Ontology-Based Semantic Recommendation for Context-Aware E-Learning

Zhiwen Yu[1,2], Yuichi Nakamura[1], Seiie Jang[2], Shoji Kajita[2], and Kenji Mase[2]

[1] Academic Center for Computing and Media Studies, Kyoto University, Japan
yu@ccm.media.kyoto-u.ac.jp, yuichi@media.kyoto-u.ac.jp
[2] Information Technology Center, Nagoya University, Japan
jang@itc.nagoya-u.ac.jp, kajita@nagoya-u.jp, mase@nagoya-u.jp

Abstract. Nowadays, e-learning systems are widely used for education and training in universities and companies because of their electronic course content access and virtual classroom participation. However, with the rapid increase of learning content on the Web, it will be time-consuming for learners to find contents they really want to and need to study. Aiming at enhancing the efficiency and effectiveness of learning, we propose an ontology-based approach for semantic content recommendation towards context-aware e-learning. The recommender takes knowledge about the learner (user context), knowledge about content, and knowledge about the domain being learned into consideration. Ontology is utilized to model and represent such kinds of knowledge. The recommendation consists of four steps: semantic relevance calculation, recommendation refining, learning path generation, and recommendation augmentation. As a result, a personalized, complete, and augmented learning program is suggested for the learner.

1 Introduction

E-learning allows learners to access electronic course contents through the network and study them in virtual classrooms. It brings many benefits in comparison with conventional learning paradigm, e.g., learning can be taken at any time, at any place (e.g., campus, home, and train station). However, with the rapid increase of learning content on the Web, it will be time-consuming for learners to find contents they really want to and need to study. The challenge in an information-rich world is not only to make information available to people at any time, at any place, and in any form, but to offer the right thing to the right person in the right way [1][2]. Therefore, e-learning systems should not only provide flexible content delivery, but support adaptive content recommendation.

For better learning experience and effect, the recommendation of learning content should take into account the contextual information of learners, e.g., prior knowledge, goal, learning style, available learning time, location and interests. This new learning paradigm is called context-aware e-learning [3]. In this paper, we propose an ontology-based approach for semantic recommendation to realize context-awareness in learning content provisioning. We aim to make recommendation by exploiting

J. Indulska et al. (Eds.): UIC 2007, LNCS 4611, pp. 898–907, 2007.

knowledge about the learner (user context), knowledge about the content, and knowledge about the learning domain. The recommendation approach is characterized with semantic relevance calculation, recommendation refining, learning path generation, and recommendation augmentation. The knowledge modeling and the whole recommendation process are performed based on ontology. In the current system, we mainly consider two kinds of the most important contexts in learning, i.e., the learner's prior knowledge and his learning goal.

The paper is structured as follows. Section 2 discusses previous work relevant to this paper. In Section 3, we present the ontology model to express knowledge about the learner, content, and the domain being learned. Section 4 describes the ontology-based semantic recommendation in detail. The prototype implementation and preliminary results are described in Section 5. Finally, Section 6 concludes the paper and points out directions for future work.

2 Related Work

There has been much work done in the area of recommendation over the past decade. The interest in developing various recommender systems still remains high because of the abundance of practical applications that help users to deal with information overload and provide personalized service [4]. The objects manipulated by recommender systems include a broad spectrum of artefacts, such as documents, books, CDs, movies, and television programs. Compared with these fields, learning content recommendation is a new topic with the emergence of e-learning. It has only been investigated in several systems in the past few years.

The EU project, LIP [3] aims to provide immediate learning on demand for knowledge intensive organizations through incorporating context into the design of e-learning systems. A matching procedure is presented to suggest personalized learning programs based on user's current competency gap.

COLDEX [5] considers the learner's preferences and hardware/software characteristics in serving learning materials. Collaborative filtering technique is utilized for content recommendation.

The authors of [6] present learning content recommendation based on ontology, which utilizes sequencing rules to connect learning objects. The rules are formed from the knowledge base and competency gap analysis.

The Elena project [7] ranks learning resources according to text filter (a weight is calculated between the specified text and each document), category filter (the distances from the specified classifications in the ontology to the entries specified in the subject field from each resource are evaluated), and the combination of the weight and the distance in the ontology.

Bomsdorf [8] introduces a concept of "plasticity of digital learning spaces" for the adaptation of learning spaces to different contexts of use. A rule-based ascertainment engine is used to identify learning resources according to learner's situation.

