Semantic Web Recommender Systems

Cai-Nicolas Ziegler

Institut für Informatik, Universität Freiburg,
Georges-Köhler-Allee, Gebäude 51,
79110 Freiburg i.Br., Germany

cziegler@informatik.uni-freiburg.de

Abstract. Research on recommender systems has primarily addressed centralized scenarios and largely ignored open, decentralized systems where remote information distribution prevails. Absence of superordinate authorities having full access and control introduces some serious issues requiring novel approaches and methods. Hence, our primary objective targets succesful deployment and leverage of recommender system facilities for Semantic Web applications, making use of novel technologies and conceptions and integrating them into one coherent framework.

1 Introduction

Automated recommender systems [1] intend to provide people with recommendations of products they might appreciate, taking into account their past ratings profile and history of purchase or interest. Most succesful systems apply social filtering techniques [2], dubbed collaborative filtering [3]. These systems identify most similar users and make recommendations based upon products people utterly fancy. Unfortunately, common collaborative filtering methods fail when transplanted into decentralized scenarios. Analyzing issues prevalent to these domains, we believe that two novel approaches may alleviate prevailing problems, namely trust networks and taxonomy-based profile generation. One aspect of our work hence addresses the conception of suitable components, specifically tailored to suit our decentralized setting, while another regards the seamless integration of latter building bricks into one single, unified framework. Empirical analysis and performance evaluations are conducted at all stages.

2 Research Issues

Deploying recommender systems on the Semantic Web implies diverse, multifaceted issues, some of them being inherent to decentralized systems in general, others being novel. Hereby, our devised Semantic Web recommender system performs all recommendation computations *locally* for one given user. Its principal difference from generic, centralized approaches refers to information storage, supposing all user and rating data *distributed* throughout the Semantic Web. We thus come to identify several research issues:

- **Ontological commitment.** The Semantic Web is characterized by machine-readable content distributed all over the Web. In order to ensure that agents can understand and reason about latter information, semantic interoperability via ontologies or common content models must be established. For instance, FOAF [4], an acronym for "Friend of a Friend", defines an ontology for establishing simple social networks and represents an open standard agents can rely upon.
- **Interaction facilities.** Decentralized recommender systems have primarily been subject to multi-agent research projects. Hereby, environment models are agent-centric, enabling agents to directly communicate with their peers and thus making synchronous message exchange feasible. The Semantic Web, being an aggregation of distributed metadata, constitues an inherently data-centric environment model. Messages are exchanged by publishing or updating documents encoded in RDF, OWL, or similar formats. Hence, communication becomes restricted to asynchronous message exchange.
- **Security and credibility.** Closed communities generally possess efficient means to control user identity and penalize malevolent behavior. Decentralized systems, among those peer-to-peer networks, open marketplaces and the Semantic Web, likewise, cannot prevent deception and insincerity. Spoofing and identity forging thus become facile to achieve [5]. Hence, some subjective means enabling each individual to decide which peers and content to rely upon are needed.
- **Computational complexity and scalability.** Centralized systems allow for predicting and limiting community size and may thus tailor their filtering systems to ensure scalability. Note that user similarity assessment, which is an integral part of collaborative filtering [3], implies computation-intensive processes. The Semantic Web will once contain millions of machine-readable homepages. Computing similarity measures for all these "individuals" thus becomes infeasible. Consequently, scalability can only be ensured when restricting latter compuations to sufficiently narrow neighborhoods. Intelligent prefiltering mechanisms are needed, still ensuring reasonable recall, i.e., not sacrificing too many relevant, like-minded agents.
- **Low profile overlap.** Interest profiles are generally represented by vectors indicating the user's opinion for every product. In order to reduce dimensionality and ensure profile overlap, some centralized systems like Ringo [6] require users to rate *small subsets* of the overall product space. Others recommenders, among those GroupLens and MovieLens [7], operate in domains where product sets are comparatively small. On the Semantic Web, virtually no restrictions can be imposed on agents regarding which items to rate. Hence, new approaches to ensure profile overlap are needed in order to make profile similarity measures meaningful.

3 Proposed Approach

Endeavors to ensure semantical interoperability through ontologies constitute the cornerstone of Semantic Web conception and have been subject to rife research projects. We do not concentrate our efforts on latter aspect but suppose data compatibility from the outset. Our interest rather focuses on handling computational complexity, security, data-centric message passing, and profile vector overlap. Hereby, our approach proposed builds upon two fundamental notions, namely taxonomy-based interest profile assembling and trust networks. Exploitation of synergies of both intrinsically separate concepts helps us leverage recommender system facilities into the Semantic Web.

