UNIVERSITY
OF TRENTO

DEPARTMENT OF INFORMATION AND COMMUNICATION TECHNOLOGY

38050 Povo – Trento (Italy), Via Sommarive 14
http://www.dit.unitn.it

A MODEL OF CULTURE FOR SOFTWARE

Aliaksandr Birukou and Enrico Blanzieri

April 2009

Technical Report # DISI-09-020

A model of culture for software

Aliaksandr Birukou, Enrico Blanzieri

DISI - University of Trento, Via Sommarive, 14, 38100 Trento, Italy
{birukou,blanzier}@disi.unitn.it

Abstract. With the advent of Web 2.0 one can speak about best practices and specific knowledge existing in a community of users of an application. This suggest that there is a kind of "culture" growing around such applications, and this culture is created by the communities. Moreover, cultures of communities "interact" leading to changes in applications, i.e. introducing tabbed browsing in NetCaptor browser in 1998 lead to its appearance in Opera in 2000, Firefox in 2001, and IE in 2006. In this paper we propose a model of culture in communities of Web 2.0 applications and other social software. Such model can be applied for characterization and comparison of culture(s) existing in communities for better understanding the use of systems by people. We provide an operative definition of culture of a community and illustrate by examples how our model of culture can be applied in different contexts in systems for communities in Web 2.0 and in software in general.
Keywords. Culture, Web 2.0, communities, Implicit Culture

1 Introduction

When software systems, especially Web 2.0 applications, are in use, communities of users grow around them and interact with systems in different ways: some use the most important features, others configure systems for a convenient use, or, even, engage in the development and shape systems for themselves as in open source software. Such interaction produces usage patterns, best practices [1], common artifacts, habits in using software. Some communities use forums, blogs, FAQ, etc. to exchange experience in using the software. All this suggest that there is a kind of "culture" growing around such applications, and this culture is, explicitly or not, created by the communities. Such culture evolves and sometimes leads to changes in the application itself, to better support the culture created within the community of users. For instance, when it became possible to reload open web pages after crashes in Firefox, many people started using that feature just to save their browsing sessions, killing the Firefox process before shutting down the computer. This lead to the development of specific plugins that allowed for restarting Firefox in a more intelligent manner. Finally, in Firefox 3.0 the developers introduced a feature called "Save and Quit" that allows one to close the application and restore the browsing session next time.

The aspect of sociality in communities of software users is represented in many ways. Social networks, social navigation [2], community search [3], user

profiles and stereotypes [4] are among examples. However, current representations of groups and communities fail to address the concept of culture in the usage of systems, focusing on other tasks, e.g. making explicit connections between users in social networks, or guiding people to relevant information in social navigation. Also, traditionally, personalization and recommendation systems approaches considered an individual user, while now there is a need for addressing groups [3]. The culture of a community of a system, once being discovered, can be separated from the community and used in specifications of next versions of software, developing test cases, helping newcomers to start using the system, describing best practices, comparing different communities. Moreover, some privacy problems are eliminated since culture is a product of a group of users.

In this paper we propose a model of culture in communities of Web 2.0 applications and social software, i.e. used by people, in general: Flickr, Delicious, BitTorrent, Firefox, CiteULike, Bibsonomy, OpenOffice, among others. The model can be applied for characterization and comparison of culture(s) existing in communities, making explicit needs of users, and automatically transforming culture in use-case scenarios and requirements. The model of culture can be also used to find deviations in the community culture and the culture "supported" by an application. By "supported" here we mean what applications can provide to its users. We provide an operative definition of culture of a community and illustrate how our model of culture can be applied in different contexts in systems for interaction between communities in Web 2.0 and in software in general.

The paper has the following structure: Section 2 provides a motivating example that shows one possible application of our notion of culture presented in Section 3. We resolve example in Section 4 and conclude in Section 5.

