Overview: UM

1. UM: What is it?
3. Early history
4. Demands & traditional (academic) developments
5. What can we adapt to?
6. Generic User Modelling techniques
7. Newer developments
8. The future?

What is a user model?

- Elaine Rich:
  
  "Most systems that interact with human users contain, even if only implicitly, some sort of model of the creatures they will be interacting with."

- Robert Kass:
  
  "...systems that tailor their behaviour to individual users' needs often have an explicit representation structure that contains information about their users, this structure is generally called a user model."
What is a user model, here:

- "If a program can change its behaviour based on something related to the user, then the program does (implicit or explicit) user modelling."

Why user modelling?

- pertinent information
  - What is pertinent to me may not be pertinent to you
  - information should flow within and between users
  - users should control the level of information-push
- large amounts of information
  - too much information, too little time
  - people often become aware of information when it is not immediately relevant to their needs
  - Difficult to handle
  - Etc.

What for?

- In tutoring systems:
  - To adapt to the student’s needs, so that better learning occurs
  - To adapt to the teacher’s needs, so better teaching takes place
- In commercial systems:
  - To adapt to the customer, so that better(?) selling takes place
- Etc.
  - TO ADAPT TO THE USER

How?

- Simplest version: Include facts about the user
- Adapt to known facts about the user
- Adapt to inferred properties of the user
  - Has Eurocard ---> likes travelling
  - Stereotypical user modelling
Adaptivity example

- *User:* Could the student's mispronunciation errors be due to dialect?
- *Response to parent:* Yes, non-standard pronunciations may be due to dialect rather than poor decoding skills.
- *Response to psychologist:* Yes, the student's background indicates the possibility of a dialectical difference.

⇒ **Stereotypes**

User modelling is always about guessing …

Early history

- **Start:** 1978, 79:
  - Allen, Cohen & Perrault: *Speech research for dialogue coherence*
  - Elaine Rich: *Building & Exploiting User Models (PhD thesis)*
- **10 year period of developments**
  - UM performed by application system
  - No clear distinction between UM components & other system tasks
- **mid 80’s:** Kobsa, Allgayer, etc.
  - Distinction appears
  - No reusability consideration

Early systems

- **GUMS** (Finin & Drager, 1989; Kass 1991)
  - *General User Modelling System*
  - Stereotype hierarchies
  - Stereotype members + rules about them
  - Consistency verification
  ⇒ set framework for General UM systems
- **Called** **UM shell systems** (Kobsa)
**Academic developments**

- **Doppelgaenger [Orwant 1995]**
  - Hardware & software sensors
  - Offers techniques of data generalization (linear prediction, Markov models, unsupervised clustering for stereotype formation)
- **TAGUS [Paiva & Self 1995]**
  - Stereotype hierarchy, inference mech., TMS, diagnostic system + misconception library

**Other UM shells**

  - Represents assumptions on knowledge, beliefs, preferences (attribute – value pairs) – actually, library
- **BGP-MS [Kobsa & Pohl 1995, Pohl 1998]**
  - Belief, Goal and Plan Maintenance System
  - Assumptions about users + user groups
  - Allows multi-user, can work as network server
- **LMS [Machado et al. 1999]**
  - Learner Modelling Server

**UM shell services (Kobsa 95)**

- Representation of assumptions on user characteristic(s)
  - E.g., knowledge, misconceptions, goals, plans, preferences, tasks, abilities
- Representation of common characteristics of users
  - Stereotypes, (sub-)groups, etc.
- Recording of user behaviour
  - Past interaction w. system
  - formation of assumption based on interaction
- Generalization of interaction (histories)
  - Stereotypes
- Inference engine
  - Drawing new assumptions based on initial ones
  - Current assumption + justification
  - Evaluation of entries in current UM and comparison w. standards
- Consistency maintenance
UM shells Requirements

- **Generality**
  - As many services as possible
  - “Concessions”: student-adaptive tutoring systems

- **Expressiveness**
  - Able to express as many types of assumptions as possible (about $U$)

- **Strong Inferential Capabilities**
  - AI, formal logic (predicate l., modal reasoning, reasoning w. uncertainty, conflict resolution)

An Example of UM system

David Benyon, 1993

Gerhard Fischer 1 HFA Lecture, OZCHI’2000

Semantic 'levels' of user modeling

*Surface Behaviour*

Immediate tasks  Dialogue Context

Preferences

Tasks and goals  Expertise

Motivational forces

(Crises, Social Context)

Information Delivery, Contextualization and Intrusiveness

*SHAL I GIVE RUM A HINT?*

Gerhard Fischer 1 HFA Lecture, OZCHI’2000
‘Deep’ or ‘shallow’ modelling?

