Reasoning Web Summer School

Personalization for the World Wide Web

Nicola Henze

Prof. Dr. N. Henze

ISI - Semantic Web Group

University of Hannover, Germany

henze@l3s.de

July 2005

Contents

1 Introduction: The User Factor - Personalization and the Semantic Web 3
1.1 User Modeling . . . 5
1.1.1 Kinds of User Models . . . 8
1.2 Personalization in Adaptive Hypermedia . . . 11
1.2.1 Applications . . . 12
1.2.2 Adaptation for Hypertext . . . 14
1.3 Case-Study: Adaptive Educational Hypermedia Systems . . . 17
1.3.1 "Simple": A Simple Adaptive Hypermedia System . . . 19
1.3.2 A simple AEHS - Extension 1 . . . 22
1.3.3 A simple AEHS - Extension 2 . . . 24
1.4 Personalization based on Web Mining . . . 31
1.4.1 Overview of recommendation techniques . . . 31
1.4.2 Content-based recommendations . . . 32
1.4.3 Collaborative recommendations / social information filtering . . . 35
1.4.4 Demographic recommendations . . . 37
1.4.5 Utility-based recommendations . . . 37
1.4.6 Knowledge-based recommendations . . . 38
1.5 Case-Study: Association Rule Mining . . . 39

1 Introduction: The User Factor - Personalization and the Semantic Web

Definition 1 (Definition of the Semantic Web) *Definition of the Semantic Web by T. Berners-Lee, J. Hendler, O. Lassila in a famous article about the "Semantic Web":*

> *"The Semantic Web is an extension of the current web in which information is given well-defined meaning, better enabling computers and people to work in cooperation." – Tim Berners-Lee, James Hendler, Ora Lassila, The Semantic Web, Scientific American, May 17, 2001*

Roots for Personalization in the (Semantic) Web

1. Hypermedia ⟶ so called "Adaptive Hypermedia"
2. Web Mining ⟶ so called "Recommender Systems"

Adaptive Hypermedia

- Origins in Hypermedia Systems and Intelligent Tutoring Systems
- First Systems: 1992/1993
- Milestone:
  P. Brusilovsky: Methods and techniques of adaptive hypermedia.
  User Modeling and User Adapted Interaction, vol. 6, no. 2-3, 1996

Web Mining-based Personalization

- Origins in Data Mining
- Breakthrough: U. Shardanand and P. Maes: Social information filtering: Algorithms for automating "word of mouth"
  "Proceedings of CHI'95 – Human Factors in Computing Systems", 1995.

1.1 User Modeling

- Idea: maintain a system's estimation about the *preferences*, *tasks*, *interests*, *etc.* of a specific user (or groups of users)

Definition 2 (User Profile) *A user profile provides access to certain characteristics of a users. These characteristics are normally stored as attribute-value pairs for each user / user group.*

- A user profile of a user $\mathcal{U}$ gives the instantiations of attributes of $\mathcal{U}$ at a certain timepoint t.

What can be in a User Profile? Characteristics of a user such as ..

- device information (display, connection, tools, ..)
- preferences (colors / html vs. pdf / ...)
- goal
- current task
- information need
- required information depth
- time constraints
- previously regarded information
- previously gained experience
- previously gained knowledge
- ... many, many more

Definition 3 (User Model) *A user model* **maintains** *and* **reasons** *over characteristics of a user stored in user profile.*

- A user model of a user $\mathcal{U}$ ascertains the values of the attributes in the user profile.
- A user model should provide provide
  - updating and modification policies of the user profile
  - instructions to detect and evaluate incidents which can lead to update or modification processes
  - methods for drawing appropriate conclusions about the incidents
  - mechanisms for detecting discrepancies in the modeling process

Crucial points in User Modeling, I

- privacy issues
- characteristics change over time
  - short term characteristics
  - long term characteristics

1.1.1 Kinds of User Models

- Modeling *individual user preferences* / modeling the individual user
  e.g. Knowledge of a user
  Crucial Point: Initialization

- Modeling similar groups of users / modeling *stereotypes*
  Classification of users into stereotypes (pre-knowledge, interests, demographic group, etc.)
  Crucial Point: Balancing (to general $\longrightarrow$ no effect, to specific $\longrightarrow$ no stereotype)

Example 1 (Modeling the individual user for e.g.)

- *e-Learning*
- *task/process support*
- *knowledge management*

Example 2 (Modeling the stereotypes for e.g.)

