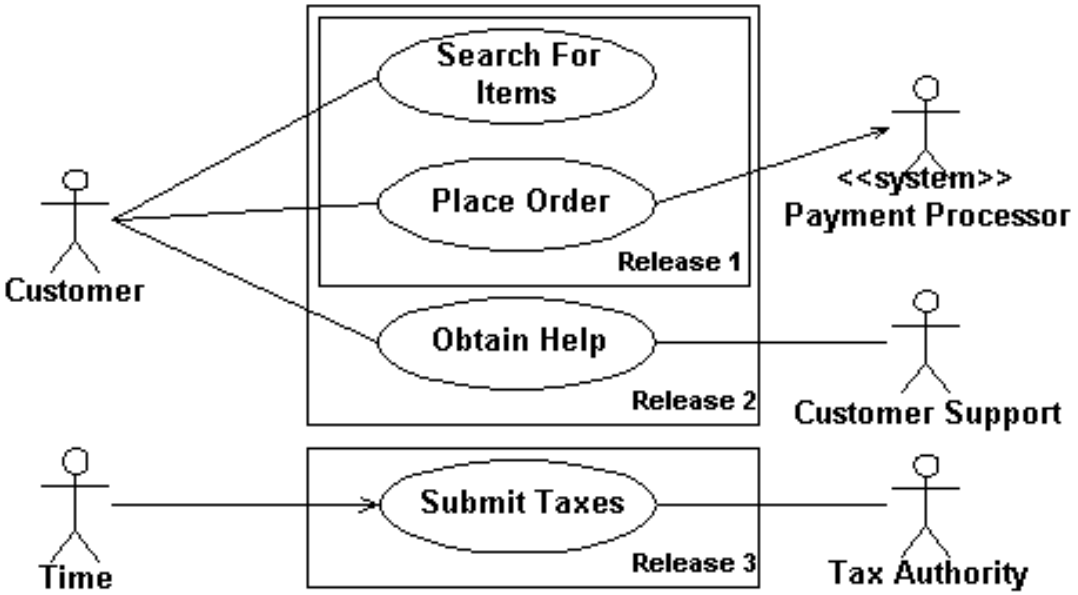
Situation Scenarios

- Concentrating on a very specific situation
- Investigate the requirements and the impact in a specific situation
- May be rather short
- Situation were the product and potentially a particular function is situated into a context
  - e.g. scanning a document in a work context (interrupting work, going to the scanner, operating the device, getting the data, ..)
- Unlikely situations that are of major importance
  - E.g. emergency procedures such as a fire or building evacuation (not applicable to a word processor but relevant for a power plant control room)
- Methods
  - Writing a fictional story
  - Playing/acting the scene with anticipated functionality

Example: Situation Scenario Video

Scenarios, Use Cases, UML

- Unified Modeling Language (UML)
  - Standard graphical modeling language for software systems
  - Includes requirements modeling
- UML “Use Case”:
  - A specific way of using the system by performing some part of its functionality
  - Usually depicted graphically showing the involved stakeholders



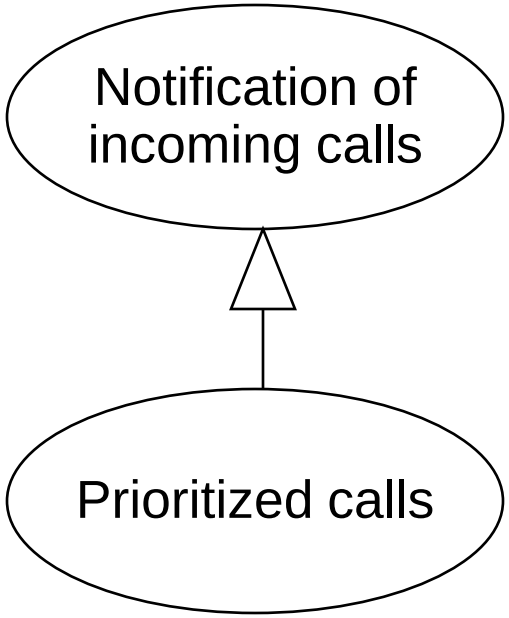
www.agilemodeling.com

- “Scenario”:
  - In UML-based environments: Example instantiation for a use case, giving a detailed sequence of events belonging to the described kind of interaction
  - In HCI: General term for a story about how the system is used, may be used to derive use cases afterwards

Scenarios and Use Cases: Integrated View (1)

Meredith is in a panel discussion at the conference. The kindergarten of her daughter wants to inform her that Sheila has suddenly developed high fever.

