Policies for Distributed User Modeling in Online Communities

Tariq Muhammad and Julita Vassileva
Department of Computer Science
University of Saskatchewan
110 Science Place Saskatoon, SK S7N5A9
CANADA
{tam706, jiv}@cs.usask.ca

Abstract. Allowing collaboration between online communities can result in a fragmented user profile. Each community will maintain a profile of user based on local context and policies. To express, discover, interpret and update these '*localized*' fragments, we propose to use policies defined by the community owners. When users move across communities, this movement can be regulated through *transfer policies*. Policies can be used to enforce the access rights and implement adaptations to users' status and roles. This policy–based user modeling approach is a variant of the purpose-based decentralized user modeling approach which computes user models and carries out adaptation on demand from fragmented user model data available in the different collaborating communities.

Keywords: Purpose-based user modeling, User policies, Online Communities.

1 INTRODUCTION

The basic purpose of online communities is to support social interactions and exchange of digital resources among people (Kimberly et al., 2003), (DeSouza and Preece, 2004). In the physical world, we see the movement of people from one place to another due to economic or social reasons. Such movement results in depopulation at one place and overpopulation at another. Although also being susceptible to user migration, online communities should not fall victim to this phenomenon. In the virtual world, the availability of technological tools such as web services, make the "virtual merger" of online communities possible. Still, the current designs of online communities do not focus on allowing collaboration among large online groups. Most existing communities are independent from each other; allow no sharing and/or interaction across online community borders, thus losing the potential advantage that the virtual world has over the physical world in terms of sharing time and space. Inter-community collaboration can help resolve this issue of participation and sustainability. One of the main design problems to ensure inter-community collaboration is the transfer of the user data, including the user identity and user model, across online communities.

Most of the online communities manage user models for multiple reasons, varying from authentication to personalization. Users create individual accounts in different communities and they have to start from scratch their participation and building their reputation in each community. This results in a fragmented user models across the communities. Even if the designers / moderators of two communities agree on exchange of contents, it is hard to transfer a user model across two sites. The reason is that different online community applications typically use different database organizations or different ontologies and can therefore not transfer and understand the user model data received upon request from another application (community). The need arises for a mechanism to create a user model on request just in time according to the current context.

Collaborating online communities face user modeling challenges similar to those in open environments with ubiquitous, service-oriented or agent-based applications. User models in these environments are fragments developed for the adaptation purposes of each service, agent or ubiquitous computing applications, and stored locally by these virtually or physically distributed applications (nodes). Some of the emerging challenges are interoperability, updating and synchronization of user models across these nodes, while preserving the autonomy of each application, service or agent. Applications have been generating and updating user models with the help of procedures. The procedural approach focuses on the algorithms of locating, deducing, and using user data for adaptation, rather than on the user data representation. However the procedural approach results in a use–specific model (Anderson, 1988), where changes in the user characteristics are hard to implement and collaboration always requires changes. Therefore, the opposite, declarative approach is the currently popular solution, where facts can be added and removed freely in a uniformly represented user model (expressed according to agreed upon ontology and language, ensuring interoperability) and applications can use a standard reasoning mechanism to make conclusions based on the information in user models. The declarative approach focuses on the expression of the user model instead of the discovery, interpretation and integration of user data, since these processes are standardized. However, in the autonomous and diverse online communities existing currently it is impossible to ensure such standardization.

User models in online communities are based on policies describing the role, status and rights of each user to ensure security, adaptation and awareness. These policies are implemented in procedures. The purpose-based user modeling approach proposed by Niu et al. (2004) can be implemented through policies as an abstraction layer ensuring functionalities like a shared user data taxonomy/ontology, security, and discovery of user information fragments, interpretation and integration of user models. While not focusing on the standard representation of user model data, this approach does not preclude it, and it puts emphasis on the processing of the data in context. Therefore, we believe that this approach is suitable for the problem of sharing user data in collaborating online communities. The paper explains how policies can be used for managing, transferring user data in multi-community environment Comtella (http://umtella.usask.ca/um).