- *characteristics that can be identified on a group of users*
- *e.g. hospital information system:*
  - *doctors, nurses, patients, visitors, sponsors*
- *groups with a unifiying characteristics, e.g. undergraduate / graduate / PhD students*

Example 3 (A simple User modeling approach) *A very simple user profile identifies all the pages that a user $\mathcal{U}$ has visited, theresfor, it is a set of couples*

$$(\mathcal{P}, visited)$$

A simple user model which can create this-like user profiles contains the following rule for interpreting incidents:

$$\text{“if } \mathcal{U} \text{ visits page } \mathcal{P}, \quad \text{then insert } (\mathcal{P}, \text{visited}) \text{ into the user profile of } \mathcal{U} \text{”}$$

An extension of this simple user model is to recognize the observation that a user $\mathcal{U}$ has bookmarked some page $\mathcal{P}$ and note this in the user profile:

$$\text{“if } \mathcal{U} \text{ bookmarks page } \mathcal{P}, \quad \text{then insert } (\mathcal{P}, \text{important}) \text{ into the user profile of } \mathcal{U} \text{”}$$

Crucial points in User Modeling, II

- If we observe a user $\mathcal{U}$ bookmarking a page $\mathcal{P}$: How can we distinguish that $\mathcal{U}$ has stored this page for future reference based on the content of the page from the fact that $\mathcal{U}$ stored this page only because he liked the design of the page?
- Can we really be sure that bookmarking expresses favour for a page in contrast to denial?
- and many, many more issues...

Appropriate mechanisms for dealing with uncertainty in the observations about the user, and for continuous affirmation of derived conclusions are essential for good user models (a good reference for studying numerical uncertainty management in user modeling is e.g. given in A. Jameson: "Numerical Uncertainty Management in User and Student Modeling: An Overview of Systems and Issues". *User Modeling and User Adapted Interaction*, Vol. 5(3/4), 1996, pp. 193–251.

http://dfki.de/~jameson/pdf/umuai96.jameson.pdf

1.2 Personalization in Adaptive Hypermedia

Definition 4 (Hypertext) *A set of nodes of text which are connected by links. Each node contains some amount of information (text) and a number of links to other nodes.*

Definition 5 (Hypermedia) *Hypermedia: Extension of hypertext which makes use of multiple forms of media, such as text, video, audio, graphics, etc.*

Adaptive hypermedia:

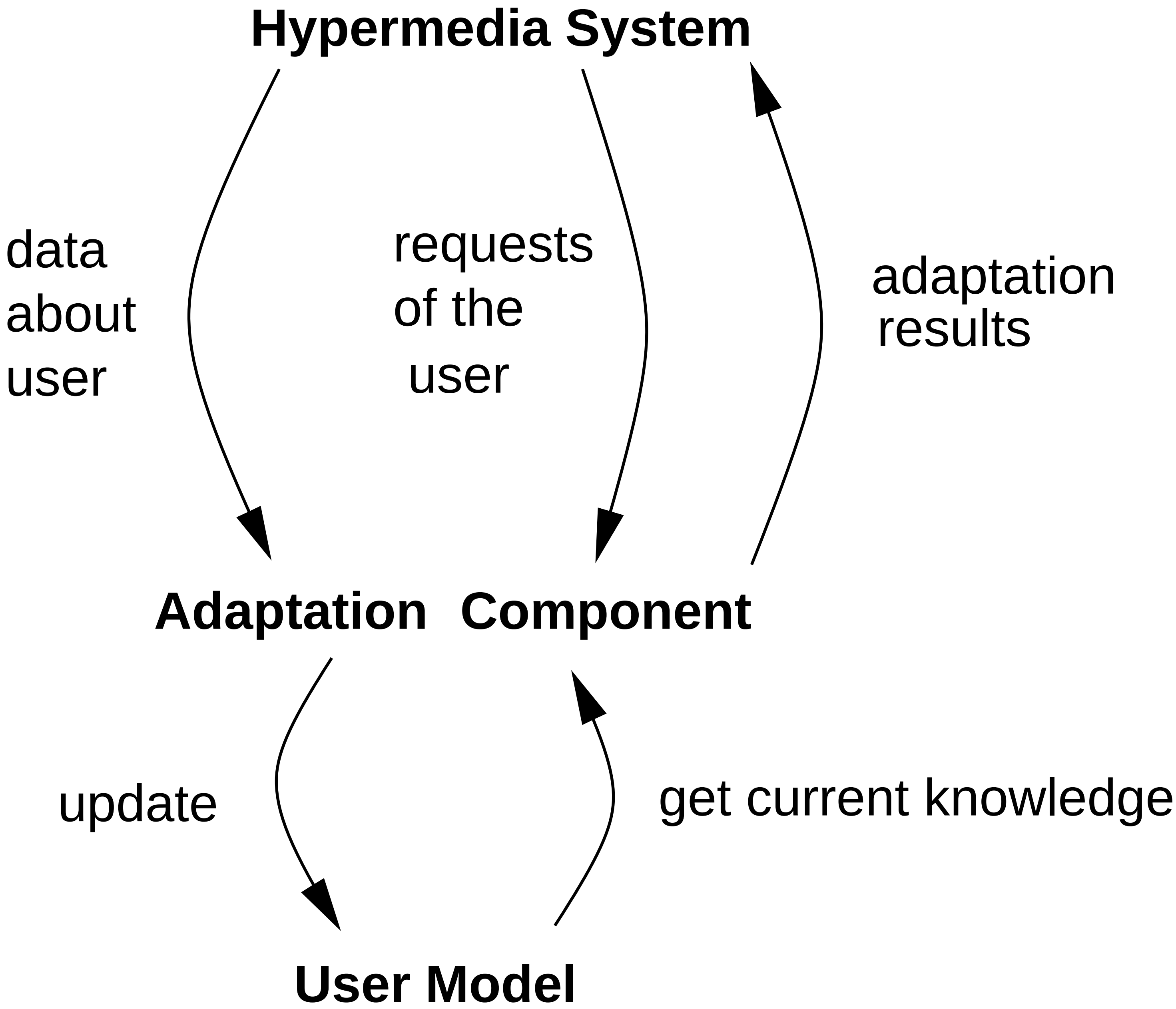
- Personalize hypermedia systems to the individual user.
- Each user has an individual view and individual navigational possibilities for working with the hypermedia system.