Paraskakis [9] proposes a paradigm of ambient learning aiming at providing access to high quality e-learning material at a time, place, pace and context that best suits the individual learner.

K-InCA [10] is an agent-based system supporting personalized, active and socially aware e-learning. The personal agent is aware of the user's characteristics and cooperates with a set of expert cognitive agents.

Our work differs from previous work in several aspects. First, we provide content recommendation through knowledge-based semantic approach. LIP project [3] retrieves objects by matching rather than through semantic relevance. COLDEX [5] recommends learning materials based on collaborative filtering not knowledge-based technique. Second, besides learning content ranking, we also support recommendation refining, learning path generation, and recommendation augmentation. The learning content recommendation presented in [6] is based on ontology and connects learning objects. However, it did not support recommendation refining and recommendation augmentation. Elena [7] provides content ranking and aggregation, while learning path recommendation and results refining are not supported. Third, as for content recommendation, we mainly consider user's personal learning context, e.g. learning goal and prior knowledge. The rule-based recommendation strategy proposed by Bomsdorf [8] mainly considers device and network context rather than personal context. Although [9] and [10] claim to provide personalized learning material access, the approach of recommendation has not been described.

3 Ontology Model

We use ontologies to model knowledge about the learner (user context), knowledge about the content, and the domain knowledge (the taxonomy of the domain being learned). Within the domain of knowledge representation, the term *ontology* refers to the formal and explicit description of domain concepts, which are often conceived as a set of entities, relations, instances, functions, and axioms [11]. By allowing learners or contents to share a common understanding of knowledge structure, the ontologies enable applications to interpret learner context and content features based on their semantics. Furthermore, ontologies' hierarchical structure lets developers reuse domain ontologies (e.g., of computer science, mathematics, etc.) in describing learning fields and build a practical model without starting from scratch.

In our system, we have designed three ontologies: Learner Ontology, Learning Content Ontology, and Domain Ontology. The Learner Ontology shown in Fig. 1 depicts contexts about a learner, e.g., subject or particular content already mastered, learning goal, available learning time, current location, desired learning style, and learning interests. The learning goal may be an abstract subject or a particular content. *lco* and *do* stand for Learning Content Ontology and Domain Ontology, respectively. Properties of contents as well as relationships between them are defined by the Learning Content Ontology (see Fig. 2). The relation *hasPrerequisite* describes content dependency information, i.e., content needs to be taken before the target content. Actually, nowadays most of the departments in university provide a course dependency chart when issuing their courses. The Domain Ontology is proposed to integrate existing consensus domain ontologies such as computer science, mathematics, chemistry, etc. The domain ontologies are organized as hierarchy to demonstrate topic classification. For instance, the hierarchical ontology of computer science domain is presented in Fig. 3. It derives from the well-known ACM taxonomy (http://www.acm.org/class/1998/).