2 RELATED WORK

The Comtella system is a web-based online community framework, which was created in the MADMUC lab to support resource sharing and discussion by students (Cheng and Vassileva 2005). The users in Comtella are assigned different *status* to reward them for participation. The status is computed based on the number of desirable actions the users perform and is rewarded with certain privileges. The users of Comtella can take also different *roles*. They can create new communities and become *owners* of communities. In the same time the architecture of Comtella has evolved from a single web-based system for one community sharing URLs for different topics (each topic being a focus of the entire community for a given time), to a single system hosting many communities created by different users (each focused on a different topic), and finally to a multi-node system, consisting of many systems at different websites, hosted by different organizations and administered by users in the roles of *administrators*. The new design of Comtella allows communities to be hosted in different web sites (nodes). Communities can collaborate within and across nodes. Members of one community can join other communities and transfer there their old user profile from their previous community; they can maintain different roles and statuses (with their associated rights and privileges) in each community.

One of the basic purposes of user models in multi-user applications, apart from personalization, is to ensure security of computer systems. The Role-Based Access Control (RBAC) system was created for the first multi-user computer environments and has been used widely in web-enabled applications. In role-based access control systems users are associated with a roles defined according to the operational needs of groups and organizations. Rights of access are defined at the role level. Users can work in one or more roles and can perform actions associated with these roles (Mohammed and Dilts 1994, Sandhu and Park 1998, Park, Sandhu and Ahn 2001).

Reward-based communities like Comtella cannot model all users with RBAC, since these applications model not just roles, but also user goals, capabilities, user attitudes and knowledge (Kass and Finn 1988). Kagal et al. (2001) proposed an ontology-based RBAC approach for pervasive computing environments. This approach allows not only representing role hierarchies but also other user properties, which are expressed in XML language. Denaux et al. (2005) proposed an ontology-based user modeling to allow for interoperability and overcome the “cold start” problem.

Many applications keep their user model hidden from the user. However, applications such as learning environments often deploy an open learner (user) model, so that the learner can interact with the model to reflect on its content or to correct errors (Bull and Pain 1995, Bull 1997, Vassileva et al., 1999). A user model framework that is based on user policies can open the user model both for the user and for other systems. Policies will not only communicate the current status of the user but also explain why she has gained this status.

Agent-based software environments, mobile applications and online communities can not work with monolithic user models as each point maintains a local profile of the user according to context. Distributed user modeling or decentralized user modeling is an option for such environments. In this approach user information is scattered around in independent and autonomous agents as user model fragments. Each agent develops these fragments according to its context and preferences. The properties and issues of these '*fragmented, relativized, local and often quite shallow*' user models is described by Vassileva et al. (1999, 2003). The active modeling approach is a decentralized user modeling for learning environments (McCalla et al, 2000). Active learner modeling can be combined with open user models to create small fragmented models just in time when requested by the user (Hansen and McCalla, 2003). Purpose-based user modeling (Niu et al, 2004) is an approach that involves computing distributed and fragmented user models from various decentralized sources for a specific purpose. The purpose consists of a process and the user data types it requires as input and output. The process computes new user model data type and/or provides a certain application-dependent adaptation. Thus, a purpose is an independent processing unit, which can be applied to whatever fragmented user data is available at the moment from available sources. The purposes can work together in an anytime manner in a hierarchy based on abstraction. More specific purposes positioned towards the leaf nodes are executed when more data from fragmented sources is available while more general purposes near the root typically demand less or easier to access data. The purpose-based modeling approach has two advantages: speed and providing a local context for computing the model fragment and adaptation.

Purpose-based user modeling can be implemented using policies instead of purposes to compute user models on the fly in online communities. The policies define the rights and privileges of users in the new communities that they join, so they do not have to start from scratch as new users. The policy document, like a purpose (Niu et al, 2004), describes a procedure, but it is also declarative in some sense, since it is modular, human-readable and editable according to the wishes of the community owner or node administrator. A policy provides all the relevant information for computing a user model and adaptation of the functionality and interface to a given type of user in a given context, e.g. when visiting a community. Through appropriate policies online communities can collaborate and transition of users across communities can be made smoother. In the next section we explain how a policy driven framework can implement a purpose-based user modeling approach for collaborating online communities.