Definition 6 (Adaptive hypermedia system) *"By adaptive hypermedia systems we mean all hypertext and hypermedia systems which reflect some features of the user in the user model and apply this model to adapt various visible aspects of the system to the user."*

Definition 7 (Adaptable vs. Adaptive) *An* **adaptable** *system allows a user to adapt the behavior of a system according to his or her needs. An* **adaptive** *system autonoumsly detects a users needs and adapts itself to these, deduced needs.*

1.2.1 Applications

- Educational hypermedia
  Focus: Knowledge / Learning
- Online Information systems
  Electronic encyclopedia, document collections, travel information system, online help systems, organisational information systems, knowledge management, etc.
  Focus: Context (quick reference, preparing a presentation, refreshing knowledge, getting an overview, ...)
- Personalized Views
  Querying a database (static), individual view (dynamic)
- E-Commerce
  Optimized product presentations, adapting to customer behavior, ...
- provide Guidance, Orientation (locally or globally)


Hypermedia System
data about user
requests of the user
adaptation results
Adaptation Component
update
get current knowledge
User Model

1.2.2 Adaptation for Hypertext

- Hypertext $\longrightarrow$ Graph
- Nodes $\longrightarrow$ content level adaptation
- Edges $\longrightarrow$ navigational level adaptation

Content Level Adaptation: What?

- Additional Explanations
- Prerequisited Explanations
- Comparative Explanations
- Explanation Variants
- Presentation order

[And, of course, the mix of methods]

Content Level Adaptation: How?

- Conditional Text
- Stretch text
- Page fragments / Page variants
- Frames

[And, of course, the mix of techniques]

Navigational Level Adaptation: What?

- Direct guidance (next step, sequence of steps, trails, etc.)
- Adaptive Sorting (Similarity, Pre-Knowledge,..)
- Adaptive Hiding
- Annotation of Links (Comments, Explanations, Icons,..)
  - Example: Traffic light metaphor
- Map annotation, annotation of overview graphics

[And, of course, the mix of methods]

Navigational Level Adaptation: How?

- *adding, removing, re-ording* hypertext links at runtime; techniques differ from system to system

1.3 Case-Study: Adaptive Educational Hypermedia Systems

Actors of an Adaptive Educational Hypermedia System:

- is a hypermedia system: graph with nodes (content) and edges (links)
  Actor 1: Document space with relations (DOCS)
- models the characteristics of users in a user model
  Actor 2: User Model (UM)
- users interact with the system during runtime
  Actor 3: Observations (OBS)
- The AEHS adapts the underlying hypermedia system by inserting / deleting / enabling relations for a users, by estimating priorities of relations, by selecting appropriate content, etc.
  Actor 4: Adaption Component: Rules for Adaptation (AC)

Definition 8 (Logic-Based Definition of Adaptive (Educational) Hypermedia System)

- *DOCS: Document Space*
  A finite set of first order logic (FOL) sentences with atoms for describing documents (and knowledge topics) and predicates for defining relations between these atoms.
- *UM: User Model*
  A finite set of FOL sentences with atoms for describing individual users (user groups), and user characteristics, as well as predicates and rules for expressing whether a characteristic applies to a user.
- *OBS: Observations*
  A finite set of FOL sentences with atoms for describing observations and predicates for relating users, documents / topics, and observations.
- *AC: Adaptation Component*
  A finite set of FOL sentences with rules for describing adaptive functionality.