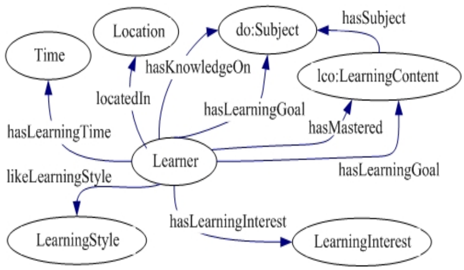


Fig. 1. Learner ontology

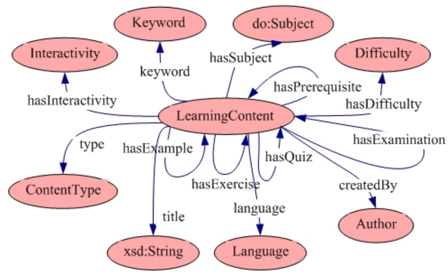


Fig. 2. Learning content ontology

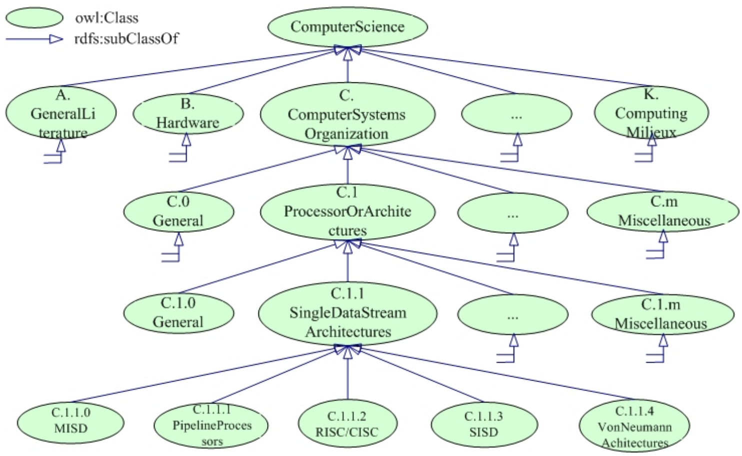


Fig. 3. Computer science domain ontology

We adopt OWL (Web Ontology Language) [12] to express ontology enabling expressive knowledge description and data interoperability of knowledge. It basically includes ontology class definition and ontology instance markups. According to the aforementioned learner ontology, the following OWL based markup segment describes the learning contexts about **`Harry`**.

```
<Learner rdf:about="Harry">
<hasLearningGoal>Distributed Comping</hasLearningGoal>
<hasMastered>CS100</hasMastered>
<hasLearningTime>20:00:00-22:00:00</hasLearningTime>
...
</Learner>
```

4 Semantic Content Recommendation

The learning content recommendation consists of four steps as shown in Fig. 4. First, the Semantic Relevance Calculation computes the semantic similarity between the learner and the learning contents, and then generates a recommendation list accordingly. Second, the Recommendation Refining provides an interactive way to adjust the result until several acceptable options are achieved. When the learner selects one item from the candidates, the Learning Path Generation builds a studying route composed of prerequisite contents and the target learning contents, which guides the learning process. Finally, the Recommendation Augmentation aggregates appendant contents related with the main course. Each step of the recommendation are performed by exploiting knowledge about the learner (goal and prior knowledge), knowledge about the contents (features and relations among them), or the domain knowledge. The Ontology Base provides persistent storage and efficient retrieval of such knowledge.

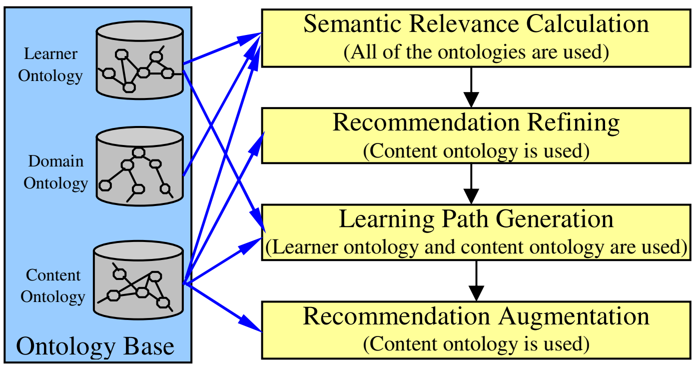


Fig. 4. Content recommendation procedure

4.1 Semantic Relevance Calculation

For recommendation, we first need to rank the learning contents with respect to how much the content satisfies the learner's context. Here we mainly consider the learning goal context. Our system uses the semantic relevance between the learner's goal and learning content as the ranking criteria.

The semantic relevance is inspired by category theory and conceptual graph [13]. It is intuitive that objects in the same domain or related domain may have some similarity within each other. In other words, instances in a category hierarchy have some commonality. Similarity between two objects in the category hierarchy can be measured according to their correlation in the hierarchy model. This is done through analyzing the positions of the objects in the hierarchy model. The closer two objects are, the larger similarity between them will be.

The semantic relevance is calculated through the following steps:

1. *Map the user's goal to the domain ontology*
2. *Locate the subject of the learning content in the domain ontology*
3. *Estimate the conceptual proximity between the mapped element and the subject node of the learning content*

The conceptual proximity ($S(e1, e2)$) is formally defined and determined according to the following rules ('e1' and 'e2' are two elements in the hierarchical domain ontology):

Rule (1): The conceptual proximity is always a positive number, that is,

$$S(e1, e2) > 0 \tag{1}$$

Rule (2): The conceptual proximity has the property of symmetry, that is,

$$S(e1, e2) = S(e2, e1) \tag{2}$$

Rule (3): If 'e1' is the same as 'e2',

$$S(e1, e2) = Dep(e1) / M \tag{3}$$

Rule (4): If 'e1' is the ancestor or descendant node of 'e2',

$$S(e1, e2) = Dep(e) / M \qquad e = \begin{cases} e1 & e1 \textit{ is the ancestor node of } e2 \\ e2 & e1 \textit{ is the descendant node of } e2 \end{cases} \tag{4}$$