3.1 Information Model

Semantic Web infrastructure defines interlinked documents comprising machine-readable metadata. Our information model presented below well complies with its design goals and allows facile mapping into RDF, OWL, etc.:

- **Set of agents** $A = \{a_1, a_2, \ldots, a_n\}$. Set A contains all agents part of the community. Globally unique identifiers are assigned through URIs.
- **Set of products** $B = \{b_1, b_2, \ldots, b_m\}$. All products considered are comprised in set B. Hereby, unique identifiers may refer to product descriptions from an online shop agreed upon, such as Amazon, or globally accepted codes, like ISBNs in case of books.
- **Set of partial trust functions** $T = \{t_1, t_2, \ldots, t_n\}$. Every agent $a_i \in A$ has one partial trust function $t_i : A \rightarrow [-1, +1]^{\perp}$ that assigns continuous trust values to its peers. Functions $t_i \in A$ are partial since agents generally only rate small subsets of the overall community, hence rendering t_i sparse:

  $$t_i(a_j) = \begin{cases} p, & \text{if trust}(a_i, a_j) = p \\ \perp, & \text{if no trust for } a_j \text{ from } a_i \end{cases} \tag{1}$$

  We define high values for $t_i(a_j)$ to denote high trust from a_i in a_j, and negative values to express distrust, respectively. Values around zero indicate absence of trust, not to be consfused with explicit distrust [8].
- **Set of partial rating functions** $R = \{r_1, r_2, \ldots, r_n\}$. In addition to functions $t_i \in T$, every $a_i \in A$ has one partial function $r_i : B \rightarrow [-1, +1]^{\perp}$ that expresses his liking or dislike of product $b_j \in B$. No person can rate every available product, so functions $r_i \in B$ are necessarily partial.

  $$r_i(b_j) = \begin{cases} p, & \text{if rates}(a_i, b_j) = p \\ \perp, & \text{if no rating for } b_j \text{ from } a_i \end{cases} \tag{2}$$

  Intuitively, high positive values for $r_i(b_j)$ denote that a_i highly appreciates b_j, while negative values express dislike, respectively.
- **Taxonomy C over set $D = \{d_1, d_2, \ldots, d_l\}$.** Set D contains categories. Each category $d_k \in D$ represents one specific topic that products $b_j \in B$ may fall into. Hereby, topics can express broad or narrow categories. Taxonomy C arranges all $d_k \in D$ in an acyclic graph by imposing partial subset order $\subseteq$ on D, similar to class hierarchies known from object-oriented languages. Hereby, *inner* topics $d_k \in D$ with respect to C are all topics having subtopics,

i.e., an outdegree greater zero. On the other hand, *leaf* topics are topics with zero outdegree, i.e., most specific categories. Furthermore, taxonomy C has exactly one top element $\top$, which represents the most general topic and has zero indegree.

– **Descriptor assignment function** $f : B \rightarrow 2^D$**.** Function f assigns a set $D_i \subset D$ of product topics to every product $b_i \in B$. Note that products may possess *several* descriptors, for classification into one single category generally entails loss of precision.

We suppose all information about agents a_i, their trust relationships t_i and ratings r_i stored in machine-readable homepages distributed throughout the Web. Contrarily, taxonomy C, set B of products and descriptor assignment function f must hold *globally* and therefore offer public accessibility. Central maintenance of latter information hence becomes inevitable. Later on, we will demonstrate that such sources of information for product categorization already exist for certain application domains.

3.2 Trust Neighborhood Formation

Trust neighborhhod computation constitutes the first pillar of our approach. Clearly, neighborhoods are subjective, reflecting every agent $a_i's$ very beliefs about the accorded trustworthiness of immediate peers. Trust makes automatic recommendation generation for a_i secure, only relying upon opinions from peers that a_i deems trustworthy. Note that in general, collaborative filtering tends to be highly susceptive to manipulation. For instance, malicious agents a_j can accomplish high similarity with a_i by simply copying its profile. Marsh [8] already indicated that trust makes agents "less vulnerable to others". However, for our scenario, trust also serves another purpose, namely that of similarity filtering. Recent studies [9] have provided empirical evidence that people tend to rely upon recommendations received from trusted fellows, i.e., friends, family members etc., more than upon online recommender systems. Ongoing research [5] has revealed that trust and interest profiles tend to correlate, justifying trust as an appropriate supplement or surrogate for collaborative filtering.