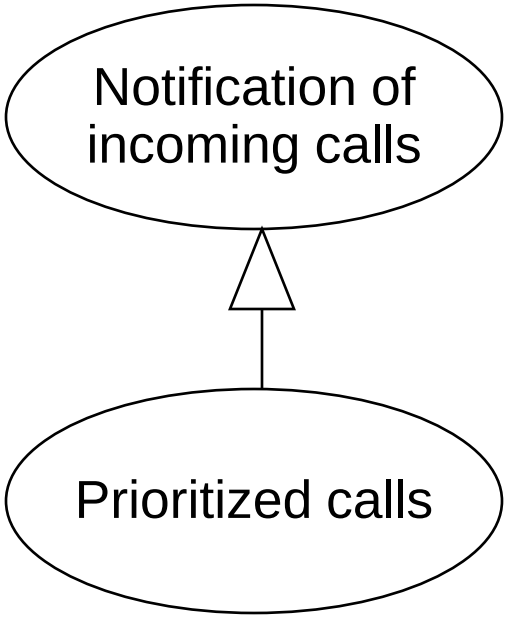
High-level scenario

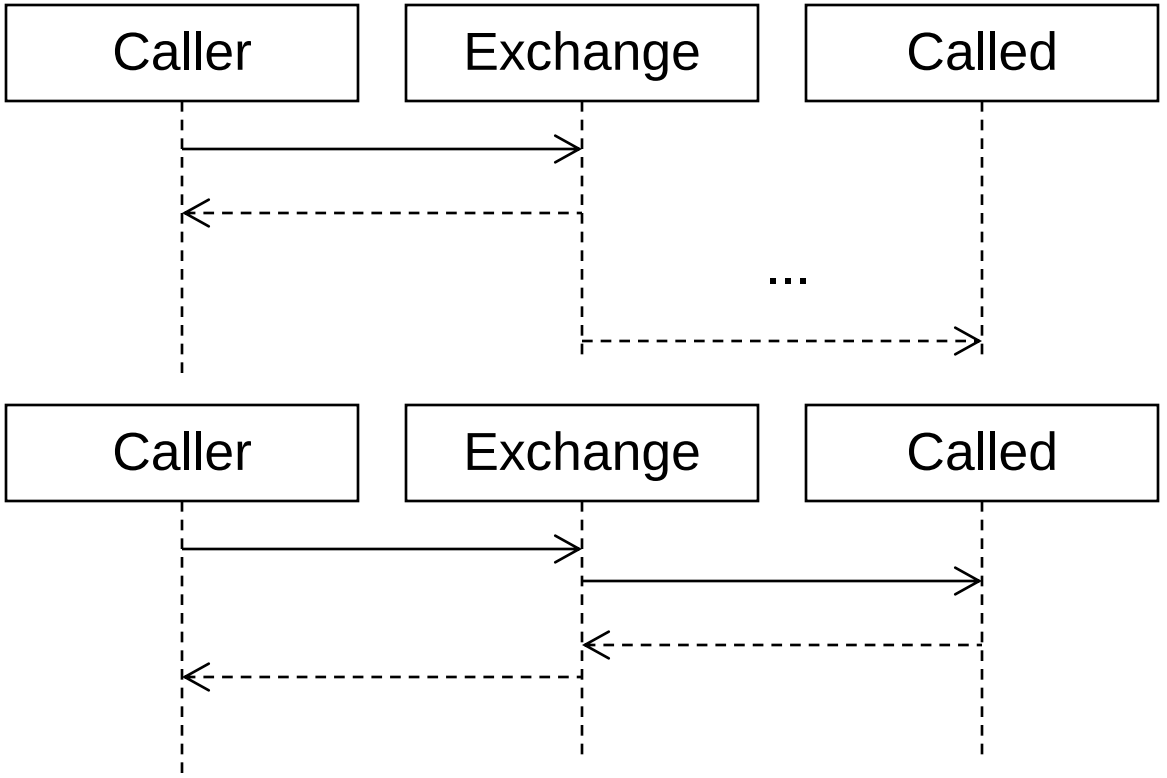


Use-case diagram

Scenarios and Use Cases: Integrated View (2)