1.3.1 "Simple": A Simple Adaptive Hypermedia System

This adaptive educational hypermedia system can annotate the links of the hypertext according to the actual knowledge state of the user (so called *traffic light metaphor*).

Simple: Document Space A set of n constants (n corresponds to the number of documents in the document space) which represent the documents:

$D_1, D_2, \ldots, D_n$.

A finite set of predicates stating the documents that need to be studied before a document can be learned, e.g. D_j is a prerequisite for D_i:

$\text{preq}(D_i, D_j)$ for certain $D_i \neq D_j$.

Simple: User Model A set of m axioms, one for each individual user:

$U_1, U_2, \ldots, U_m$.

Simple: Observations One constant for the observation whether a document has been visited:

Visited.

And a set of predicates

obs(D_i, U_j, Visited) for certain D_i, U_j.

Simple: Adaptation Component One constant for describing the values of the adaptive functionality "learning_state":

Recommended_for_reading,

and two constants representing values of the adaptive functionality:

Green_Icon, Red_Icon.

Rules for describing the learning state of a document

$\forall U_i \forall D_j$
($\forall D_k$preq(D_j, D_k) $\Longrightarrow$ obs(D_k, U_i, Visited))
$\Longrightarrow$ learning_state(D_j,U_i, Recommended_for_reading).

And rules for describing the adaptive link annotation with traffic lights:

$$\forall \mathrm{U}_i \forall \mathrm{D}_j$$
$$\text{learning\_state}(\mathrm{D}_j,\ \mathrm{U}_i,\ \text{Recommended\_for\_reading})$$
$$\Longrightarrow \text{document\_annotation}(\mathrm{D}_j,\ \mathrm{U}_i,\ \text{Green\_Icon}),$$
$$\forall \mathrm{U}_i \forall \mathrm{D}_j$$
$$\neg\ \text{learning\_state}(\mathrm{D}_j,\ \mathrm{U}_i,\ \text{Recommended\_for\_reading})$$
$$\Longrightarrow \text{document\_annotation}(\mathrm{D}_j,\ \mathrm{U}_i,\ \text{Red\_Icon}).$$

1.3.2 A simple AEHS - Extension 1

We extend our AEHS Simple by an additional rule in the user model UM. The visible adaptive functionality of this system, which we call *Simple 1*, will remain the same as in Simple, however Simple 1 deduces more information from the user observations as Simple.

Simple 1: Document Space Same as the document space in Simple.

Simple 1: User Model As the user model in Simple, plus a rule for inferring that whenever a document has been learned by a user, all the documents that are prerequisites for this document are learned, too. *Simple 1* uses an additional constant for describing user characteristics:

Learned.

A document D is assumed to be learned by a user, if it has been visited,

$$\forall \mathrm{U}_i \forall \mathrm{D}_j$$
$$\mathrm{obs}(\mathrm{D}_j,\ \mathrm{U}_i,\ \mathrm{Visited}) \Longrightarrow \mathrm{p\_obs}(\mathrm{D}_j,\ \mathrm{U}_i,\ \mathrm{Learned}).$$

or if a document D', for which D is a prerequisite, has been visited:

$$\forall \mathrm{U}_i \forall \mathrm{D}_j$$
$$(\ \exists \mathrm{D}_k \mathrm{preq}(\mathrm{D}_k,\ \mathrm{D}_j) \wedge \mathrm{obs}(\mathrm{D}_k,\ \mathrm{U}_i,\ \mathrm{Visited})\ )$$
$$\Longrightarrow \mathrm{p\_obs}(\mathrm{D}_j,\ \mathrm{U}_i,\ \mathrm{Learned}).$$

These inference rules process an observation, they are therefore abbreviated by p_obs for process observation.

Simple 1: Observations Same as Simple.