- Deep models give more inferential power!
- Same knowledge can affect several parts of the functionality, or even several applications
- Better knowledge about how long an inference stays valid
- But deep models are more difficult to acquire
  - Where do all the inference rules come from?
  - How do we get information about the user?

Aim of UM

- Obtaining of user metal picture
  - Vs.
  - User behaviour modelled *per se*

A Comparison between Adaptive and Adaptable Systems

<table>
<thead>
<tr>
<th>Definition</th>
<th>Adaptive</th>
<th>Adaptable</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dynamic adaptation by the system itself to current task and current user</td>
<td>User changes with substantial system support to the functionality of the system</td>
<td></td>
</tr>
<tr>
<td>Knowledge</td>
<td>Contained in the system; projected in different ways</td>
<td>Knowledge is extended</td>
</tr>
<tr>
<td>Stengths</td>
<td>Little or no effort by the user; no special knowledge of the user is required</td>
<td>User is in control; system knowledge will fit better; success model exists</td>
</tr>
<tr>
<td>Weaknesses</td>
<td>User has difficulty developing a coherent model of the system; loss of control; few (if any) success models exist (except humans)</td>
<td>Systems become incompatible; user must do substantial work; complexity is increased (user needs to learn the adaptation component)</td>
</tr>
<tr>
<td>Mechanisms Required</td>
<td>Models of users, tasks and dialogues; knowledge base of goals and plans; powerful matching capabilities; incremental update of models</td>
<td>Layered architecture; human problem-domain communication; “back-talk” from the system; design rationale</td>
</tr>
<tr>
<td>Application Domains</td>
<td>Active help systems; critiquing systems; differential descriptions; user interface customization</td>
<td>End-user modifiability; tailorable; filtering; design-in-use</td>
</tr>
</tbody>
</table>

Gerhard Fischer | HFA Lecture, OZCHI’2000

What can we adapt to?

- User knowledge
- Cognitive properties
  - (learning style, personality, etc.)
- User goals and plans
- User mood and emotions
- User preferences
Adaptation to User Knowledge

- **$U$** option knowledge
  - about possible actions via an interface
- Conceptual knowledge
  - that can be explained by the system
- Problem solving knowledge
  - how knowledge can be applied to solve particular problems
- Misconceptions
  - erroneous knowledge

How can we infer user knowledge?

- It’s in general hard to infer something about the user’s knowledge. Techniques used:
  - Query the user (common in tutoring systems)
  - Infer from user history (if you’ve seen an explanation, you understand the term)
  - Rule-based generalisation based on domain structure (if you understand a specific term, you understand its generalisation)
  - Rule-based generalisation based on user role (if you’re a technician, you should understand these terms)
  - ‘Bug’ libraries (recognise common errors)
  - Generalization based on other similar users’ past
The “Model overlay” technique

- Advantage: Simple and ‘cheap’
- Cannot model misconceptions, or ‘new’ knowledge

What can we adapt to?

- User knowledge
  - Cognitive properties (learning style, personality, etc.)
  - User goals and plans
  - User mood and emotions
  - User preferences

Why model cognitive properties?