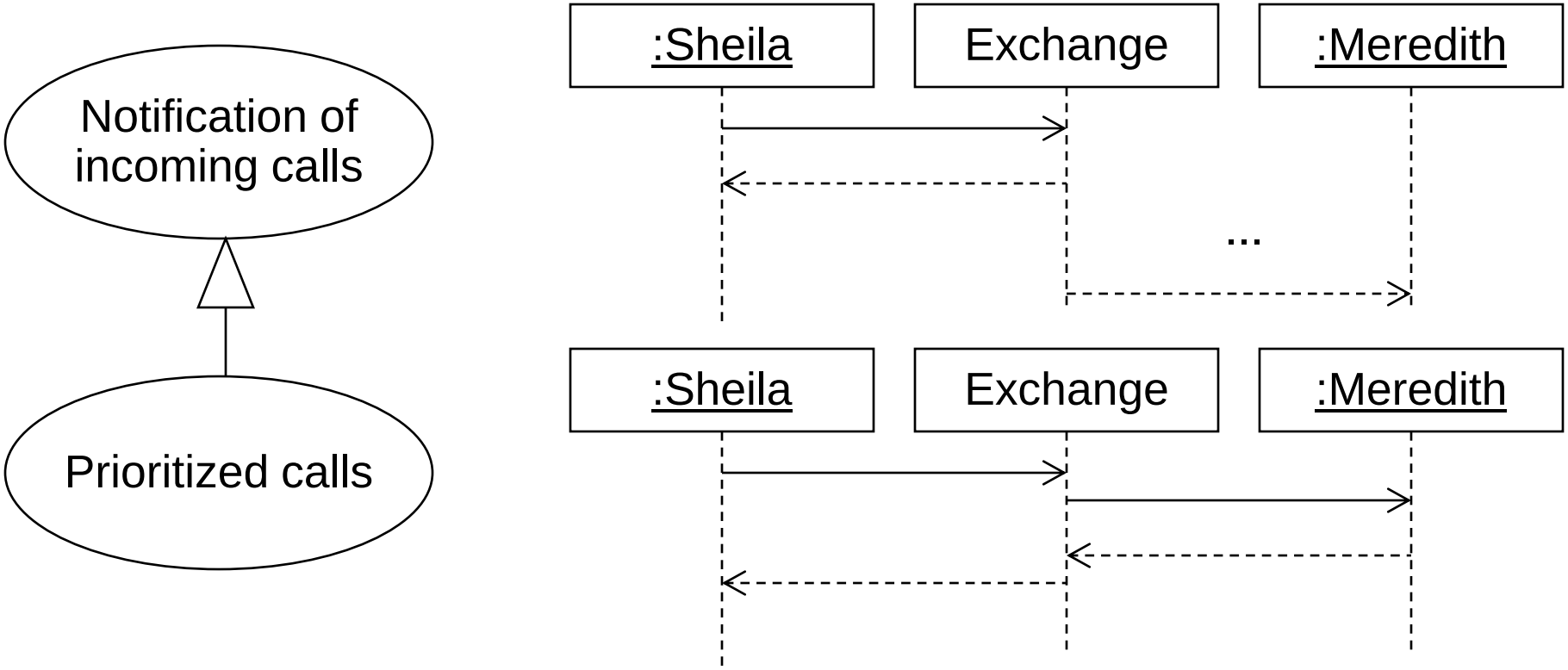
In Software Engineering (specifically using UML), use cases are described e.g. by Sequence Diagrams





Scenarios and Use Cases: Integrated View (3)

Using concrete examples, the difference between SE/UML-scenarios and high-level scenarios often disappears.



Who to Design for? – Personas

- **Don’t design for the average user!!!**
- Differentiate and create a set of typical users
  - “Persona” = concrete representative of one kind of typical users
- Use background information about the user group
  - Literature
  - Interviews
  - Statistics
  - Analysis and observations
- *Invent* a set of specific persons
  - Age, place of birth, current location where she lives
  - Education, profession, job profile, background, hobbies
  - Social environment, family, work relationships
  - Goals and abilities
- Personas are representative for the target audience, but they are NOT average!
- Personas often do not fully correspond to market segments!

Persona Examples (1)