Trust neighborhood detection for a_i implies *computing* trust values for peers a_j not directly trusted by a_i, but one of the peers latter agents trusts directly and indirectly. Note that functions $t_i(a_j)$ are commonly sparse, providing values for only few a_j compared to A's overall community size. Numerous scalar metrics [10,11] have been proposed for computing trust between two given individuals a_i and a_j. However, our approach requires metrics that compute *nearest trust-neighbors*, and not evaluate trust values for any two given agents. We hence opt for local group trust metrics [12], which have only been attracting marginal interest until now. The most important and most well-known local group trust metric is Levien's Advogato metric [11]. However, latter metric can only make *boolean* decisions with respect to trustworthiness. Appleseed [12], our own novel proposal for local group trust computation, allows more fine-grained analysis, assigning continuous trust *ranks* for peers within trust computation range. Its principal concepts derive from spreading activation models [13]. Appleseed operates on partial trust graph information, exploring the social network within predefined ranges only and allowing the neighborhood detection process ro retain scalability. Hereby, high ranks are accorded to trustworthy peers, i.e., those agents which are largely trusted by others with high trustworthiness. These ranks are used later on for selecting agents deemed suitable for making recommendations.

3.3 Similarity-based Filtering

The second processing step performs collaborative filtering over all peers whose trustworthiness lies above some given threshold. Collaborative filtering intends to track most similar peers, considering the principal's history of interests. Hereby, we overcome low profile overlap by introducing taxonomy-based profile generation [5]. Common collaborative filtering approaches apply Pearson's correlation coefficient [6,3] to compute similarity between *product* vectors. Considering the domain of books, the probability that two persons have read several same books becomes considerably low. Category-based collaborative filtering [14] and related methods reduce dimensionality by generating vectors containing *categories*, along with information about the peer's liking and dislike for each of these. However, the more fine-grained latter categories are defined, the less profile overlap we may expect. Furthermore, relationships and mutual impact between categories become lost.

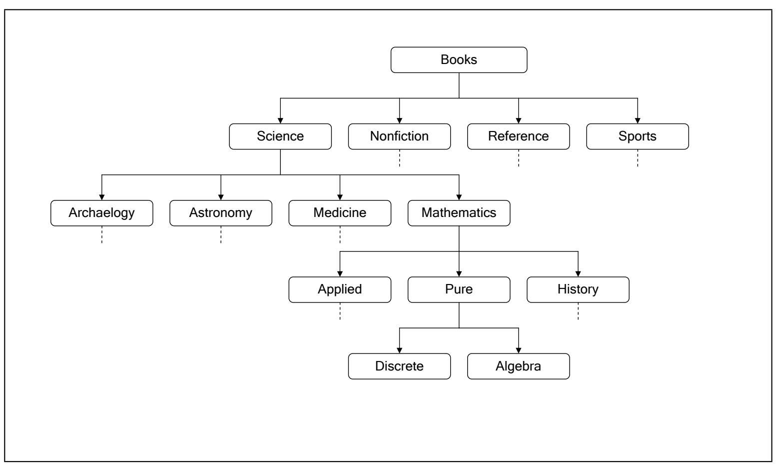

Fig. 1. Small fragment from the Amazon book taxonomy

Taxonomy-based Profile Generation. We are investigating taxonomy-aided generation of interest profiles [5], inspired by Middleton's ontology-enhanced content-based filtering [15]. Categories still play an important role, but we have them arranged in taxonomy C and not separate from each other. Items b_j bear topic descriptors $d_{j_k} \in f(b_j)$ that relate products b_j to taxonomic nodes. Several classifications per item are possible, hence $|f(b_j)| \geq 1$. Each item the user likes infers some interest score for those $d_{j_k} \in f(b_j)$. Since these categories d_{j_k} are arranged in taxonomy C, we can also infer fractional interest for all *super-topics* of d_{j_k}. Hereby, remote super-topics are accorded less interest score than super-topics close to d_{j_k}. For simplicity, suppose C tree-structured and assume that $(p_0, p_1, \ldots, p_q)$ gives the path from top element $p_0 = \top$ to node $p_q = d_{j_k}$. Function $\text{sib}(p)$ returns the number of p's siblings, while $\text{sco}(p)$ returns its score:

$$\forall m \in \{0, 1, \ldots, q-1\} : \text{sco}(p_m) = \frac{\text{sco}(p_{m+1})}{\text{sib}(p_{m+1}) + 1} \tag{3}$$