- Navigation in hypermedia:
  - Very large differences (20:1) in performance, partially related to spatial ability. New tools needed!
- Learning:
  - Different people require different learning styles (example / theory / group based)

Example of cognitive styles

<table>
<thead>
<tr>
<th>Cognitive style</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Field independence</td>
<td>Field-dependent persons develop a fixation to a certain solution that is valid in one situation, and encounter difficulties in a new one that is analogous but has closer similarities.</td>
</tr>
<tr>
<td>Reactivity / reflectivity</td>
<td>Reactive people are quick to make errors, often by following logic that contradicts the transfer from the higher level to the lower, or by constraining understanding higher level models from lower level experiences.</td>
</tr>
<tr>
<td>Operation learning / comprehension / learning</td>
<td>Operation learning is about learning systems, while comprehension learning concerns concept and task level learning. Tools are needed in order to become an expert.</td>
</tr>
</tbody>
</table>

Van der Meer et al.
Kolb (1984) 2-D learning styles scale; 4 extreme cases:

- **1. Converger** (*abstract, active*):
  - abstract conceptualization and active experimentation; great advantage in traditional IQ tests, decision making, problem solving, practical applications of theories; knowledge organizing: hypothetical-deductive; question: "How?".
- **2. Diverger** (*concrete, reflective*):
  - concrete experience and reflective observation; great advantage in imaginative abilities, awareness of meanings and values, generating alternative hypotheses and ideas; question: "Why!"
- **3. Assimilator** (*abstract, reflective*):
  - abstract conceptualization and reflective observation; great advantage in inductive reasoning, creating theoretical models; focus more on logical soundness and preciseness of ideas; question: "What?".
- **4. Accomodator** (*concrete, active*):
  - concrete experience and active experimentation; focus on risk taking, opportunity seeking, action; solve problems in trial-and-error manner; question: "What if?".

Inferring user cognitive characteristics

- The user does not know - not possible to ask!
- Stable properties - use lots of small signs over time.
- Studies required to establish correlation between indications and properties.
- A better solution may be to use these aspects of user models only at design time (offer different interaction alternatives)?

What can we adapt to?

- ✔ User knowledge
- ✔ Cognitive properties
  * (learning style, personality, etc.)
- User goals and plans
- User mood and emotions
- User preferences

User Goals and Plans

What is meant by this?
  - A **user goal** is a situation that a user wants to achieve.
  - A **plan** is a sequence of actions or event that the user expects will lead to the goal.

System can:
  - Infer the user’s goal and suggest a plan
  - Evaluate the user’s plan and suggest a better one
  - Infer the user’s goal and automatically fulfil it (partially)
  - Select information or options to user goal(s) (shortcut menus)
What information is available?

• **Intended Plan Recognition:**
  – Limit the problem to recognizing plans that the user intends the system to recognize
  – User does something that is characteristic for the plan

• **Keyhole Plan Recognition:**
  – Search for plans that the user is not aware of that the system searches for.

• **Obstructed Plan Recognition:**
  – Search for plans while user is aware and obstructing

Keyhole Plan Recognition

• Kautz & Allen 1990:
  – Generalized plan recognition
  – Hierarchical plan structures
  – Method for inferring ’top-level’ actions from lower level observations.

Axioms

• **Abstraction:**
  – Cook-spaghetti $\Rightarrow$ Cook-pasta

• **Decomposition:**
  – Make-pasta-dish $\Rightarrow$ Preconditions, Effects, internal constraints, Make Noodles, Make Sauce, Boil

Intended Plan Recognition

• Used in Natural Language Interpretation.
  – “I want to take the eight o’clock train to London. How do I get to platform four?”

  • Speaker intends to do that by taking the eight o’clock train.
  • Speaker believes that there is an eight o’clock train to London.
  • Speaker wants to get to London.
  • Speaker believes that going to platform four will help in taking the eight o’clock train.
Are these models useful?

- The keyhole case suffers from:
  - Very little actual information from users
  - Users that change their plans and goals
- The intended case suffers from:
  - Need of complex models of intentionality
  - Multiple levels of plans
    - plans for interaction, domain plans, plans for forming plans
  - Differences in knowledge between user and system

“Local” plan recognition

- Make no difference between system and user plans (the keyhole case is limited to recognising plans that belong to the plan library anyway).
- Only domain (or one-level) plans.
- Be forgetful -- inferences based on latest actions.
- Let the user inspect and correct plans.
- Works best with probabilistic or heuristic methods.

What can we adapt to?

- User knowledge
- Cognitive properties
  - (learning style, personality, etc.)
- User goals and plans
- User mood and emotions
- User preferences

Moods and emotions?