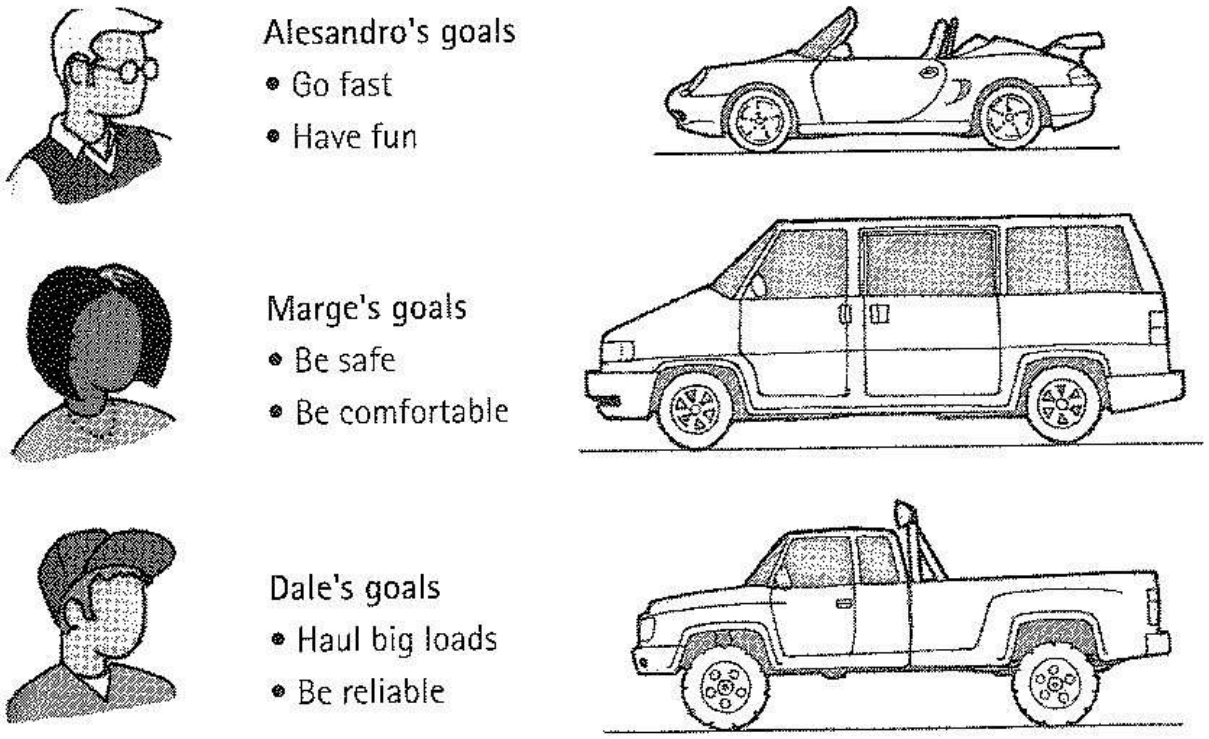


Figure 5-2: A simplified example of how personas are useful. By designing different cars for different people with different specific goals, we are able to create designs that other people with similar needs to our target drivers also find satisfying. The same holds true for the design of digital products and software.

A. Cooper

Persona Examples (2)

