Need for Collective Decision When Divergent Thinking Arises in Collaborative Tasks of a Community of Practice

D. Assimakopoulos, M. Tzagarakis and J. Garofalakis

Abstract Based on previous work (Assimakopoulos et al. Proceedings of KICSS'2013 [5]) where we introduced HelpMe tool, a tool that automatically selects a group of people according to rules and metrics, we now introduce inconsistency between divergent ranking concerning answers during collaborative tasks among experts. In HelpMe tool, users collaboratively create a knowledge base about a subject and evaluate user opinions in order to achieve quality of knowledge. The basic assumption of the tool is that "Knowledge comes from experts." This is achieved by collective evaluation, through voting and discussion in every stage (Task) of the discussion (Activity). Inconsistency appears when a set of sentences cannot be true at the same time (Adrian et al. Proceedings of KICSS'2013 [3]). During collaborative tasks in Communities of Practice, inconsistency may inspire new associations and lead to more interesting solutions. However, there are cases such as medical, legal, etc. issues in which contradicting views are not helpful since a final decision has to be made. This paper is focused on such cases and examines the possible options and the methods that have to be implemented in order for a final decision to be made. When difference in the evaluation range is observed (divergent voting), the community should be informed in order to evaluate the existing answer with an "up vote" or a "down vote." No other option is available. In case of inconsistency, we introduce a loop procedure that informs experts in order

D. Assimakopoulos (✉) · J. Garofalakis
Computer Engineering and Informatics Department, University of Patras,
N. Kazantzaki, 26500 Rion, Greece
e-mail: asimakop@ceid.upatras.gr

J. Garofalakis
e-mail: garofala@cti.gr

D. Assimakopoulos · M. Tzagarakis · J. Garofalakis
CTI, Computer Technology Institute and Press "Diophantus", University of Patras,
N. Kazantzaki, 26500 Rion, Greece
e-mail: tzagara@upatras.gr

M. Tzagarakis
Department of Economics Research Academic Computer Technology Institute,
University of Patras, N. Kazantzaki, 26500 Rion, Greece

A.M.J. Skulimowski and J. Kacprzyk (eds.), *Knowledge, Information and Creativity Support Systems: Recent Trends, Advances and Solutions*, Advances in Intelligent Systems and Computing 364, DOI 10.1007/978-3-319-19090-7_25

to evaluate the task being observed as inconsistent. If inconsistency remains, or the number of evaluators is small, the process keeps going until it meets the criteria introduced in this paper. On the other hand, the system should be able to find, in its knowledge database, related conversations and examines the decisions made on related or same subject.

Keywords Collaboration · Web 2.0 · Communities of Practice (CoPs) · Argumentative collaboration · Social media · Recommender systems · Expertise finding · Inconsistency

1 Introduction

1.1 Communities of Practice and Argumentation

Communities of practice is a gathering place of people with common interests, is created. CoPs have become important places for people that seek and share experience and interests. CoPs are contribution places in which knowledge is offered to its members [19]. Basic characteristics of such a community are: (a) Interest field, (b) The community itself, and (c) Practice as mentioned in [5].

Argumentation is the study of how conclusions can be reached through logical reasoning. Argumentation focuses on a particular kind of semantic structure for organizing elements. Of central interest, therefore, is the Web 2.0 emphasis away from predefined information organizing schemes, toward self-organized, community indexing ("tagging") of elements, resulting in so-called "folksonomies" that can be rendered as tag clouds and other visualizations. Persuading "normal people" (in contrast to skilled information scientists or ontology engineers) to create structured, sometimes high-quality, metadata was previously thought impossible, and the success and limits of this approach are now the subject of a new research field that studies collaborative tagging patterns, e.g., [10].