- New, relatively unexplored area!
- Unconscious level difficult to recognise, but it is possible to look at type speed, error rates / facial expressions, sweat, heartbeat rate...
- Conscious level can be guessed from task fulfilment (e.g. failures)
- Emotions affect the user’s cognitive capabilities
  - it can be important to affect the user’s emotions (e.g. reduce stress)
Conscious and unconscious emotions

Conscious

"Show hatred"

"Paul is happy"

Analysis / Decisions

Mood

Pattern Recognition / Synthesis

Emotional expressions

Unconscious

We address how emotions arise from an evaluation of the relationship between environmental events & an agent’s plans and goals, as well as the impact of emotions on behaviour, in particular the impact on the physical expressions of emotional state through suitable choice of gestures & body language.

Sample model of emotion assessment

The layers in student modeling

knowledge & cognitive model layer

learning profile layer

believability and emotional layer

Conati, AAAI, North Falmouth, Massachusetts 2001

Abou-Jaoude & Frasson, AI-ED99, Le Mans, France, 1999
What can we adapt to?

- User knowledge
- Cognitive properties
  - (learning style, personality, etc.)
- User goals and plans
- User mood and emotions
  - User preferences

Adaptation to user preferences

- So far, the most successful type of adaptation. Preferences can in turn be related to knowledge / goals / cognitive traits, but one needs not care about that.
- Examples:
  - Firefly
  - www.amazon.com
  - Mail filters
  - Grundy (Rich: personalized book recommendation expert system)

Inferring preferences

- Explicitly stated preferences
  - (CNN News)
- Matching the user’s behaviour towards the user group
  - (Amazon)
- Matching the user’s behaviour towards rule base, and modify the rule base based on groups of users
  - (Grundy)

Elaine Rich: GRUNDY (~1979)

Syst: Please type in your name
User: John Greene
Syst: You haven’t used this system before, have you?
User: No

<Two stereotypes are now possible: "man" and "any-person">
Combining values from several stereotypes

- high value + high value
  - <high value + high certainty>
- high value + low value
  - <weighted mean + low certainty>
- low value + low value
  - <low value + high certainty>

Adaptation model in Grundy

- The characteristic properties are those that have high or low value and high confidence.
  - Choose a book that fits these.
  - Describe those properties of the books that fit the user’s interests.

Can the stereotypes be learned?

- Positive feedback -->
  - Increase certainty on key and property in all triggered stereotypes.
- Negative feedback -->
  - Decrease certainty on key and property in all triggered stereotypes.
- No method to learn totally new stereotypes

Preference models in general

- Advantages:
  - Simple models
  - Users can inspect and modify the model
  - Methods exist to learn stereotypes from groups of users (clustering)
- Disadvantages:
  - The Grundy model for stereotypes does not work in practice => machine learning!
What can we adapt to?

- User knowledge
- Cognitive properties
  - (learning style, personality, etc.)
- User goals and plans
- User mood and emotions
- User preferences

Generic User Modelling Techniques

- Rule-based frameworks
- Frame-based frameworks
- Network-based frameworks
- Probability-based frameworks
- A decision theoretic framework
- Sub-symbolic techniques
- Example-based frameworks

Rule-based frameworks

- Declarative Representation:
  - BGP-MS(Kobsa): A User Modelling Shell
- A Hybrid Representation: SB-ONE
  - Pure Logic Based
- Rule-based adaptations
- Quantification (levels of expertise)
- Stereotypes (U classified)
- Overlay (actual use compared to ideal)

Knowledge representation

- The system knowledge is partitioned into different parts,
  - System beliefs
  - User beliefs
  - Joint beliefs
  - and more…
  - User goals
- Stereotypes: can be activated if certain information is present.
- User Model Partitions
Pros and Cons

- Very general and ‘empty’ - difficult to use
- Truth Maintenance required (expensive)
- There are weights and thresholds, but not much theory behind those
- Learning from feedback not included

Frame-based frameworks

- E.g., semantic network
- Knowledge stored in structures w. slots to be filled
- Useful for small domain

Network-based framework

- Knowledge represented in relationships between facts
- Can be used to link frames
Statistical models, pros and cons