3 POLICY-DRIVEN ONLINE COMMUNITIES

Allowing users to move across communities results in a user profile fragments in all of the visited communities and requires interoperability of their user modeling components and a trust relationship among the collaborating communities. A typical user joins one community according to her primary interest. However the same user can visit other communities of marginal interest. In Comtella the user models are represented in a database which is updated according to user policies. By inspecting the community policies a user can understand the reason for the current state of the user model in a given context (provenance). Different policies command the transfer of user data along with the user's identity to any new community where a new user model can be established according to the context. Policies in Comtella determine the access rights, the status of users and user roles (and the privileges associated with roles and status) both in the home communities of the users and in new communities they are visiting. Policies can be created only by users in a particular role – the role of community owner (the user who created the community). The owner of a community creates and manages four types of policies: access control policies, status policies, role policies and transfer policies. Examples of these policies are shown in Fig. 1.

Status Policies

A. Policy to update user participation		Description
Policy Type:	Status	This is a policy defines how the measure of user participation (UP) is calculated in community "Pictures". UP is an important derived user data in the user model. Specifically, the policy defines weights for the different user activities (primary user data, e.g. *pn* – number of shared papers, *pq* – average rating of user's shared papers, *m* – number of ratings given by user and *rq* – similarity of ratings given by user to the average rating of the paper) and the formula for the UP calculation.
Effective Date	Jan 10, 2007	
Node	http://kardam.usask.ca	
Community id:	1	
Community Title:	Pictures	
Weight for Paper Quantity (Wpn)	3	
Weight for Paper Quality (Wpq)	4	
Weight for Rating Quantity (Wrn)	3	
Weight for Rating Quality (Wrq)	4	
Action	UP:= *Wpn*pn+Wpq*pq+Wrn*rn+Wrq*rq*	

B. Policy to calculate user status level				Description
Level	Description	Start Value	End Value	This policy is used to classify the user into appropriate status level, i.e. to compute the user status (US) depending on his/her participation measure (UP). It sets the margins of each status level.
1	Gold	700	1000	
2	Silver	500	699	
3	Bronze	300	499	
4	Plastic	0	299	

Action	If (Plastic(StartValue)<= UP <= Plastic(Endvalue)) →US:=Plastic
	If(Bronze(StartValue)<=UP <=Bronze(Endvalue)) →US:=Bronze
	If (Silver(StartValue)<= UP <= Silver(Endvalue)) →US:=Silver
	If (Gold(StartValue)<= UP <= Gold(Endvalue)) →US:=Gold			

C. Policy for Status Permissions		√ Action allowed × Action not allowed				Description
Level	Description	Share link	Share File	Post	Rate	This policy defines user permissions based on the user status level. These permissions are used for interface adaptation and the effect is to disable certain options to the user.
1	Gold	√	√	√	√	
2	Silver	√	√	√	√	
3	Bronze	√	×	√	√	
4	Plastic	√	×	√	×	
Action	If US==Plastic or US==Bronze → disable "Share File" Interface widget If US==Plastic → disable "Rate" Interface widget					

Role Policies

D. Policy for Role Permissions		√ Action allowed × Action not allowed				**Description**
Level	Role	Delete link	Create Community	Edit Policy	Edit Role	This policy defines access rights and special permissions based on the role of the user. By editing this policy, the community owner can grant to users in different roles special rights and permissions for advanced actions such as "delete link", "create community", "edit policy" and "edit roles".
1	Owner	√	√	√	√	
2	Expert	√	√	√	×	
3	Operator	√	√	×	×	
4	Member	×	√	×	×	
Action	If UR==Member → disable "Delete Link", "Edit Policy", "Edit Role" widgets from the user interface If UR==Operator → disable "Edit Policy", "Edit Role" from interface If UR == Expert → disable "Edit Role" from interface					

Figure 1. Examples of different policies in Comtella.

Access control policies are rules representing conditions under which users can perform certain actions on a resource, such as reading, rating, replying, commenting, or deleting a posting. Usually access control policies are the basic policies that are used by higher level policies, such as status-, role- and transfer-policies to express specific decisions, e.g. allowing or disallowing a user request.

Status policies in Comtella implement the reward mechanism to stimulate desirable actions in the community (Bretzke and Vassileva, 2003). Status policies in Comtella (e.g. the one shown in Figure 1-A) define how the user participation metric is computed based on giving reward points for frequency and quality of certain desirable activities such as sharing and rating resources. Other status policies (E.g. Figure 1-B) describe the computation of the user status attribute. Community owners can manipulate the status policies for their communities and define their own user status levels (e.g. plastic, bronze, silver and gold or regular, prestige, elite) and their point thresholds. The two status policies described above result in changes in the user model. Other status policies define the access permissions that should be granted to users with a certain status. They result in adaptations of the interface that enable or disable certain functionality or look and feel (e.g. Figure 1-C). For example, the gold status users in Comtella have access to the gold-coloured interface frame, while plastic status users have access to the green-coloured interface.