1.2 Existing Problems in Communities of Practice

Basic problems in CoPs as described in [5] are: (a) Information overload and (b) data interconnection. All these above-mentioned problems are major to CoPs because intense effort is required in order to search and sort information and to find a user's opinion about a subject through time. In argumentative tools, some of the members evaluate questions and answers using Web 2.0 characteristics while others do not. For example, reddit's "down vote" is not the same as a "Dislike." Down vote shows that a specific post does not deserve to be in the front page. Tools with

Web 2.0 characteristics cannot get away from stack representation like forums we all do know. New ways of representation should be explored which will make it is easy for someone to: (a) visualize a conversation with the form of graph nodes or clouds or something else, (b) watch the conversation as it evolves or get involved in any stage, (c) be able to find appropriate users that would like to get involved in a specific conversation. A conversation at HelpMe tool is a collection of messages concerning the same tags. HelpMe tool chooses users who would like to participate in the conversation introducing two new metrics ULQI (user label query importance) and ULCI (user label communication importance) [5].

1.3 The Role of Tools in CoPs

Over the years, a variety of tools supporting argumentative collaboration have appeared; they usually facilitate argumentative discussions among members of a group and range from simple ones such as email, chat, and web-based forums to dialogue mapping and argumentative collaboration tools, reaching even into the realm of sophisticated conferencing and formal argumentation systems [16]. Tools in CoPs are very popular with everyone. These tools help people find users to help them with their problem. This is what is called expertise finders. Expertise finders, or expertise location engines, are systems that help find others with the appropriate expertise to answer a question. These systems have been explored in a series of studies, including Streeter and Lochbaum [15], Krulwich and Burkey [13], McDonald and Ackerman [1] as well as the studies in Ackerman et al. [2]. Newer systems, which use a social network to help find people, have also been explored, most notably in Yenta [9], ReferralWeb [12], and most recently commercial systems from Tacit and Microsoft. These systems attempt to leverage the social network within an organization or community to help find the appropriate others. In reality, relatively few people will claim themselves as an expert, but many people agree that they have some measure of expertise in some area. These systems allow everyone to contribute as they can. A person's expertise is usually described as a term vector and is used later for matching expertise queries using standard IR techniques. The result usually is a list of related people with no intrinsic ranking order or ranks derived from term frequencies. It may reflect whether a person knows about a topic, but it is difficult to distinguish that person's relative expertise levels [2].

Due to argumentative collaboration, why do we keep using tools? The answer is simple: because users' needs cannot be covered by the use of only one tool. For example, a search engine does not take into consideration the experience of the user who submits the question and as a result, the received answers do not match the user's profile. In conclusion, we can say that communities of practice use these tools in order to solve the problem collectively.

1.4 Existing Tools and Approaches in Communities of Practice

Based on previous work [5], five tools in the area of CoPs were examined: (a) CoPe_it!: an incremental formalization approach that facilitates the emergence of individual and loosely coupled resources into coherent knowledge structures and finally decisions. (b) Parmenides: is designed to support web-based policy consultation with the public and incorporates a formal model of argumentation [6]. (c) Cohere: a web tool for social bookmarking, idea linking, and argument visualization. It incorporates the Web 2.0 principles to create an environment called Cohere which aims to be semantically and technically open, provide an engaging user experience and social network, but provide enough structure to support argument analysis and visualization. Here, connections are created through ideas [8]. (d) Stackoverflow: as an alternative view of a forum. Its basic characteristic is questions and answers given under a wide variety of subjects to the community of software developers. Users can ask, answer, vote, and edit questions and answers like wiki or dig (social news web site) style. Stackoverflow users can earn reputation points and badges, depending on how other users judge them by the answers they have given. [17]. It is very well known that Stackoverflow is not just simply a revamp of a Q&A. It automatically suggests similar discussions and also, the community responses can be judged so that if a similar question was asked, the system can quickly get to the correct answer. (e) Reddit: is a social news and entertainment web site where registered users submit content in the form of either a link or a text ("self") post. Other users then vote the submission "up" or "down," which is used to rank the post and determine its position on the site's pages and front page. Content entries are organized by areas of interest called "subreddits." [18].