Scores are normalized, i.e., all topic score that a_i's profile assigns to nodes from taxonomy C amounts to some fixed value s. Hence, high product ratings from agents with short product rating histories have higher impact on profile generation than product ratings from persons issuing rife ratings. Score s is divided evenly among all products that contribute to a_i's profile makeup.

Example 1 (Topic score assignment). Suppose the taxonomy given in Figure 1 which represents a tiny fragment from the Amazon book taxonomy. Let user a_i have mentioned 4 books, namely *Matrix Analysis*, *Fermat's Enigma*, *Snow Crash*, and *Neuromancer*. For *Matrix Analysis*, 5 topic descriptors are given, one of them pointing to leaf topic *Algebra* within our small taxonomy. Suppose that $s = 1000$ defines the overall accorded profile score. Then the score assigned to descriptor *Algebra* amounts to $s \,/\, (4 \cdot 5) = 50$. Ancestors of leaf *Algebra* are *Pure*, *Mathematics*, *Science*, and top element *Books*. Score 50 hence must be divided among these topics according to Equation 3. Score 29.087 becomes accorded to topic *Algebra*. Likewise, we get 14.543 for topic *Pure*, 4.848 for *Mathematics*, 1.212 for *Science*, and 0.303 for top element *Books*. These values are then used to update the profile vector of user a_i.

Success or failure of our approach largely depends upon taxonomy C used for classification. The more thoroughly crafted and fine-grained latter taxonomy, the more meaningful our profile information becomes. Clearly, topic descriptors $f(b_j)$ for products b_j must be chosen skillfully, too. Thanks to inference of fractional interest for super-topics, one may establish high user similarity for users which have not even rated one single product in common. According to our scheme, the more score two profiles have accumulated in same branches, the higher their computed similarity.

Similarity Computation. Interest profiles form the grounding for collaborative filtering, which computes similarity between users. For our approach, we apply common nearest-neighbor techniques, namely Pearson's coefficient [6,3] and cosine distance from Information Retrieval. Hereby, profile vectors map *category* score vectors from C instead of plain product-rating vectors. High similarity evolves from interest in many identical or related branches, whereas negative correlation indicates diverging interests. For instance, suppose a_i reads literature about *Applied Mathematics* only, and a_j about *Algebra*, then their computed similarity will be high, considering significant branch overlap from node *Mathematics* onward.

3.4 Rank Synthesization and Recommendations

Trust neighborhood computation and collaborative filtering return two diverse rankings for every agent a_j within our bounding trust neighborhood. One must now merge trust rank and similarity rank into one single measure, i.e., its overall rank weight.

We have not attacked latter issue yet. Moreover, besides selecting most suitable peers a_j from which to receive recommendations, one must determine *products* mentioned by latter a_j most favorable for recommendation. Numerous alternatives are possible, like, for instance, every a_j voting for all its appreciated products $b_k \in r_j$ with its own rank weight. Products positively mentioned within *several* rating histories r_j of high weighted peers a_j thus have greater chance of being recommended. Other recommendation schemes, based upon content, are also possible. For instance, one might propose agent a_i products from categories that a_i has left untouched until now. Latter approach assumes that a_i might appreciate these new products since people with similar taste have told to like them. Incentive for trying new product groups becomes created.

Recommendation-making opens numerous alternatives one can take. Our future research will thus focus on finding most promising ones and, what will become likewise important, on trying to match these approaches against each other within an experimental framework allowing for some quantitative analysis.

4 Real-world Deployment

Section 3.1 has exposed our envisioned information infrastructure. We will show that such an architecture may actually come into life and become an integral part of the Semantic Web. For instance, some initial projects towards deploying and maintaining decentralized trust networks are already under way: FOAF defines machine-readable homepages based upon RDF and allows weaving acquaintance networks. Golbeck [4] has proposed some modifications making FOAF support "real" trust relationships instead of mere acquaintanceship.