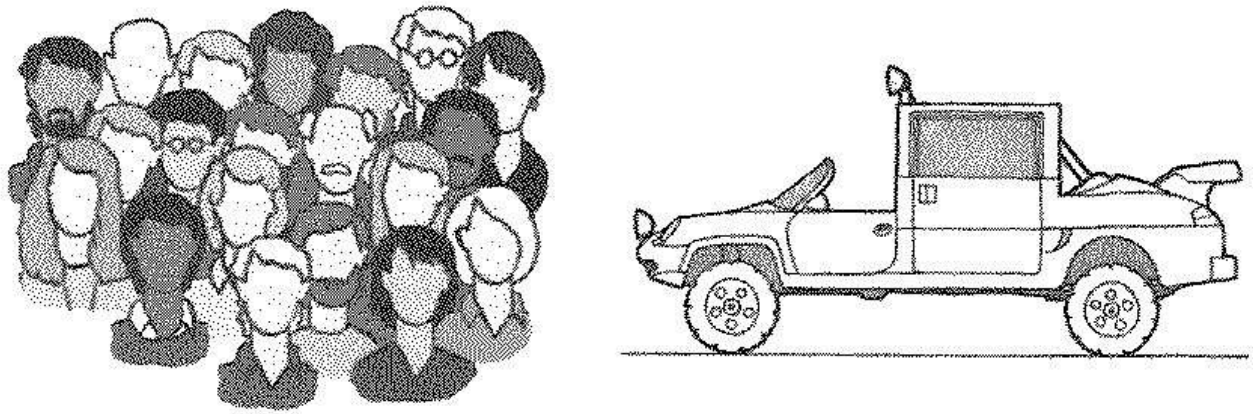


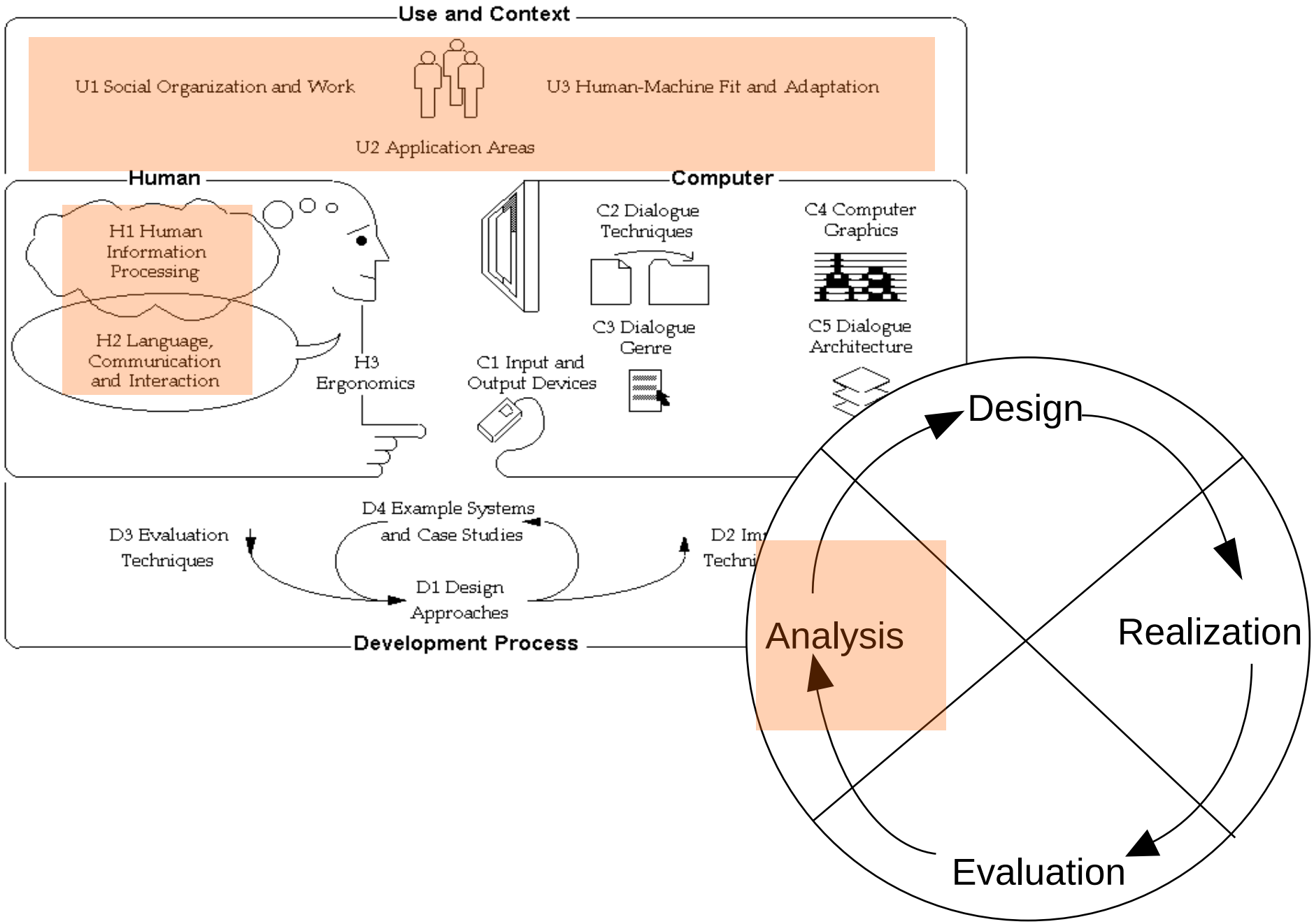
Figure 5-1: A simplified example of how personas are useful. If you try to design an automobile that pleases every possible driver, you end up with a car with every possible feature, but which pleases nobody. Software today is too often designed to please too many users, resulting in low user satisfaction. Figure 5-2 provides an alternative approach.

A. Cooper

Scenario Development
Why Persona?

- Avoiding the “elastic user”
  - If you do not specify the user you can change their abilities to support a design decision made = “elastic user”
- Avoiding self-referential design
  - The designer or developer often assumes (implicitly) that users have his goals and his skills and abilities.
- Avoiding design edge cases
  - Focusing on the design issues which are on the edge of the anticipated audience can consume a lot of effort. By use of typical users the focus on edge cases can be reduced.
- Generally, make requirements concrete
  - Seemingly unnecessary detail helps in making the requirements accessible and understandable for a large audience (users, managers, developers)

4 Analyzing the Requirements


Use and Context
U1 Social Organization and Work
U3 Human-Machine Fit and Adaptation
U2 Application Areas
Human
Computer
H1 Human Information Processing
H2 Language, Communication and Interaction
H3 Ergonomics
C1 Input and Output Devices
C2 Dialogue Techniques
C3 Dialogue Genre
C4 Computer Graphics
C5 Dialogue Architecture
D3 Evaluation Techniques
D4 Example Systems and Case Studies
D1 Design Approaches
Development Process
Design
Realization
Evaluation
Analysis

Motivation
Conceptual Models

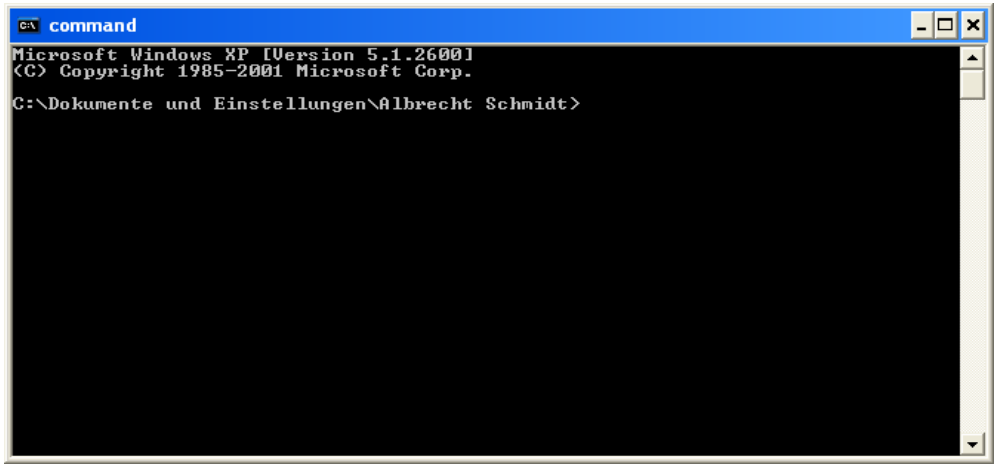
- How do you figure out that those objects are not usable?
- How do you do it for software?