2 Motivating example

To help the reader to get into the context and to illustrate an example of the problem we are targeting at, we give a motivating example. Let us consider activities related to bibliography management in CiteULike.org, a free online service to organize someone's collection of academic papers. Users of CiteULike are mainly scientists and there are groups dedicated to specific interests. The site allows people to add papers in their personal collections or to the collections of the groups users belong to and to tag those papers. It is also possible to search for the papers using keywords or browse the papers for a specific tag.

Let us suppose that Michael, a user of CiteULike, has some papers about recommendation systems in his bibliography and has tagged them as shown in Table 1[1]. He discovers that there are groups on CiteULike and that there are at least three groups that seem relevant to his research interests: GroupA, GroupB, and GroupC. In the group bibliography each group has a list of papers tagged as shown in Table 1. Michael would like to join some group, but he does not have much time to read group feeds, so he would like to choose only one group. How

[1] Of course, we present a simplified example here, real users and groups on CiteULike have much more papers in their bibliographies

does he decide which group fits more with his interests? The bibliography of a group contains several hundred of items, looking through it will take some time.

Michael		
paperID	**paper**	**tags**
PolyLens	PolyLens: a recommender system for groups of users	recommendation, collaborative-filtering
TrustInRS	Trust in recommender systems	trust, recommendation
GroupLens	GroupLens: An Open Architecture for Collaborative Filtering of Netnews	collaborative-filtering, grouplens
RefWeb	Referral Web: Combining Social Networks and Collaborative Filtering	collaborative-filtering, trust
TrustCF	Trust-Aware Collaborative Filtering for Recommender Systems	trust, recommendation
Group A		
EComRec	E-Commerce Recommendation Applications	collaborative-filtering, e-commerce, recommender
TechLens	Enhancing digital libraries with TechLens+	recommender, academic-reference
GetToKnow	Getting to know you: learning new user preferences in recommender systems	collaborative-filtering, recommender
GroupLens	Group Lens: An open architecture for collaborative filtering of netnews	collaborative-filtering, recommender
PolyLens	PolyLens: a recommender system for groups of users	recommendation, collaborative-filtering
Group B		
TechLens	Enhancing Digital Libraries with TechLens+	collaborative-filtering, content-based-filtering, papers, recommender-systems
Citations	On the Recommending of Citations for Research Papers	citations, collaborative-filtering, personalization, recommender-systems
Scouts	Scouts, promoters, and connectors: The roles of ratings in nearest-neighbor collaborative filtering	recommender-systems, recommendation, collaborative-filtering
EComRec	E-Commerce Recommendation Applications	collaborative-filtering, e-commerce, recommender
ContRec	A content-collaborative recommender that exploits WordNet-based user profiles for neighborhood formation	collaborative-filtering, concept-extraction, concept-map, recommender
Group C		
GroupLens	Group Lens: An open architecture for collaborative filtering of netnews	collaborative-filtering, recommender, recommendation
VirtCom	Recommending and evaluating choices in a virtual community of use	collaborative-filtering, recommender
TagCF	Tag-aware recommender systems by fusion of collaborative filtering algorithms	tagging, recommender, collaborative-filtering
TrustInRS	Trust in recommender systems	trust, recommender, collaborative-filtering
RefWeb	Referral Web: Combining Social Networks and Collaborative Filtering	collaborative-filtering, social-network

Table 1. Users and groups on CiteULike.org

Let us assume that all tags are from the same taxonomy and there are no syntactical (e.g., tags `recommendation system`, `recommender systems`, `RS` are replaced with a single tag) and semantical (e.g., tags like `recommendation system`, `adaptive system` correspond to very same concepts in all bibliographies) inconsistencies in the names of papers and tags. Thus, we can represent a group or a user as a set of tags and a set of papers in their bibliography and calculate the degree of the fit between a user and a group as similarity between their sets of tags and papers. Moreover, we can see which papers are common for all three groups, creating for Michael a list of papers to read.