Role policies define the conditions under which users with a given status can acquire a certain role and the accompanying rights and responsibilities. Like any organization online communities should manage a separation of duties. Role policies allow community owners to share the burden of community management with deserving community members. A community owner may designate a few members through either individual policy (by naming individuals) or through a selection-based policy (e.g. all gold-status members) to special roles, such as operators or experts. The moderator can assign special access rights to these roles, such as editing and deleting resources (see example policy in Figure 1-D). Role-based policies result in defining user groups based on their functional responsibilities such as expert, community moderator, and operator.

The three types of polices presented above define how to update the user model and what access rights to grant the user when she is working within her community. Each user in Comtella has a home community, which she can select from all communities hosted on the user's node when she starts using the system. It is expected that the user will contribute and participate mostly in her home community. The user accumulates a participation score which is represented in her main user model (the model related to her activity in her home community). The user's identity is also linked to her home community.

Users can search freely and find resources shared in other communities. In order to access and read these resources, they have to "visit" the other community. When a user moves from one community to another, for example, by requesting access to a resource in a new community, there is a question what rights and privileges, role and status this user should have in the new community. To govern movement of users across two communities, the communities must have a contract/agreement about the status, role and access rights of visiting users. These contracts are called transfer policies and can be unilateral (e.g. the owner of the receiving community defines the policy according to which to treat visitors from specific communities or in general) or bilateral (e.g. the two owners agree about mutual recognition of status, roles and rights). For example, the community owners may decide that visitors from the other community will be given automatically status with one level lower than the status they enjoy in their home community. In Comtella these policies are unilateral. If a user wants to visit a new community (e.g. to read an article posted in this community), she has to send a request to the owner of the new community. The community owner sets a transfer policy after reading the policy under which user was working in her home community. Comtella allows three options to the community owner: (I) enforce the current policy of the community; (II) allow the policy of the user's home community; and (III) define a new policy for visiting users from the user's home community. The definition of a new policy can be achieved by using different approaches. One may be to show the community owner the policies of both communities so that she can compare them and provide the owner with an editing tool allowing her to create a new policy, as shown in Figure 2. In this approach community owners can define new status levels and their respective thresholds. Another approach may be to declare one of the status slots of the community equal to one or more slots of the other community from where a user is coming. We have used the former approach as it provides finer grained control to community owners.