- A theory exist for the calculations
- Usually requires training before usage (no learning from feedback)
- Weak representation of ‘true’ knowledge
- Example: The MS Office assistant (the Lumière project)

UM in Bayesian Networks

- Normally, relates observations to explanations
- Plan Inference, Error Diagnosis
- In Lumière, models the whole chain from observations to adaptation
- The BN approach allows for a combination of declarative knowledge about structure with empirical knowledge about probabilities

Lumière: Network Bayesian Nodes

- Observations
- Explanations
  - as parameters in the user model
- Selection of adaptation
  - help message
- Selection of adaptation strategy
  - active / passive help

Lumière & Office helper
Problems of BN in UM

- Dealing with previous ‘wrong guesses’
- Dealing with changes over time
- Providing end-user inspection and control
Advantages and Disadvantages

- Explicit model of adaptation rules
- Not possible to learn new rules
- Rules could be taken from HCI literature
- BUT - there exist no such rules for adaptive behaviour!
- Possible to tune the adaptations based on feedback
- What should be tuned? User modelling or adaptation modelling?

Example-based framework

- Knowledge represented implicitly within decision structure
- Trained to classify rather than programmed w. rules
- Requires little knowledge acquisition

Some Challenging Research Problems for User Modeling

- identify user goals from low-level interactions
- active help systems, data detectors
- “every wrong answer is the right answer to some other question”
- integrate different modeling techniques
- domain-orientation
- explicit and implicit
- give a user specific problems to solve
- capture the larger (often unarticulated) context and what users are doing (especially beyond the direct interaction with the computer system)
- embedded communication
- ubiquitous computing
- reduce information overload by making information relevant
- to the task at hand
- to the assumed background knowledge of the users
- support differential descriptions (relate new information to information and concepts assumed to be known by the user)

Commercial Boom (late ’90s)

- E-commerce:
  - Product offering
  - Sales promotion
  - Product news
  - Banners
targeted to individual U

- Group Lens (Net Perceptions)
  - Collaborative filtering alg.
  - Explicit/implicit rating (navigational data)
  - Transaction history
- LikeMinds (Andromedia)
  - More modular architecture, load distribution
- Personalization Server (ATG)
  - Rules to assign $U$ to $U$ groups (demographic data: gender, age) – stereotype approach
- Frontmind (Manna)
  - Bayesian networks
- Learn Sesame (Open Sesame)
  - Domain model: objects + attributes + events
  - Clustering algorithms

Characteristics of CS

- Client-server architecture for the WEB !!!
- Advantages:
  - Central repository w. $U$ info for 1/more applic.
    - Info sharing between applications
    - Complementary info from client DB integrated easily
  - Info stored non-redundant
    - Consistency & coherence check possible
    - Info on user groups maintained w. low redundancy (stereotypes, a-priori or computed)
  - Security, id, authentication, access control, encryption can be applied for protecting $UM$

UM server Services

- Comparison of $U$ selective actions
  - Amazon: “Customers who bought this book also bought: […]”
- Import of external $U$ info
  - ODBC(Open Database Connectivity) interfaces, or support for a variety of DB
- Privacy support
  - Company privacy policies, industry, law

UM server Requirements

- Quick adaptation
  - Preferably, at first interaction, to attract customers
  - levels of adaptation, depending on data amount
- Extensibility
  - To add own methods, other tools $\Rightarrow$ API for $U$ info exchange
- Load balancing
  - Reaction to increased load: e.g., CORBA based components, distributed on the Web
- Failover strategies (in case of breakdown)
- Transaction Consistency
  - Avoidance of inconsistencies, abnormal termination
Conclusion:

New UM server trends

- More *recommender systems* than real UM
- Based on *network* environments
- Less sophisticated UM, other issues (such as response time, privacy) are more important

- *Separation of tasks* is essential, to give flexibility:
  - Not only system functions separately from UM functions, but also
  - UM functions separation:
    - domain modelling, knowledge, cognitive modelling, goals and plans modelling, moods and *emotion* modelling, preferences modelling, and finally, interface related modelling
  - In this way, the different levels of modelling can be added at different times, and by different people