Further extending this example we might take into account not only artifacts such as papers or tags, but also behaviors of users, such as tagging some paper with a specific tag. For instance, using information about authors of the papers and citations, it is possible to consider behaviors such as self-citation and to see if there are communities whose members follow this practice more than an average author. Using information about the publication date and the date of posting the publication in someone's library it is possible to consider behaviors such as "tagging paper before its publication" and see which communities have the practice of dissemination of drafts of the papers.

3 A model of culture

In this section, we propose a model of culture that can be used for describing best practices, habits, usage patterns, rules, and artifacts of a community. Our goal here is to provide an operational model of culture that can be used for better understanding the needs of communities, finding similarities and differences in communities, and adapting software for communities.

We call a member of a community an *agent* and we assume that an agent can have different *traits*, which are "characteristics of human societies that are potentially transmitted by non-genetic means" [5]. Behaviors, beliefs, knowledge, norms, rules, values mentioned by many authors as elements of culture, in our formulation are just particular kinds of traits. Community artifacts, habits, etc. are also traits. The list of traits given here is not exhaustive, and whenever some item is seen as a potential culture element and it is not innate (the requirement of being transmitted by non-genetic means) it can be classified as a trait.

A community is represented as a set of agents. For each pair of agents we say that a trait is *shared* by them iff they both have such a trait. Given a set of agents and a set of traits we define the notions of weak sharing and strong sharing. A set of traits is *weakly shared* by a set of agents iff for each trait there exists a pair of agents that share this trait. A set of traits is *strongly shared* by a set of agents iff each trait is shared by all pairs of agents.

Given a set of more than one agents, we introduce the notion of culture of the set. A non-empty set of traits is a *culture* of the set of agents iff: 1) the set of traits is weakly shared by the set of agents; 2) each agent has at least one trait from the set of traits. If the set of traits is also strongly shared then it is a *culture in the strong sense*.

We define a *culture* of an individual as a set of traits the individual has.

Existing literature on culture in many cases emphasizes the aspect of transmission of traits, see, e.g., Sperber [6]. Some other authors suggest that culture is necessarily learned [7]. However, there are also approaches that consider only the sharing aspect [8]. We argue that for defining a culture of a community it is enough to consider just the aspect of sharing for the two following reasons: 1) in our domain, measuring transmission is hard if not impossible. For instance, it is probably hard for anyone to recall how the ability of copy-paste fragments of texts using CTRL+C and CTRL+V is acquired - if they learned it from

manuals, or from someone else; 2) since traits are transmitted by non-genetic means, they have been acquired during someone's life, so they were learned, or transmitted in another way, but not innate. For instance, it is hard to imagine someone who knows how to copy-paste text since their birth. Consideration of only shared traits also allows for faster computation of the culture of a group.

The model we present in this paper is only a part of larger ongoing work on formalizing the notion of culture. In that work we are focusing on the culture in general, not only in case of communities of users of information systems.

4 The example resolved

Now, let us consider the example from Section 2 and show how it can be addressed using the model of culture presented in Section 3.

In our formalism, the users and groups are agents that are represented as a set of traits, which are papers and tags. For each agent, its culture is the set of traits. Thus, `Michael.papers={PolyLens, TrustInRS, GroupLens, RefWeb, TrustCF}`, `Michael.tags={recommendation, collaborative-filtering, trust, grouplens}`, `GroupA.papers={EComRec, TechLens, GetToKnow, GroupLens, PolyLens}`, `GroupA.tags={collaborative-filtering, e-commerce, recommender, academic-reference,recommendation}`, `GroupB.papers={TechLens, Citations, Scouts, EComRec, ContRec}`, `GroupB.tags={collaborative-filtering, content-based-filtering, papers, recommender-systems, citations, personalization, recommendation, e-commerce, recommender, concept-extraction, concept-map}`, `GroupC.papers={GroupLens, VirtCom, TagCF, TrustInRS, RefWeb}`, `GroupC.tags={collaborative-filtering, recommender, recommendation, tagging, trust, social-network}`.