Simple 1: Adaptation Component The rule describing the learning state of a document is updated as follows:

$$\forall \mathrm{U}_i \forall \mathrm{D}_j$$
$$\forall \mathrm{D}_k(\mathrm{preq}(\mathrm{D}_j,\ \mathrm{D}_k) \Longrightarrow (\ \mathrm{obs}(\mathrm{D}_k,\mathrm{U}_i,\mathrm{Visited}) \vee \mathrm{p\_obs}(\mathrm{D}_k,\mathrm{U}_i,\mathrm{Learned})\ )$$
$$\Longrightarrow \mathrm{learning\_state}(\mathrm{D}_j,\ \mathrm{U}_i,\ \mathrm{Recommended\_for\_reading}).$$

The rules for adaptive link annotation remain unchanged with respect to *Simple*.

1.3.3 A simple AEHS - Extension 2

We can extend this simple AEHS by using a knowledge graph instead of a domain graph. The system, called *Simple 2* is able to give a more differentiated traffic light annotations to hypertext links as Simple or Simple 1. It is able to recommend pages (green icon), shows which links lead to documents that will become understandable (dark orange icon), which might be understandable (yellow icon), or which are not recommended yet (red icon).

Simple 2: Document Space The document space contains all axioms of the document space of Simple, but does not contain any of the predicates. In addition, it contains a set of s constants (s corresponds to the number of topics in the knowledge space) which name the knowledge topics:

$T_1, T_2, \ldots, T_s.$

A finite set of predicates stating the learning dependencies between these topics: Topic $T_k$is required to understand T_j:

depends(T_j, T_k) for certain $T_j \neq T_k$.

The documents are characterized by a set of n predicates which assign a non-empty set of topics to each document. This can be compared by assigning a set of keywords to each document (keep in mind that more than one keyword might be assigned to a document):

$\forall D_i \exists T_j$
keyword(D_i,T_j).

Simple 2: User Model The user model is the same as in Simple, plus an additional rules which defines that a topic T_i is assumed to be learned whenever the corresponding document has been visited by the user. Therefore, *Simple 2* uses like *Simple 1* the constant

Learned.

The rule for processing the observation that a topic has been learned by a user:

$\forall U_i \forall T_j$
($\exists D_k$keyword(D_k, T_j) $\wedge$ obs(D_k,U_i,Visited)
$\Longrightarrow$ p_obs(T_j,U_i,Learned).

Simple 2: Observations Are the same as in Simple.

Simple 2: Adaptation Component The adaptation component of Simple 2 contains two further constants (im comparison to Simple) representing new values for the learning state of a document,

Might_be_understandable, Will_become_understandable.

and two further constants representing new values for adaptive link annotation:

Orange_Icon, Yellow_Icon.

The following rules describe the educational state of a document. Rule_1 states that a document is recommended for learning if *all* prerequisites for the keywords of this document are learned

$$\forall \mathsf{U}_i \forall \mathsf{D}_j$$
$$\forall \mathsf{T}_k \Big(\mathsf{keyword}(\mathsf{D}_j, \mathsf{T}_k) \Longrightarrow (\forall \mathsf{T}_\ell \mathsf{depends}(\mathsf{T}_k, \mathsf{T}_\ell) \Longrightarrow \mathsf{p\_obs}(\mathsf{T}_\ell, \mathsf{U}_i, \mathsf{Learned}))\Big)$$
$$\Longrightarrow \mathsf{learning\_state}(\mathsf{D}_j, \mathsf{U}_i, \mathsf{Recommended\_for\_reading}).$$

Rule_2 states that a document might be understandable if at least some of the prerequisites have already been learned by this user:

$$\forall \mathsf{U}_i \forall \mathsf{D}_j$$
$$(\ \forall \mathsf{T}_k \mathsf{keyword}(\mathsf{D}_j, \mathsf{T}_k) \Longrightarrow$$
$$(\ \exists \mathsf{T}_\ell \mathsf{depends}(\mathsf{T}_k, \mathsf{T}_\ell) \Longrightarrow \mathsf{p\_obs}(\mathsf{T}_\ell, \mathsf{U}_i, \mathsf{Learned})\ )\ )$$
$$\wedge \neg\ \mathsf{learning\_state}(\mathsf{D}_j, \mathsf{U}_i, \mathsf{Recommended\_for\_reading})$$
$$\Longrightarrow \mathsf{learning\_state}(\mathsf{D}_j, \mathsf{U}_i, \mathsf{Might\_be\_understandable}).$$