Images from: D. Norman, The Design of everyday things.

Background: The Psychology of Everyday Things

- Norman 2002
- Not primarily aimed at computer science problems but:
  With technologies (web, interactive media, embedded computers) moving into everyday life of most people it becomes highly relevant!
- Terms: Perceived and Real Affordances
  - Affordances determine the range of possible - usually physical - actions by a user on an system/object.
  - Perceived Affordances are the actions perceived by a user that appear to be possible.
  - Example: certain materials afford/support certain forms of vandalism (e.g. glass is smashed, wood is carved, graffiti appears on stone)
- This is also applicable to digital materials and designs.

Low affordance



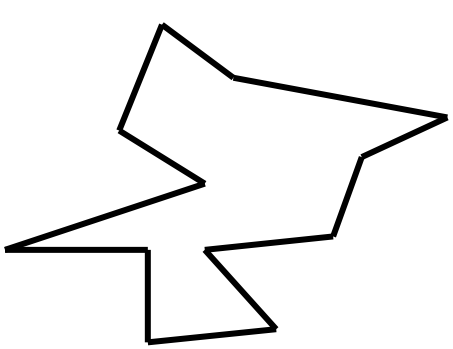
Wrong affordance

High affordance

Example: Heating Control

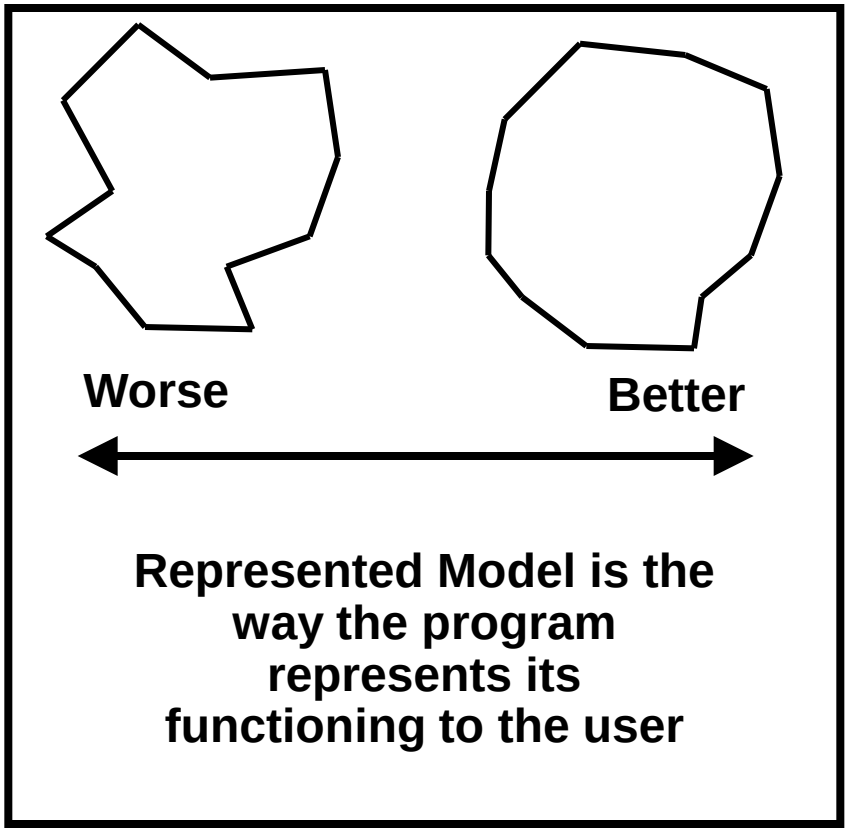
- You come home and it is very cold. Heating is off.
- Your heating system is thermostat controlled.
- To which setting do you turn the thermostat?
  - 1, 2, 3, 4, 5, 6

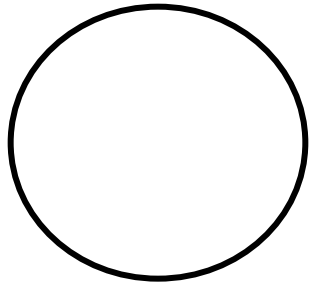
Implementation, Represented, Conceptual Model



Implementation Model

reflects technology





Conceptual Model

reflects user's understanding

From A. Cooper, About Face 2.0

Example: ‘Geldkarte’ (1)