Let us define a similarity between the cultures of two individuals as the fraction of shared features. Since the number of distinct papers in Michael's and GroupA bibliographies is eight, the number of common papers is two, the number of distinct tags is seven and the number of common tags is two, the similarity between Michael and GroupA, sim(Michael,GroupA) is equal to $0.5 \cdot \frac{2}{8} + 0.5 \cdot \frac{2}{7} = 0.268$. The similarity between Michael and GroupB is $0.5 \cdot \frac{0}{10} + 0.5 \cdot \frac{2}{13} = 0.077$, while the similarity between Michael and GroupC is $0.5 \cdot \frac{3}{7} + 0.5 \cdot \frac{3}{7} = 0.429$. From this simple exercise we can conclude that Michael's research interests, as represented by his bibliography, are closer to GroupC. The program realizing such algorithm in real CiteULike.org settings, i.e. with hundreds of groups with thousands of papers, would solve the problem of choosing which community to join mentioned in Section 2.

Let us further illustrate how our formalism can be applied to these data. Let us consider each group as an agent and see which traits are shared in the set of agents `{GroupA, GroupB, GroupC}`. Papers `EComRec, TechLens, GroupLens` and tags `collaborative-filtering, e-commerce, recommender, recommendation` are weakly shared by the set and therefore are a culture of the set. Moreover, while there are no strongly shared papers, tags `collaborative-filtering, recommender, recommendation` are strongly shared and therefore are a strong culture of the set.

The cultures of the set we calculated might provide the papers that are widely known in different groups related to Michael's research interests and therefore could be interesting for him to look at. The tags could be used for his "pool of tags", i.e. next time he tags a paper he might consider one of those tags shared by the groups so as to facilitate the establishment of a common vocabulary, similarly to what Shaw et al. did [1].

5 Conclusion

In this paper we have presented a model of culture in software and illustrated one of its possible applications. We are currently conducting research on the formalization of our model of culture in a much broader context, not limited to software. However, even in software domain the application of our model to finding the closest group at CiteULike.org should be considered only as one possible variant. In future work we are going to elaborate more complex scenarios, including determining best default configuration for software using culture of the target community.

6 Acknowledgements

This research is partially supported by the EU COMPAS project and we would like to thank Paolo Giorgini and Fausto Giunchiglia, numerous discussions with whom lead to some ideas presented in this paper.

References

1. Shaw, M.L.G., Gaines, B.R.: Supporting modeling of the social practices of other users in internet communities. In: UM '99: Proceedings of the 7th International Conference on User Modeling. (1999)
2. Farzan, R., Brusilovsky, P.: Social navigation support through annotation-based group modeling. In: UM '05: Proceedings of the 10th International Conference on User Modeling. (463-472)
3. Jameson, A., Smyth, B.: Recommendation to groups. In Brusilovsky, P., Kobsa, A., Nejdl, W., eds.: The Adaptive Web: Methods and Strategies of Web Personalization. Springer (2007)
4. Lock, Z., Kudenko, D.: Interactions between stereotypes. In: AH'06: Proceedings of the 4th International Conference on Adaptive Hypermedia and Adaptive Web-Based Systems. (2006)
5. Mulder, M.B., Nunn, C.L., Towner, M.C.: Cultural macroevolution and the transmission of traits. Evolutionary Anthropology: Issues, News, and Reviews **15**(2) (2006) 52–64
6. Sperber, D.: Explaining Culture: A Naturalistic Approach. Blackwell Publishers (1996)
7. Axelrod, R.: The dissemination of culture: A model with local convergence and global polarization. The Journal of Conflict Resolution **41**(2) (1997) 203–226
8. Wyatt-Haines, R.: Align IT: Business Impact Through IT. Wiley (2007)