Rule_3 derives that a document will become understandable if the user has some prerequisite knowledge for at least one of the document's keywords:

$$\forall \mathsf{U}_i \forall \mathsf{D}_j$$
$$\exists \mathsf{T}_k \mathsf{keyword}(\mathsf{D}_j, \mathsf{T}_k) \Longrightarrow$$
$$(\exists \mathsf{T}_\ell \mathsf{depends}(\mathsf{T}_k, \mathsf{T}_\ell) \Longrightarrow \mathsf{p\_obs}(\mathsf{T}_\ell, \mathsf{U}_i, \mathsf{Learned})\ )$$
$$\wedge \neg\ \mathsf{learning\_state}(\mathsf{D}_j, \mathsf{U}_i, \mathsf{Might\_be\_understandable})$$
$$\Longrightarrow \mathsf{learning\_state}(\mathsf{D}_j, \mathsf{U}_i, \mathsf{Will\_become\_understandable}).$$

Four rules describe the adaptive link annotation:

$\forall U_i \forall D_j$
learning_state(D_j, U_i, Recommended_for_reading)
$\Longrightarrow$ document_annotation(D_j,U_i,Green_Icon)

$\forall U_i \forall D_j$
learning_state(D_j, U_i, Will_become_understandable)
$\Longrightarrow$ document_annotation(D_j,U_i, Orange_Icon)

$\forall U_i \forall D_j$
learning_state(D_j, U_i, Might_be_understandable)
$\Longrightarrow$ document_annotation(D_j,U_i, Yellow_Icon)

$\forall U_i \forall D_j$
$\neg$ learning_state(D_j, U_i, Recommended_for_reading)
$\Longrightarrow$ document_annotation(D_j,U_i, Red_Icon)

System	DOCS	UM	OBS
Simple	D_1, D_2, ..., D_n.	U_1, U_2, ... U_m.	Visited.
Simple 1	D_1, D_2, ..., D_n.	U_1, U_2, ... U_m, Learned.	Visited.
Simple 2	D_1, D_2, ..., D_n, T_1, T_2, ... T_s.	U_1, U_2, ..., U_m. Learned.	Visited.

System	AC–Learning State	AC–Adaptive Link Annotation
Simple	Recommended_for_reading.	Green_Icon. Red_Icon.
Simple 1	Recommended_for_reading.	Green_Icon. Red_Icon.
Simple 2	Recommended_for_reading. Might_be_understandable. Will_become_understandable.	Green_Icon. Red_Icon. Orange_Icon. Yellow_Icon.

Table 1: Constants used in *Simple*, *Simple 1* and *Simple 2*.

Table 2 shows the different relations between objects. Table 3 gives an overview about rules used in *Simple*, *Simple 1* and *Simple 2*.

System	DOCS	UM	OBS	AC
Simple	preq(D_i, D_j).	–	obs(D_k, U_j, Visited).	–
Simple 1	preq(D_i, D_j).	–	obs(D_k, U_j, Visited).	–
Simple 2	keyword(D_i,T_j) depends(T_j,T_k).	–	obs(D_k, U_j, Visited).	–

Table 2: Predicates used in *Simple*, *Simple 1* and *Simple 2*.

System	DOCS	UM	OBS
Simple	–	–	–
Simple 1	–	p_obs(D_i,U_j,Learned)	–
Simple 2	–	p_obs(D_i,U_j,Learned)	–

System	AC–Learning State	AC–Adaptive Link Annotation
Simple	learning_state(D_i,U_j,X), X is a constant from AC	document_annotation(D_i,D_j,Y), Y is an constant of AC
Simple 1	"	"
Simple 2	"	"

Table 3: Rules used in *Simple*, *Simple 1* and *Simple 2*.

1.4 Personalization based on Web Mining

- use the effects and dynamics of the structure of the World Wide Web in order to detect relations between Web resources
- analysis the *Web graph*:
  - detecting relations between Web resources:
    * *existing relations* - hypertext linkgs
    * *virtual relations* - resources ae related but not neccessarily linked to each other by a hyperlink
- two major targets:
  - Web **content** mining
  - Web **usage** mining
- As systems which use Web Mining-based Personalization normally point a user to Web resources, e.g. to draw attention to interesting resources, related resources, etc., these systems are normally subsumed as **Recommender Systems**
- in contrast to adaptive hypermedia
  - does not work on such well-defined corpora like a hypertext system
  - normally has no external models like domain or expert models, instead creates dynamic models which grow with the number of Web resources integrated into the model

1.4.1 Overview of recommendation techniques

- Content-based Recommendations
- Collaborative Recommendations
- Demographic Recommendations
- Utility-based Recommendations
- knowledge-based Recommendations

1.4.2 Content-based recommendations

- each user is assumed to operate independently
- recommendations can exploit information derived from document contents
- user profile:
  profile construction: initially, users apply candidate profiles against their own preferences
  Example: A candidate user profile for the rating of today's news article is presented, the user can accept / reject the ratings for the articles
  profile maintenance: keywords / content descriptors which contribute to the rating of each article
- **quality measures**: Precision, Recall