Moreover, FOAF seamlessly integrates with so-called "weblogs", which are steadily gaining momentum. These personalized "online diaries" are especially valuable with respect to product rating information. For instance, some crawlers extract certain hyperlinks from weblogs and analyze their makeup and content. Hereby, those referring to product pages from large catalogs like Amazon (*http://www.amazon.com*) count as implicit votes for these goods. Mappings between hyperlinks and some sort of unique identifier are required for diverse catalogs, though. Unique identifiers exist for *some* product groups like books, which are given “International Standard Book Numbers”, i.e., ISBNs. Efforts to enhance weblogs with explicit, machine-readable rating information have also been proposed and are becoming increasingly popular. For instance, BLAM! (*http://www.pmbrowser.info/hublog/*) allows creating book ratings and helps embedding these into machine-readable weblogs.

Besides user-centric information, i.e., agent a_i's trust relationships and product ratings, taxonomies for product classification play an important role within our approach. Luckily, these taxonomies exist for certain domains. Amazon defines an extensive, fine-grained and deeply-nested taxonomy for books containing more than $20,000$ topics. More important, Amazon provides books with subject descriptors referring to latter taxonomy. Similar taxonomies exist for DVDs and videos. Standardization efforts for product classification are channelled through the “United Nations Standard Products and Services Code” project (*http://www.unspsc.org/*). However, the UNSPSC's taxonomy provides much less information and nesting than, for instance, Amazon's taxonomy for books.

4.1 Mining Trust Statements and Ratings

We have created an experimental environment simulating the infrastructure proposed above. Hereby, we mined rife information from various trust-aware online communities like All Consuming (*http://www.allconsuming*), and Advogato (*http://www.advogato.org*), extracting information about approximately 9,100 users, their trust relationships and implicit product ratings. Ratings were obtained from All Consuming only. Moreover, we captured Amazon's huge book taxonomy and categorization data about $9,953$ books that All Consuming community members have mentioned. Tailored crawlers search the Web for weblogs and ensure data freshness. All our experiments and empirical evaluations were based upon latter “real-world” data.

5 Related Work

Recommender systems have begun attracting major research interest during the early nineties [3]. Nowadays, commercial and industrial systems are rife and wide-spread, detailed comparisons concerning features and approaches are given in [16]. Recommender systems differ from each other mainly through their filtering method. Hereby, distinctions between three types of filtering systems are made [3], namely collaborative, content-based and economic. Collaborative filtering systems [6] generate recommendations obtained from persons having similar interests. Content-based filtering only takes into account the content of products, based upon metadata and extracted features. Economic filtering has seen little practical application until now and exerts marginal impact only. Modern recommender systems are hybrid, combining both content-based and collaborative filtering facilities in one single framework. Fab [17] counts among the first popular hybrid systems, more recent approaches have been depicted in [18], and [15].

Our filtering approach, comprising taxonomy-based profile generation and similarity computation, also exploits both content-based and collaborative filtering facilities. Trust networks add another supplementary level of filtering.

Initial attempts have been taken towards transplanting recommender systems into decentralized scenarios. Olsson [19] offers an extensive overview of existing approaches. Montaner [20], and Chen et al. [21] devise agent-based approaches, where agents acquire knowledge about other peers from interaction experience. Hereby, reputation evolves over time and simple trust relationships become tied.

6 Future Directions

Our past efforts have mainly focused on designing suitable trust metrics for computing trust neighborhoods [12], and conceiving metrics for making collaborative filtering applicable to decentralized architectures [5]. Moreover, we have shaped and synthesized an extensive infrastructure based upon "real-world" data from various communities and online stores.

Until now, analysis has been largely confined to the book domain only. Future research will also include movies and other specific product groups and investigate intrinsic differences between these groups. For instance, Amazon's taxonomy for DVD classification contains more topics than its book counterpart, though being less deep. We would like to better understand the impact that taxonomy structure may have upon profile generation and similarity computation. Furthermore, we are currently investigating applicability of taxonomy-based profile generation for automated stereotype generation and efficient behavior modelling. Efforts for extracting rife usage and profile information from various other communities are well under way.

Merging ranks from both filtering paradigms into one metric and recommendation generation have remained untouched until now. Thorough empirical analysis will be required for selecting most appropriate alternatives and integrating them into our recommender application.