Rule (5): If 'e1' is different from 'e2' and there is no ancestor-descendant relationship between them,

$$S(e1,e2) = Dep(LCA(e1, e2)) / M \tag{5}$$

In equations (3), (4) and (5), *M* denotes the total depth of the domain hierarchy ontology; *Dep(e)* is the depth of node 'e' in the hierarchy (the root node always has the least depth, say 1); *LCA(x, y)* means the Least Common Ancestor node for node 'x' and 'y'.

With the above definitions and the domain hierarchy structure given in Fig. 3, we can infer that M=5; LCA(MISD, SISD)= SingleDataStreamArchitecture; Dep(LCA(MISD, SISD))=4; hence S(MISD, SISD)= Dep(LCA(MISD, SISD))=4/5=0.8.

The semantic relevance is based on the intuitive notion that the amount of relevance between the learning goal and content subject increases as they are nearer and more is known about them. For example, two contents of "SingleDataStreamArchitecture" are known to be more similar than two contents of "ProcessorOrArchitecture".

With semantic relevance calculated, we can recommend those contents whose semantic relevance is larger than a preset threshold.

4.2 Recommendation Refining

A recommendation list can be provided to the learner with respect to semantic relevance. However, it may still include overwhelming contents or those contents that are not satisfactory according to the learner's preferences, e.g., difficulty level. Our system offers interactive recommendation refining [14], through which the learner can interact with the system critiquing its recommendation and interactively refining the results until several acceptable options are achieved. The recommendation result can be refined according to the following features: speciality, difficulty, and interactivity.

Speciality. If the result contains very few items and the learner wants to get more generalized contents, the system can give all contents whose subject falls one upper level of LCA (here we define LCA as the least common ancestor of the current recommendation items, which may contains subclass or not) in the hierarchy. Similarly, if the result includes a lot of items and the learner wants to get more specialized contents, the system can return those contents whose subject is one lower level of LCA in the hierarchy. When "More specialized" refining action is triggered, a dialog will pop up to ask the learner to choose one subclass of the LCA.

Difficulty. The learner can refine the result to choose easier or more difficult contents. This can be achieved through the property of *hasDifficulty* of the contents. Each content are assigned a difficulty level when authored, which includes "very easy", "easy", "medium", "difficult", and "very difficult". The difficulty critiquing is made to a particular candidate in the recommendation list. For example, if the learner wants to obtain easier contents with item X as reference, the system will put forward the contents whose difficulty level is lower than that of X while the other features are the same.

Interactivity. Similar to difficulty, the learner can get contents with preferred interactivity by increasing or decreasing the interactivity level of a particular item. The critiquing can be accomplished through the property of *hasInteractivity* of the contents. When created, the content is given an interactivity level according to its presentation method and layout. The interactivity level ranges from "very low", "low", "medium", "high", to "very high".

4.3 Learning Path Generation

Usually a single learning content will not be practicable for the learner to meet his goal, because learning contents themselves may have prerequisites that the user has not mastered yet. Therefore we need to provide the learner with a learning path to guide the learning process and suggest the user to obtain some preliminary knowledge before immersing in the target content.

When the learner selects one item from the recommendation list, the system can generate a learning path connecting with prerequisite contents and the target learning content. This is accomplished by recursively adding prerequisite contents of the learning content into the path until it reaches the basic contents that have no prerequisites, and then pruning it based on the learner's prior knowledge. The prerequisite course information is provided by the *hasPrerequisite* relation of a particular content. The learning path should be a DAG (Directed Acyclic Graph). We therefore detect and eliminate cyclic graph in building the path.

The following algorithm outlines the executions taking place during learning path generation. For each content $\boldsymbol{C}^i$ in the current learning path, first extract the prerequisite list from its XML description file *xml_*$\boldsymbol{C}^i$ (line 10). Then for each content $\boldsymbol{C}^j$ in the prerequisite list, if it does not belong to the user's prior learned course list and the current learning path, add it into the learning path and revise $\boldsymbol{C}^j$'s direct subsequence list, offspring list, and number of steps to the target learning content (line 14-18). If $\boldsymbol{C}^j$ already exists in the learning path, but does not belong to the user's prior learned course list and the offspring list of $\boldsymbol{C}^i$, it is not necessary to add again, but need to update its direct subsequence list, offspring list, and number of steps to the target learning content (line 19-24).