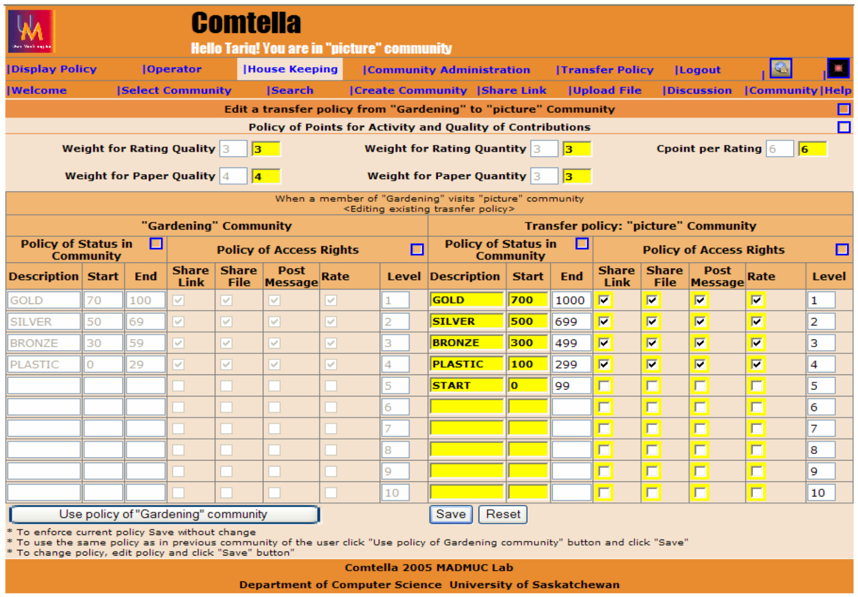

Figure 2: Editing a transfer policy in Comtella

One problem in many learning communities is the '*cold start*' (Denaux et al. 2004), (Sun and Vassileva, 2006) where the system fails to provide adaptation due to the lack of information about users when they first visit a community. With transfer policies the community does not have to wait for the accumulation of user information to offer customization and adaptation. Transfer policies can help in acquiring user model data about the previous experience of the user from other communities. These transfer policies provide a guideline whenever a user may visit a community for the first time. It will give the user a starting point to participate in the community instead of starting from scratch. The subsequent visits of the user will follow the same policy and update profile on every visit. Yet, through transferring back and forth across communities, users may find ways of increasing their status due to the inconsistencies between the policies in each community and too generous transfer policies. Therefore, the transfer policies for temporary visitors should be different from the transfer policies for users who want to make the community their new home community.

4 POLICIES IMPLEMENTING PURPOSES

Each community has a policy framework, which consists of:

- *shared view* used for all context and user data, both raw data as well as calculated user attributes;
- a set of user policies governing the community, each specifying the input data (about the user and context), the process and the output data (user model update or functionality / user interface adaptation);
- an execution mechanism running in a loop which selects an appropriate policy for the current user request and context and executes its process.

The policy framework ensures that the relevant policy is selected depending on the user request (which arrives on the *shared view*). The invoked policy in turn picks the required user data items (either raw data or user model data computed as output by other policies, in the same community or requested from other communities). There are many different policies in the set, which can be seen as managing different levels of decisions. For example, there are high-level policies that compute the role and status of the user in the community, using data received from other communities (which is either raw participation data or data computed by other policies). Lower-level policies control the user access rights using data about the user role or status computed by the higher-level policies. In this way, the framework provides both personalization and a simple security layer to protect against unauthorized users and actions.

Just like the purposes in purpose-based user modeling (Niu et al, 2004) a policy has three components: input, process and output. The input is either raw user data or user model data computed as output by other policies. For example, the input of a policy controlling user access to a community can be a user action attempting to access an item shared in the community. As another example, the input of a policy controlling user actions on community resources can be an action of a user attempting to rate a posting in the community. We call such raw data indicating user intentions a user request. A request consists of three parts: the subject, action and resource (Merrells, 2004) (OASIS, 2005) (Seth, 2004). Here "subject" is a primary identity key produced by a shared identity provider to which the collaborating communities have access or one of a federation of identify providers. This identity key can be used to fetch the user attributes hosted in the user database from both the current community (that is receiving request), from the home community of the user and from any other community which has data about this user. These user attributes can be inputs of another policy, for example, one that decides what status to grant the user in the community.

The process of a policy involves the algorithm that computes in context the output user model data or makes an adaptation decision. The process is executed by execution mechanism of the policy framework which retrieves the local and remote user profile, data required by the policy as input and places it in the *shared view*. The policy framework execution mechanism then computes the policy output data using the available input and current context data from the *shared view* and makes a decision, for example to allow / disallow the request or to adapt functionality or interface. For example, the process of an access control policy distinguishes between new users and local users (whose profiles are stored at the community). For a local user it retrieves the location of her user model, which becomes the output of the purpose and either grants or denies access depending on the role of the user. For new visitors it calls the appropriate transfer policy whose inputs match the user request and the current context and produces its output. The process of the transfer policy (using the user id as input) requests information from all other collaborating communities that have stored a model of this user and according to the mapping algorithm described by the community owner in the process of the transfer policy generates a local user model for the new user, which contains her status and role. This data will then be used as input by the community's access control, status and roles policies that decide about the user's rights and privileges.

5 DISCUSSION AND FUTURE WORK

Collaborating online communities have to deal with fragmented user models. These environments need interoperable, context- and purpose-sensitive user models. Online communities need a shared framework to express, discover, transfer and secure user models. User policies can be used for establishing the user's purpose and compute the required user model just in time. We propose a policy framework with the following advantages:

- Interactions between different communities will result in exchange of both users and contents, which otherwise would not be possible due to island nature of online communities (Harth et al., 2005), (Breslin et al., 2005). In this way there will be no necessity for each community to gain a critical mass of participation to be sustainable by itself.
- Transfer policies provide a starting point for customization for the user without '*cold start*'.