References

1. Resnick, P., Varian, H.: Recommender systems. Communications of the ACM **40** (1997) 56–58
2. Kautz, H., Selman, B., Shah, M.: Referral web: Combining social networks and collaborative filtering. Communications of the ACM **40** (1997) 63–65
3. Goldberg, D., Nichols, D., Oki, B., Terry, D.: Using collaborative filtering to weave an information tapestry. Communications of the ACM **35** (1992) 61–70
4. Golbeck, J., Parsia, B., Hendler, J.: Trust networks on the semantic web. In: Proceedings of Cooperative Intelligent Agents, Helsinki, Finland (2003)
5. Ziegler, C.N., Lausen, G.: Analyzing correlation between trust and user similarity in online communities. In Jensen, C., Poslad, S., Dimitrakos, T., eds.: Proceedings of the 2nd International Conference on Trust Management. Volume 2995 of LNCS., Oxford, UK, Springer-Verlag (2004) 251–265

6. Shardanand, U., Maes, P.: Social information filtering: Algorithms for automating "word of mouth". In: Proceedings of the ACM CHI'95 Conference on Human Factors in Computing Systems. Volume 1. (1995) 210–217
7. Miller, B., Albert, I., Lam, S., Konstan, J., Riedl, J.: MovieLens unplugged: Experiences with an occasionally connected recommender system. In: Proceedings of the ACM 2003 Conference on Intelligent User Interfaces (Accepted Poster), Chapel Hill, NC, USA, ACM (2003)
8. Marsh, S.: Formalising Trust as a Computational Concept. PhD thesis, Department of Mathematics and Computer Science, University of Stirling, Stirling, UK (1994)
9. Sinha, R., Swearingen, K.: Comparing recommendations made by online systems and friends. In: Proceedings of the DELOS-NSF Workshop on Personalization and Recommender Systems in Digital Libraries, Dublin, Ireland (2001)
10. Beth, T., Borcherding, M., Klein, B.: Valuation of trust in open networks. In: Proceedings of the 1994 European Symposium on Research in Computer Security. (1994) 3–18
11. Levien, R., Aiken, A.: Attack-resistant trust metrics for public key certification. In: Proceedings of the 7th USENIX Security Symposium, San Antonio, Texas, USA (1998)
12. Ziegler, C.N., Lausen, G.: Spreading activation models for trust propagation. In: Proceedings of the IEEE International Conference on e-Technology, e-Commerce, and e-Service, Taipei, Taiwan, IEEE Computer Society Press (2004)
13. Quillian, R.: Semantic memory. In Minsky, M., ed.: Semantic Information Processing. MIT Press, Boston, CA, USA (1968) 227–270
14. Sollenborn, M., Funk, P.: Category-based filtering and user stereotype cases to reduce the latency problem in recommender systems. In: Proceedings of the Sixth European Conference on Case-based Reasoning. Volume 2416 of LNCS., Aberdeen, GB (2002) 395–405
15. Middleton, S., Alani, H., Shadbolt, N., De Roure, D.: Exploiting synergy between ontologies and recommender systems. In: Proceedings of the WWW2002 International Workshop on the Semantic Web. Volume 55 of CEUR Workshop Proceedings., Maui, HW, USA (2002)
16. Schafer, B., Konstan, J., Riedl, J.: Recommender systems in e-commerce. In: Proceedings of the 1st ACM Conference on Electronic Commerce, Denver, CO, USA, ACM Press (1999) 158–166
17. Balabanović, M., Shoham, Y.: Fab - content-based, collaborative recommendation. Communications of the ACM **40** (1997) 66–72
18. Huang, Z., Chung, W., Ong, T.H., Chen, H.: A graph-based recommender system for digital library. In: Proceedings of the Second ACM/IEEE-CS Joint Conference on Digital Libraries, Portland, OR, USA, ACM Press (2002) 65–73
19. Olsson, T.: Bootstrapping and Decentralizing Recommender Systems. PhD thesis, Uppsala University, Uppsala, Sweden (2003)
20. Montaner, M., López, B., de la Rosa, J.: Opinion-based filtering through trust. In Ossowski, S., Shehory, O., eds.: Proceedings of the Sixth International Workshop on Cooperative Information Agents. Volume 2446 of LNAI., Madrid, Spain, Springer-Verlag (2002) 164–178
21. Chen, M., Singh, J.P.: Computing and using reputations for internet ratings. In: Proceedings of the 3rd ACM Conference on Electronic Commerce, Tampa, FL, USA, ACM Press (2001) 154–162