In [5] was introduced the HelpMe tool which is a Web 2.0 tool that supports: (a) Users who want to find answers to their problem by finding other users of the community. (b) Labels (or tags) to characterize a conversation: the Labels become the key for the HelpMe tool in order to automatically suggest similar discussions. By clicking a tag, all similar conversations are displayed. (c) Visualization through node graph, tag clouds, and statistic web pages. Node graphs can display the content tree of a conversation, user grades at any task, etc. (d) Recording of conversations for future retrieval. All conversations are stored in a SQL server database. (e) Evaluation system through metrics.

The above-mentioned tools were examined in the following Web 2.0 characteristics: (1) Rating issues: is the assessment of issues as listed by users of a community. (2) Rating of user answers: is the evaluation of the responses as rated by the users. (3) Rating of user evaluation: the answers provided by users are evaluated by others. So, as time goes by, a user rating index is created according to the issues which it tackles. (4) Karma or user reputation: some tools like Reddit encourage posts so that they do not show outside the community (i.e., self posts). Users are rated according to the self posts they make. Therefore, a karma rating is created for the Links and another karma for user responses. Similarly, users of

Table 1 Comparison between CoPs tools

	Rating issues	Rating of user answers	Rating of user evaluation	Karma or user reputation	Comments
CopeIt!	Yes	Yes	No	No	Yes
Parmenides	No	No	No	No	Yes
Cohere	No	No	No	No	Yes
Stackoverflow	Yes	Yes	Yes	Yes	Yes
Reddit	Yes	Yes	Yes	Yes	Yes
HelpMe	Yes	Yes	Yes	Yes	Yes

stackOverflow make reputation, when doing specific actions in the community. Such actions are considered the retagging of queries or close a query. (5) Comments: is the evaluation of the text answer given by users on a specific subject (Table 1).

2 HelpMe Community Tool

2.1 A Scenario of HelpMe Tool

A user seeks for a solution to his problem. He sends an email at helpme@westgate.gr and receives a link that leads him to a confirmation page. In this page he connects the initial question to an existing tag or set of tags, or creates a new tag (if no tag exists) to define the initial question. In this page, he also specifies the criteria or metrics that will be used by the system in order to find, in an automatic way, users capable of participating in a conversation whose goal is to find a solution to the initial question. These users will create an autonomous group in the community and may participate in as many conversations as the system decides. The conversation is represented as nodes to denote users and edges to denote an answer or like, dislike, best answer to an answer [5]. It should be mentioned that HelpMe tool is constantly developing and therefore it is expectable that sometimes it is not functional due to upgrading procedures. It is obvious it is used for research purposes and it is currently at a preliminary development stage. It is not a commercial product ready to use and it is subject to further research and development.

2.2 Start a New Conversation

The HelpMe tool receives a new query by the form of an email. Then, after user registration (if not registered), init user receives a link in his mailbox that sends him to the HelpMe confirmation page. In there, start user characterizes his question

using existing or new tags and specifies the way he wants HelpMe tool to find users, in order to answer his init question, so that a new conversation will start. At this stage, three main issues are considered:

- User initialization: the HelpMe tool is informed that a new user exists, so he/she may use HelpMe existing tags.
- Tag initialization: the HelpMe tool is informed that a new label exists, so it may be used by the existing users of the tool. In other words, all members of the community are initialized toward the new label.

User defines metrics association: Init user creates a relationship between ULQI and ULCI, in order for HelpMe tool to find users to start a conversation with him. ULQI is the average of all queries (questions) evaluated, concerning a user label. *Query evaluation concerns how community users rank the initial question.* ULCI is the average of all answers evaluated, concerning a user label. *Answer evaluation concerns how community users rank the user label after the first stage of initial question,* i.e., the conversation that takes place between community users. Finally, init user may create his own model of choosing users to participate in his initial question, using the score formula: ULQI * perc1 + ULCI * perc2. Score formula is the resulting label score from the previous formula. It should be mentioned that, perc1 and perc2 are defined by init user and the sum should be 1 (perc1 + perc2 = 1). For example, init user may give the formula 0.7 * ULQI + 0.3ULCI, which means that he wants the system to find users whose average of questions evaluated, concerning the label stated by the initial user, is 70 % more important in comparison with the average of all answers evaluated, concerning the same label, which is 30 % less important to the init user.