- Explicitly assigned roles for users lead to a more sophisticated user model, representing the context, purpose, trust and reputation of users within and across communities.
- People in other communities will feel more comfortable since ‘strangers’ will be allowed only after policy negotiation with their trusted domain/community.
- User policies are open and readable for the community members, so that they know the consequences of their actions and activity. Community owners can change the policies according to the changing needs of the communities.
- Availability of policy documents for a community system, owner and user work as tool to establish a trust between these entities.

The user policies implement a purpose-based user modeling approach. Yet there are some differences between the original purpose-based approach (Niu et al., 2004) and our approach. In Nui’s approach the whole hierarchy of purposes has access to the raw data items denoted as R1, R2……Rn coming from distributed sources. These raw data items include information that comes with the user’s request, information stored by application such as login information in database and data from peer assessment. The output of each purpose is transferred directly from one purpose to another depending on the hierarchical and sequence relationships between the purposes, which are pre-defined at design time. These pre-defined relationships limit the flexibility and reuse of purposes in Niu’s approach. In our approach, the outputs of policies (purposes) are placed on the *shared view* and can be used as inputs for other purposes together with the other raw data and context data. This allows for more flexibility.

The policy-based user modeling approach is currently being implemented in Comtella. The next step is to carry out experiments to evaluate its reliability, efficiency, scalability and also the ease of editing policies that the interface provides for community owners. This implementation will also provide a platform to study the dynamics of online community. We envisage carrying out the following studies in the future:

Study of single community: Comtella has been used for the study of reward mechanisms and its effects on the participation in communities. The flexible reward mechanism of the current implementation provides an opportunity to observe the effects of different reward strategies. For example what should be the parameter values at the start of the community to attract users and how the reward mechanism may be adjusted to achieve the quality in contributions of users in the later stages of community’s life? This information will be useful for defining the reward policies in the current and future deployments of Comtella and other reward-based communities.

Study of interaction between communities: Previous Comtella studies were focused on a single group and its dynamics. This implementation can be used to study both interactions within one community and interactions between communities. This study will capture the transfer of users between communities and will point out what factors trigger the transfer of users. The percentage of local versus visiting users in a community and statistics of the home-communities of visiting users will help to discover and develop relationships between communities. Knowledge about the movement of users between communities and factors contributing for the movement will be useful for both attracting and retaining users in future communities.

Study of user activity and sustainability: The study of contributions by local and visiting community members will help to appreciate the effects of community collaboration on its sustainability. This study will visualize activities such as *sharing* and *rating* by local and visiting community members. It would be interesting also to study the effects of policy-based user modeling on the *cold start* problem by comparing the time taken by local and visiting users to attain the top status in the community.

These studies will help us also understand better the strengths and limitations of policies as decentralized open user modeling approach and their benefits for enabling inter-community collaboration.

Acknowledgement: This work is supported by the NSERC Discovery Grant of the second author and by funding the NSERC/Cameco Chair for Women in Science and Engineering (Prairies).