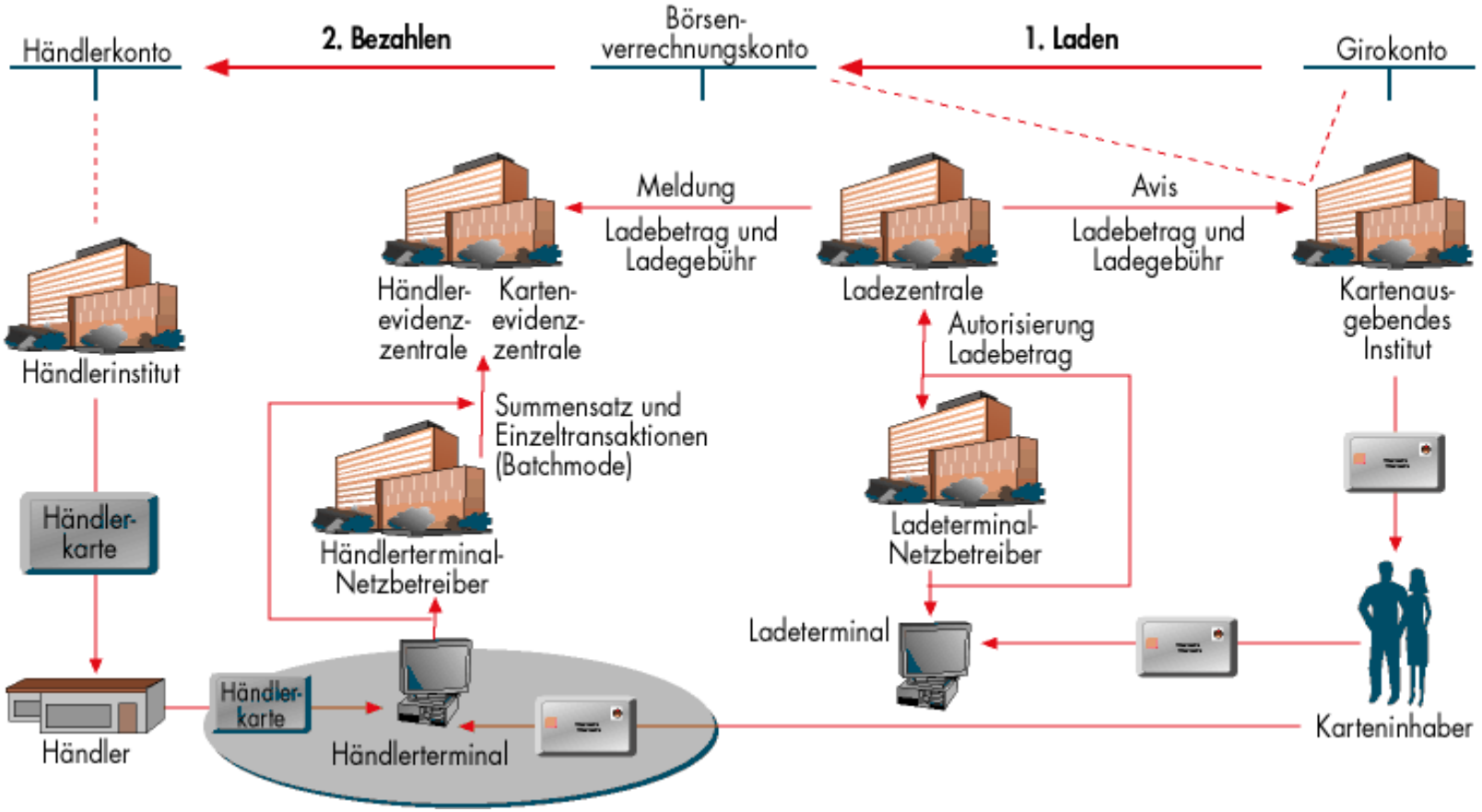
- Store cash on the card

Conceptual Model – by the user

- Pay with the card

Example: ‘Geldkarte’ (2)