Figure on the right displays how the initial question is transformed into a conversation place (Fig. 1).

2.3 Messages Received

When init user chooses the criteria in the confirmation page, a HelpMe tool function finds users who would like to participate and give answers by doing the following:

- Creating a dynamic query in such a way that users fulfill the criteria of the score formula.

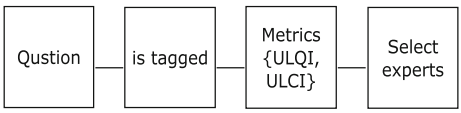

Fig. 1 HelpMe expertise finder model

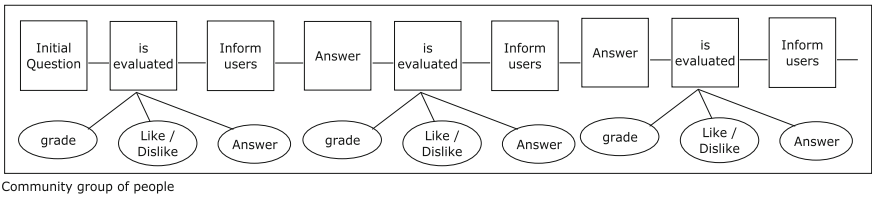

Fig. 2 Conversation model

- All selected users are recorded in database table. Every insert in that table fires a post insert trigger that sends a mail to the corresponding user.
- Then, the user node is created so that user may give an answer to a specific task of the conversation. This user node is stored in a database recursive structure that is explained in TheJit visualization section below.

Users chosen by HelpMe tool receive an email that asks for their participation in order to solve the problem (i.e., label defined by init user at the confirmation page). HelpMe tool records the users who receive the email and the corresponding link. The link creation is a very important procedure for HelpMe and gives the opportunity to any user who has received it, to participate at any time of the conversation. Important information that is carried through the link is the serial number of the conversation (i.e., Activity ID). Activity ID is created by an identity field in a database table, after HelpMe has received an email from init user. By knowing the Activity ID, HelpMe can display graphically all phases of the conversation (i.e., every phase is a TaskID) (Fig. 2).

A question is evaluated by the community users, see conversation model above, giving a grade or like/dislike or text answer, which are related to the question through the tag that characterizes the question. The first user of the community who responses to a question (box Answer of the above figure) leads the discussion to the next level (TaskID ← TaskID + 1). Any user is free to evaluate any answer at any level of the conversation. The conversation never ends, just loses interest at some time. Future work is to discover and exploit provenance of collective decisions of a conversation which took place at past time.

2.4 Visualization

In [5] HelpMe tool represents users as nodes and answers as edges. This information is stored in a database table where a recursive structure is created between father node (field name: id) and children node (filed name: children). In any step of the conversation (taskID), the text answered, the grade given, like/dislike/best answer, the average of the task, the average of the activity, and the number of conversation views are saved in an appropriate database structure. HelpMe tool uses

Fig. 3 TheJit visualization of user nodes: a user is displayed as a node with his id and email

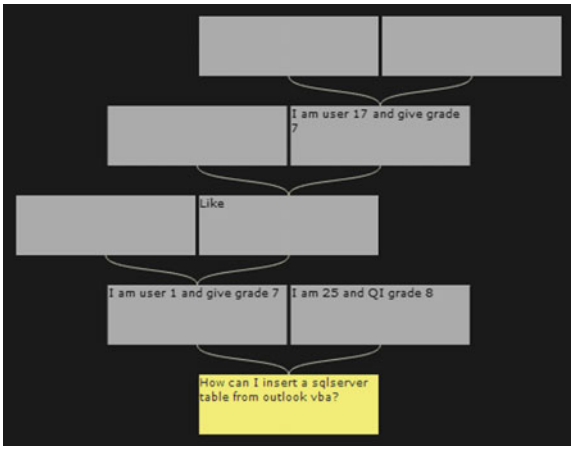

Fig. 4 TheJit Content visualization: displays the answers given at all tasks. Blank means no answer

Javascript Infovis Toolkit (TheJit) to represent users (nodes) and answers (edges). Graph visualization is more flexible because it provides many different views to the community users such as content, user info and statistics like traffic at a node, label ranking, user ranking, etc. (Figs. 3, 4 and 5).