Algorithm. Generating the learning path for the specified learning content ***LC***.

```
Input: LC, user's prior learned course list PC_List
Output: LC's learning path LP
Procedure: Generate_Learning_Path
Begin
LP = NULL
LC_Offspring_List = NULL
LC_JumpsToGoal = 0
LP ← LP+LC
For each C^i ∈ LP
    C^i_Prerequisite_List ← Get_Prerequisite(xml_C^i)
    For each C^j ∈ C^i_Prerequisite_List
      C^j_DirectSubsequence_List = NULL
      C^j_Offspring_List = NULL
      If C^j ∉ PC_List AND C^j ∉ LP then
          LP ← LP + C^j
          C^j_ DirectSubsequence_List ← C^j_ DirectSubsequence_List + C^i
          C^j_Offspring_List ← C^j_Offspring_List + C^i + C^i_Offspring_List
          C^j_JumpsToGoal = C^i_JumpsToGoal + 1
      Else If C^j ∉ PC_List AND C^j ∈ LP AND C^j ∉ C^i_Offspring_List then
          C^j_ DirectSubsequence_List ← C^j_ DirectSubsequence_List + C^i
          C^j_Offspring_List ← C^j_Offspring_List + C^i + C^i_Offspring_List
          If C^j_JumpsToGoal < C^i_JumpsToGoal + 1 then
              For each C^k ∈ LP_After_C^j
                  C^k_JumpsToGoal = Max(C^k_DirectSubsequence_JumpsToGoal) + 1
Return LP
End
```

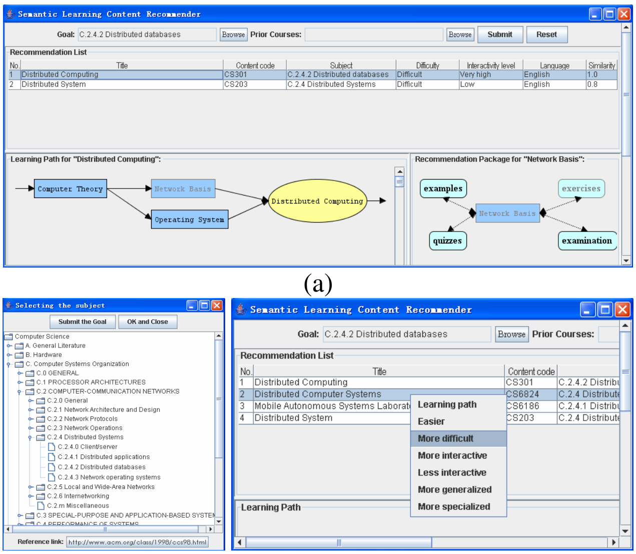
4.4 Recommendation Augmentation

While studying the main course, the learner usually needs to refer to some appendant contents within the course. For instance, when given a concept, the learner hopes to see some examples about it so as to strengthen his understanding, and after a section finishes, the learner may want to take a quiz to verify whether he has mastered the knowledge in the section. In our system, we provide recommendation augmentation with references to examples, exercises, quizzes, and examination related with the main course that the user is currently studying. This is accomplished by aggregating the contents through the properties of "hasExample", "hasExercise", "hasQuiz", and "hasExamination". Then the system provides links for such appendant contents along the main course. With the recommendation augmented, the learner needs merely to click on a button rather than looking up such contents in a large space by himself.

5 Prototype Implementation and Experiment

With the proposed recommendation approach, we built a semantic learning content recommender system. It was developed with Java (JDK1.5).

Fig. 5 shows several client-side interfaces for the semantic content recommendation. Fig. 5a is the main interface. It mainly consists of four parts. The top part provides interface for the learner to input learning goal and select the courses already learned. To ease the learning goal input, we provide a subject tree for the learner to choose one. Here the subject tree is automatically generated from ACM taxonomy (see Fig. 5b). Then, the recommendation list is presented below. Here the learner can refine the result through 6 options (i.e., "Easier", "More difficult", "More interactive", "Less interactive", "More generalized", and "More specialized", see Fig. 5c). The learning path is generated and shown in the left bottom column, while the recommendation package for the content selected from the path is presented in the right bottom column.