Some aspects of the implementation model



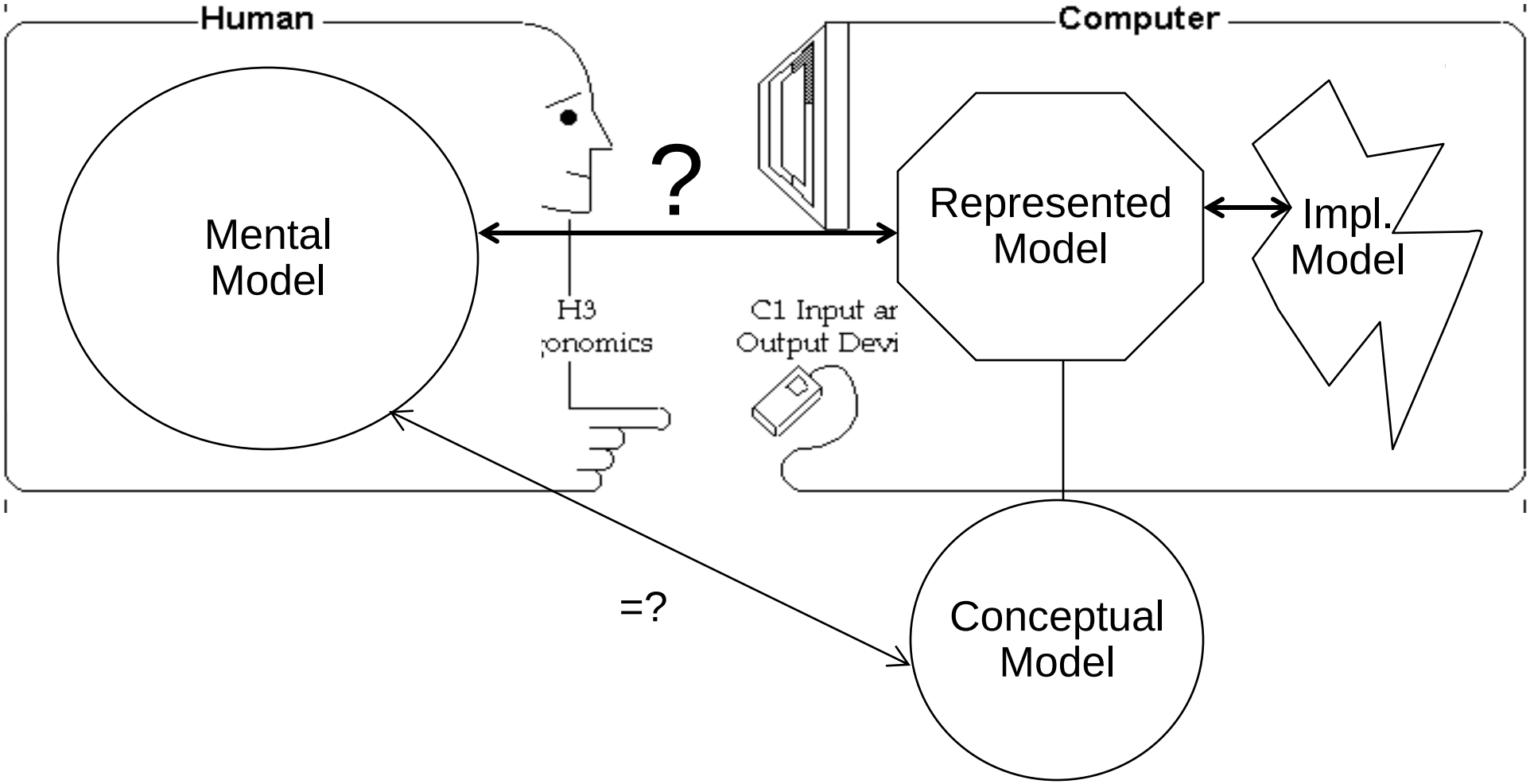
From IX-Article: Chipgeld by Hans-Bernhard Beykirch, http://www.heise.de/ix/artikel/1998/12/148/

Models – Human and Computer

- Applications work on an Implementation Model
- They were designed after a Conceptual Model
- Users operate on their Mental Model
- The user interface translates between models

- Provocative Statement from A. Cooper
  “Computer literacy is nothing more than a euphemism for making the user stretch to understand an alien logic rather than having software-enabled products stretch to meet the user’s way of thinking”

Mental Model and Implementation Model



Conceptual Model

- A conceptual model is “the proposed system in terms of a set of integrated ideas and concepts about what it should do, behave and look like, that will be understandable by the users in the manner intended”
  (Preece, Rogers & Sharp, 2002, Interaction Design, Wiley, p 40)

- “The most important thing to design is the user’s conceptual model. Everything else should be subordinated to making that model clear, obvious and substantial. That is almost exactly the opposite of how most software is designed.”
  (David Liddle, 1996, Design of the conceptual model. In T. Winograd, (editor), Bringing Design to Software. Reading, MA: Addison-Wesley, p17)

Why is This a Big Issue with Digital Products?

- For simple mechanical systems/processes, the conceptual model and implementation model are very similar, e.g.
  - Hammer
  - Power drill

- For digital systems the implementation model is often very complex
  - Many components, often distributed
  - The service provided is a result of contributions from different parts
  - The digital components are not visible – even when you open the device

- Users still have a simple conceptual models to operate digital products
  - Based on what they see and their experience gained in use
  - By the control options they are given
  - By the behaviour and reactions they observe
  - By what they have learned about the system

References

- Alan Dix, Janet Finlay, Gregory Abowd and Russell Beale: Human Computer Interaction (3rd edition), Prentice Hall 2003
- Jennifer Preece, Yvonne Rogers, Helen Sharp: Interaction Design, Wiley 2002
- M. D. LeCompte, J. Preissle: Ethnography and qualitative design in educational research (2nd ed.), Academic Press 1993
- Gary Gaffney: Contextual enquiry - a primer, www.sitepoint.com
- William Hudson. HCI and the web: A tale of two tutorials: a cognitive approach to interactive system design and interaction design meets agility, ACM *interactions* 12(1):2005, 49-51
- A. Khella: Objects-Actions Interface Model http://www.cs.umd.edu/class/fall2002/cmsc838s/tichi/oai.html
- Alan Cooper, Robert M. Reimann. (2007) About Face 3.0: The Essentials of Interaction Design; ISBN 978-0470084113
- D. Norman. The Psychology of Everyday Things, B&T 1989
- K. Bittner, I. Spence: Use Case Modeling, Addison-Wesley 2002