Another type of visualization implemented in [5] was tag clouds [4, 7]. There are three types of clouds in the area of social software: (a) Tag clouds: are usually represented by html elements and may appear in alphabetical order, randomly, or according to weights. (b) Global tag clouds: where info frequencies appear with accumulative manner against elements and users (c) Sized categories: suggest the number of sub categories. HelpMe tool uses text clouds in order to represent the

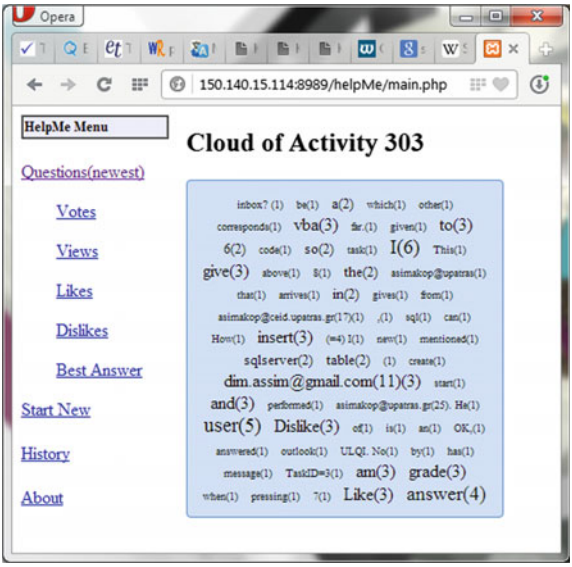

Fig. 5 HelpMe tool uses text clouds in order to represent the word frequency as a weighted list

Fig. 6 Statistics can display questions by label. By clicking each button, a specific page opens (i.e., answers button, leads to a page that displays all conversation answers, label button leads to relative conversations, user button leads to community user involvement, and so on.) By clicking ActvID, for example, 290, the node graph opens. By clicking cloud, the tag cloud of the conversation opens

word frequency as a weighted list. In order to visualize the tags, appropriate structures exist. Every tag is saved separately and ranked.

The third type of visualization implemented in [5] was statistics represented in web pages (Fig. 6).

2.5 Ranking Method

A user in HelpMe tool can give a grade from 1 to 10 to a previous answer and write some text explaining the reasons of his grade as well as to justify his own opinion about the subject. Alternatively, he may like, dislike the previous answer. A like corresponds to grade 8 while a dislike corresponds to −1. Alternatively, if and only if he is the start user, he may give a best answer which counts 10 points in HelpMe statistics. HelpMe tool gives users the opportunity to display community queries: (a) In a descending/ascending order according to date and time (b) According to: views, likes, dislikes, best answer, ULQI, ULCI, label ranking, user ranking, traffic information at any node (see figure: degrees of nodes). Each question takes the place of a row where is being displayed the ActvID, votes, answers, number of views, likes, dislikes, best answers, subject, label, and init user. Statistics can be expanded on a specific user to see user questions, answers, labels used, etc., or can be expanded on a specific label to see label ranking, etc.

2.6 HelpMe Tool Workflow Algorithm

When a user in the community receives a link, by clicking it HelpMe tool makes a series of checks:

1. It checks if he is the first who gives an answer. It is easy to find if the is a TaskID greater than the current one for this conversation. The conversation, as mentioned previously, is represented by the Activity ID.
2. If there no greater TaskID for that ActvID, then he is the first who answers the question. In that case, TaskID ⟵ TaskID + 1. The first user answers a previous question, assigns a new TaskID. On the same time he gives an answer or chooses between like, dislike, or best answer.
3. If next TaskID exists, then someone else answered first. In that case, the TaskID does not change. Then he also gives an answer or chooses between like, dislike, or best answer.