Definition 9 (Precision) *Let* R *the set of* relevant *documents (of a test reference collection), and let* A *be the* answer *set.*

Precision is the fraction of the retrieved documents which is relevant:

$$Recall = \frac{|R \cap A|}{|A|}$$

Definition 10 (Recall) *Let* R *the set of* relevant *documents (of a test reference collection), and let* A *be the* answer *set.*

Recall is the fraction of the relevant documents among the documents retrieved.

$$Recall = \frac{|R \cap A|}{|R|}$$

More information:

http://information-retrieval.de/irb/ir.part_1.chapter_3.section_7.subdiv1_3.html

Content-based recommendation

Let U, I be sets fo users or items, and $\mathcal{U}, \mathcal{I}$ denote an individual user item:

typical background: Features of items in I

typical input: $\mathcal{U}$'s ratings of items in I

typical process: Generate a classifier that fits $\mathcal{U}$'s rating behavior and use it on I.

see: Tutorial "AI Techniques for Personalized Recommendation, A. Jameson, J, Konstan, and J. Riedl,

www.dfki.de/~jameson/aaai02-ttrl.html

Limitations of Content-based recommendation

- items must be machine parsable or with assigned attributes
- only recommendations based on what the user has already seen before (and indicated to like)
- no filtering based on quality, style, or point-of-view (only based on **content**

1.4.3 Collaborative recommendations / social information filtering

- process of "word-of-mouth"
- recommendations: items are recommended to a user based upon values assigned by other people with *similar taste*
- **hypothesis:** people's tastes are not randomly distributed: there are general trends and patterns within the taste of a persona as well as between groups of people.
- user profile:
  profile construction: initially, users explicitely rank some sample objects **profile maintenance:** update with rankings made by this user. **predictions:** Input: "Ratings of *similar users*". The similarity is measured on base of the user profile values

Collaborative recommendations

Let U, I be sets fo users or items, and $\mathcal{U}, \mathcal{I}$ denote an individual user item:

typical background: Ratings from U of items in I

typical input: $\mathcal{U}$'s ratings of items in I.

typical process: Identify users in U similar to $\mathcal{U}$, and extrapolate from their ratings of I

see: Tutorial "AI Techniques for Personalized Recommendation, A. Jameson, J, Konstan, and J. Riedl,
www.dfki.de/~jameson/aaai02-ttrl.html

Limitations of Collaborative recommendation

- required: critical mass of users before the system can make recommendations
- how to get the first rating of a new object?

1.4.4 Demographic recommendations

Let U, I be sets fo users or items, and $\mathcal{U}, \mathcal{I}$ denote an individual user item:

typical background: Demographic information about U and their ratings of items in I

typical input: Demographic information about $\mathcal{U}$

typical process: Identify users in U that are demographically similar to $\mathcal{U}$, and extrapolate from their ratings of I

1.4.5 Utility-based recommendations

Let U, I be sets fo users or items, and $\mathcal{U}, \mathcal{I}$ denote an individual user item:

typical background: Features of items in I

typical input: A utility function over items in I that describes $\mathcal{U}$'s preferences

typical process: apply the function to the items and determine I's rank

Example: Find a product that meets a users needs.

1.4.6 Knowledge-based recommendations

Let U, I be sets fo users or items, and $\mathcal{U}, \mathcal{I}$ denote an individual user item:

typical background: Features of items in I; knowledge of how these items meet a user's needs

typical input: A description of $\mathcal{U}$'s needs or interests

typical process: Infer a match between I and $\mathcal{U}$'s needs

see: Tutorial "AI Techniques for Personalized Recommendation, A. Jameson, J, Konstan, and J. Riedl,

www.dfki.de/~jameson/aaai02-ttrl.html

1.5 Case-Study: Association Rule Mining

Idea:

- detecting "relations" between data items
- uncovered relationships are not inherent data:
  - not like functional dependencies
  - do not represent any sort of causality or correlation
- association rules detect "common usage" of items

Example 4 (Grocery Store) *A grocery store chain keeps a record of weekly transaction where each transaction represents the items brought during one cash register transaction.*

The executives of the chain receive a summarized report of the transactions indicating what types of items have sold at what quantity. In addition, they periodically request information about what items are commonly purchased together.

Goal: find which items are purchased together, e.g. to mount a marketing campaign.