REFERENCES

1. Anderson, J. R. (1988). The expert module. In M. Polson & J. Richardson (Eds.), Handbook of Intelligent Training Systems. Hillsdale, NJ: Erlbaum, 21-53.
2. Breslin, J. G. (2005). Towards Semantically-Interlinked Online Communities. Lecture Notes in Computer Science, vol. 3532, 500-515.
3. Bretzke H., Vassileva J. (2003) Motivating Cooperation in Peer to Peer Networks. Proceedings of User Modeling (UM'03), Johnstown, PA, June 22-26, Springer Verlag LNCS 2702, 2003, 218-227.
4. Bull, S., and Nghiem, T. (2002) Helping Learners to Understand Themselves with a Learner Model Open to Students, Peers, and Instructors. Proceedings of Workshop on Individual and Group Modeling Methods that Help Learners Understand Themselves, International Conference on Intelligent Tutoring Systems 2002, page 5-13.
5. Bull, S., and Pain, H. (1995) 'Did I say what I think I said and do you agree with me?': Inspecting and Questioning the Student Model. Proceedings of World Conference on Artificial Intelligence and Education (ACCE). Charlottesville, VA, pages 501-508.
6. Cheng R., Vassileva J. (2006) Design and Evaluation of an Adaptive Incentive Mechanism for Sustained Educational Online Communities. User Modeling and User-Adapted Interaction, special issue on User Modeling Supporting Collaboration and Online Communities, 16(3/4), 321-348.
7. DeSouza, C. S., Preece, J. (2004). A framework for analyzing and understanding online communities. Interacting with Computers, vol. 16, 579-610.
8. Denaux R., V. Dimitrova, and L. Aroyo. (2004) Interactive ontology-based user modeling for personalized learning content management. In AH 2004: Workshop Proceedings Part II, pages 338–347.
9. Denaux, R., Aroyo, L., and Dimitrova, V. (2005). An approach for ontology-based elicitation of user models to enable personalization on the semantic web. In Special interest Tracks and Posters of the 14th international Conference on World Wide Web (Chiba, Japan, May 10 - 14, 2005). WWW '05. ACM Press, New York, NY, 1170-1171. http://doi.acm.org/10.1145/1062745.1062923
10. Hansen c. and McCalla, G. (2003) Active Open Learner Modelling. Proceedings of AIED2003, July 20-24, 2003, Sydney, Australia.
11. Harth A., John G. Breslin, Ina O'Murchu, Stefan Decker (2005) Linking Semantically-Enabled Online Community The Semantic Web: Research and Applications. Second European Semantic Web Conference, ESWC 2005. Proceedings (Lecture Notes in Computer Science Vol. 3532), 2005, p 500-14
12. Kagal, L., Finin, T., and Joshi, A. (2001). Trust-Based Security in Pervasive Computing Environments. Computer 34, 12 (Dec. 2001), 154-157. http://dx.doi.org/10.1109/2.970591
13. Kass, R. and Finin, T. (1988) Modeling the user in natural language systems. Compute. Linguist. 14, 3 (Sep. 1988), 5-22.
14. Kimberly, R. Swinth, Farnham, D. Shelly and Davis, P. (2003). Sharing personal information in online community member profiles. 04-13-2005. http://research.microsoft.com/scg/#papers
15. McCalla G, Vassileva G., Greer J., Bull, S. (2000) Active Learner Modelling, in Gautier, Frasson & VanLehn (eds.) *Proceedings of ITS'2000*, Springer LNCS 1839, 53-62.
16. Merrells J. (2004) XACML: XML Access Control. http://www.idealliance.org/papers/ dx_xmle04/papers/04-01-04/04-01-04.pdf

17. Mohammed, I. and Dilts, D. M. 1994. Design for dynamic user-role-based security. Compute. Secure. 13, 9 (Oct. 1994), 661-671.
18. Niu X. , McCalla G. I., and J. Vassileva (2004) Purpose-based Expert Finding in a Portfolio Management System. *Computational Intelligence Journal*, Vol. 20, No. 4, 548-561.
19. OASIS (2005) eXtensible Access Control Markup Language 2 (XACML) Version 2.0 OASIS Standard, 1 Feb 2005 http://docs.oasis-open.org/xacml/2.0/access_control-xacml-2.0-core-spec-os.pdf visited: Oct 2006
20. Park, J. S., Sandhu, R., and Ahn, G. (2001) Role-based access control on the web. ACM Trans. Inf. Syst. Secur. 4, 1 (Feb. 2001), 37-71. http://doi.acm.org/10.1145/383775.383777
21. Sandhu, R. and Park, J. S. (1998) Decentralized user-role assignment for Web-based intranets. In Proceedings of the Third ACM Workshop on Role-Based Access Control Fairfax, Virginia, United States, October 1998. http://doi.acm.org/10.1145/286884.286887
22. Seth P. (2004). Sun's XACML Implementation. Programmer's guide for version 1.2. 2006 (Aril, 2006).
23. Sun L., Vassileva J. (2006) Social Visualization Encouraging Participation in Online Communities, In Groupware: Design, Implementation, and Use, Proceedings of CRIWG'2006, Springer LNCS 4154, 349-363.
24. Vassileva J., McCalla G., Greer J. (2003) Multi-Agent Multi-User Modeling, *User Modelling and User Adapted Interaction* **13**:(1), 2003, pp. 1-31.
25. Vassileva, J., Greer, J., McCalla, G. (1999) Openness and Disclosure in Multi-Agent Learner Models. Workshop on Open, Interactive, and Other Overt Approaches to Learner Modeling (Proceedings from 9th International Conference, AIED 1999), pages 43-49.