(a)

(b) (c)

Fig. 5. Client-side interfaces for the semantic content recommendation

We tested the overhead of the semantic content recommendation in terms of response time. The experiment was deployed on a PC with 1.60GHz Pentium 4 CPU and 1GB memory running Windows XP. The running time for each step, i.e., content recommendation list generation, recommendation refining, learning path generation and package generating, was an average value of 10 runs. We observed that the time for content recommendation list generating was the largest, say 78 ms; the learning path generation took 16ms; both of the refining and package generating cost less than 1 ms. The total time for semantic recommendation is therefore less than 100ms. Through this experiment, we could conclude that our approach is light-weight and feasible to be deployed.

6 Conclusion and Future Work

As the amount of electronic course content becomes very large, providing adaptive and personalized content recommendation is significant for today's e-learning systems. In this paper, we present a semantic recommendation approach for learning content based on ontology. For future work, we plan to incorporate additional learner contexts, e.g., available learning time, location, learning style, and learning interests into the recommendation process in order to make the system more comprehensive and intelligent. We also plan to consider the shared-knowledge among group members so as to recommend content to a group of learners [15].

Acknowledgement

This work was partially supported by the Ministry of Education, Culture, Sports, Science and Technology, Japan under the projects of "Development of Fundamental Software Technologies for Digital Archives" and "Cyber Infrastructure for the Information-explosion Era".

References

1. Fischer, G.: User Modeling in Human-Computer Interaction. User Modeling and User-Adapted Interaction 11(1/2), 65–86 (2001)
2. Yu, Z., et al.: Supporting Context-Aware Media Recommendations for Smart Phones. IEEE Pervasive Computing 5(3), 68–75 (2006)
3. Schmidt, A.: User Context Aware Delivery of E-Learning Material: Approach and Architecture. Journal of Universal Computer Science 10(1), 28–36 (2004)
4. Adomavicius, G., Tuzhilin, A.: Toward the Next Generation of Recommender Systems: A Survey of the State-of-the-Art and Possible Extensions. IEEE Transactions on Knowledge and Data Engineering 17(6), 734–749 (2005)
5. Baloian, N., et al.: A Model for a Collaborative Recommender System for Multimedia Learning Material. In: de Vreede, G.-J., Guerrero, L.A., Marín Raventós, G. (eds.) CRIWG 2004. LNCS, vol. 3198, pp. 281–288. Springer, Heidelberg (2004)
6. Shen, L., Shen, R.: Ontology-Based Learning Content Recommendation. International Journal of Continuing Engineering Education and Life-Long Learning 15(3/4/5/6), 308–317 (2005)
7. Simon, B., Mikls, Z., Nejdl, W., Sintek, M., Salvachua, J.: Smart Space for Learning: A Mediation Infrastructure for Learning Services. In: WWW 2003, May 2003, Hungary (2003)
8. Bomsdorf, B.: Adaptation of Learning Spaces: Supporting Ubiquitous Learning in Higher Distance Education. In: Dagstuhl Seminar Proceedings 05181, Mobile Computing and Ambient Intelligence: The Challenge of Multimedia (2005)
9. Paraskakis, I.: Ambient Learning: a new paradigm for e-learning, m-ICTE2005, Spain, pp. 26–30 (2005)
10. Nabeth, T., et al.: InCA: A Cognitive Multi-Agents Architecture for Designing Intelligent & Adaptive Learning Systems. ComSIS Journal 2(2), 99–114 (2005)
11. Gruber, T.: A Translation Approach to Portable Ontology Specification. Knowledge Acquisition 5(2), 199–220 (1993)
12. McGuinness, D.L., Harmelen, F.: OWL Web Ontology Language Overview, W3C Recommendation (2004)
13. Sowa, J.F.: Conceptual Structures. Addison-Wesley, Reading, MA (1984)
14. Burke, R.: Hybrid Recommender Systems: Survey and Experiments. User Modeling and User-Adapted Interaction 12(4), 331–370 (2002)
15. Yu, Z., et al.: TV Program Recommendation for Multiple Viewers Based on User Profile Merging. User Modeling and User-Adapted Interaction 16(1), 63–82 (2006)