- *a database in which an association rule is to be found is viewed as a set of tuples:*
  - *each tuple contains a set of items*
    - *items: represent the items purchased*
    - *tuples: the list of items purchased together*

Example Data for Grocery Store:

Transaction	Items
t1	Bread, Jelly, Peanut-Butter
t2	Bread, Peanut-Butter
t3	Bread, Milk, Peanut-Butter
t4	Beer, Bread
t5	Beer, Milk

Definition 11 (Association Rule) *Given a set of items $I = \{I_1, I_2, \ldots, I_m\}$ and a database of transactions $D = \{t_1, t_2, \ldots, t_n\}$ where $t_i = \{I_{i1}, I_{i2}, \ldots I_{ik}\}$ and $I_{ij} \in I$, an* **association rule** *is an implication of the form $X \Longrightarrow Y$, where $X, Y \subset I$ are sets of items classed* itemsets *and $X \cap Y = \emptyset$.*

- we are normally not interested in **all** association rules, only in "important" ones
- measures for "importance" association rules: **support** and **confidence**.

Definition 12 (Support) *The* **support (s)** *for an association rule $X \Longrightarrow Y$ is the percentage of transactions in the database that contain $X \cup Y$.*

$$\textbf{support}(X \Longrightarrow Y) = \frac{|\{t_i \in D : X \cup Y \subset t_i\}|}{|D|}$$

Set	Support	Set	Support
Beer	40	Beer, Bread, Milk	0
Bread	80	Beer, Bread, Peanut-butter	0
Jelly	20	Beer, Jelly, Milk	0
Milk	40	Beer, Jelly, Peanut-butter	0
Peanut-butter	60	Beer, Milk, Peanut-butter	0
Beer, Bread	20	Bread, Jelly, Milk	0
Beer, Jelly	0	Bread, Jelly, Peanut-butter	20
Beer, Milk	20	Bread, Milk, Peanut-butter	20
Beer, Peanut-butter	0	Jelly, Milk, Peanut-butter	0
Bread, Jelly	20	Beer, Bread, Jelly, Peanut-butter	0
Bread, Milk	20	Beer, Bread, Milk, Peanut-butter	0
Bread, Peanut-butter	60	Beer, Bread, Jelly, Milk	0
Jelly, Milk	0	Beer, Jelly, Milk, Peanut-butter	0
Jelly, Peanut-butter	20	Bread, Jelly, Milk, Peanut-butter	0
Milk, Peanut-butter	20	Beer, Bread, Jelly, Milk, Peanut-butter	0
Beer, Bread, Jelly	0		

Definition 13 (Confidence / Strength) *The* **confidence** *or* **strength** **(α)** *for an association rule $X \implies Y$ is the ratio of the number of transactions that contain $X \cup Y$ to the number of transactions that contain X.*

$$\textbf{confidence}(X \implies Y) = \frac{|\{t_i \in D : X \cup Y \subset t_i\}|}{|\{t_i \in D : X \subset t_i\}|} = \frac{\textbf{support}(X \cup Y)}{\textbf{support}(X)}$$

Example: Support and Confidence for Some Association Rules:

$X \implies Y$	s	α
Bread $\implies$ Peanut-Butter	60%	75%
Peanut-Butter $\implies$ Bread	60%	100%
Beer $\implies$ Bread	20%	50%
Peanut-Butter $\implies$ Jelly	20%	33.3%
Jelly $\implies$ Peanut-Butter	20%	100%
Jelly $\implies$ Milk	0%	0%

Discussion

- support: measures how often the rule occurs in the database
  - lower values for support may be allowed as support indicates the percentage of time the rule occurs throughout the database
- confidence: measures the strength of a rule
- example: Jelly $\Longrightarrow$ Peanut-Butter: support: 20%, confidence: 100%
  - this association rule exists only in 20% of the transactions
  - but when antecedent "Jelly" occurs, the consequent always occurs
  - possible reaction on this association rule: an advertising strategy targeted to people who purchase Jelly
- typically: large confidence values and smaller support values are used

Definition 14 (Association Rule Problem) *Given a set of items* $I = \{I_1, I_2, \ldots, I_m\}$ *and a database of transactions* $D = \{t_1, t_2, \ldots, t_n\}$ *where* $t_i = \{I_{i1}, I_{i2}, \ldots I_{ik}\}$ *and* $I_{ij} \in I$, *the* **association rule problem** *is to identify all association rules* $X \Longrightarrow Y$ *which satisfy a minimum support "s" and a minimum confidence "*α*". The values (s,*α*) are given as input to the problem.*

Hard part: finding the high-support (frequent) itemsets

Easy part: checking the confidence of associations rules for given frequent itemsets is relatively easy.