2.7 HelpMe Tool Workflow Procedure

A workflow procedure of a query stated by an initial user is presented in the figure at the bottom. The conversation with Activity ID 223 is first initiated by the start user (i.e., user1). He creates the first step of the conversation (TaskID = 001) on 07/07/2011 at 18:30. The first user who responds to that question is user3. User3 gives his own answer to the user1' label question, perhaps gives a good vote or likes the init question and then he takes the conversation to the next level

Step	ActvID	TaskID	Mail From - (out)	Mail To - (in)	Answered to	Date
1	223	001	User1	User3		07/07/2011 18:30
	223	001		User6		07/07/2011 18:30
	223	001		User7		07/07/2011 18:30
2	223	002	User3	User1	User1 (223-001-1)	07/07/2011 19:90
	223	002		User6		07/07/2011 19:90
	223	002		User7		07/07/2011 19:90
3	223	003	User7	User1	User1 (223-001-1)	08/07/2011 10:15
	223	003		User3		08/07/2011 10:15
	223	003		User6		08/07/2011 10:15
4	223	004	User6	User1	User7 (223-003-1)	08/07/2011 13:31
	223	004		User3		08/07/2011 13:31
	223	004		User7		08/07/2011 13:31
5	223	005	User3	User1	User6 (223-004-2)	08/07/2011 14:40
	223	005		User6		08/07/2011 14:40
	223	005		User7		08/07/2011 14:40

Fig. 7 Workflow tasks of a conversation

(TaskID = 002). The first user who answers at TaskID 002 is user7. User 7 has two options: to answer to the initial question of user1 or/and to answer to user2. By doing the first, the taskID will not change, while by doing the second TaskID will become 003. It should be mentioned that he is able to do both options. Every user is able to evaluate an answer at any level (taskID). A conversation never ends, it only loses its interest in time (Fig. 7).

In order for a conversation to get to the next level, someone needs to answer first to the question stated by the previous user. At any time, any other user may give his own answer but the taskID will not change. TaskID increases by one, only by the user who goes the conversation one step further (Figs. 8 and 9).

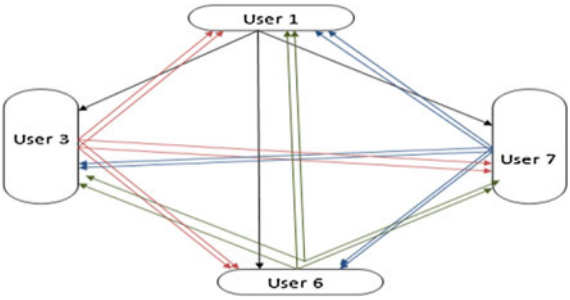

Fig. 8 Displays the traffic (answers) at any node. A node represents a user and directed edges represent traffic between users. So, at a glance, one may discover the active pairs of users

User	graph node degree in (mail TO)	graph node degree out (mail FROM)
User1	6	3
User3	5	6
User6	5	6
User7	5	6

Fig. 9 "Node in" is the number of times users replied to a user. For example, six users replied to user1. "Node out" is the number of times a user replied to a question. For example, user1 replied three times (to users: user3, user7, user6). These facts may help to build the profile of each user and are displayed in the statistics section

3 Recommender Systems

Recommender systems or recommendation systems are a subclass of information filtering system that seek to predict the "rating" or "preference" that user would give to an item or social element (e.g., people) they had not yet considered, using a model built from the characteristics of an item (content-based approaches) or the user's social environment (collaborative filtering approaches) [14]. According to Groza et al. [11], requirements of discovering and exploiting provenance are to provide a framework that consists of a model based on modularization, provenance information, identification and revision, support for domain knowledge, support for linguistic features, and complementing argumentation with orthogonal models.

Collaborative filtering methods are based on collecting and analyzing a large amount of information on users' behaviors, activities, or preferences and predicting what users will like based on their similarity to other users. Content-based filtering methods are based on information about and characteristics of the items that are going to be recommended. In other words, these algorithms try to recommend items that are similar to those that a user liked in the past. Approaches like collaborative filtering (e.g., user rating, rank labels, user opinions, keep record of views of a conversation, etc.) or content-based filtering (like, dislike buttons) have been partially implemented at HelpMe tool. HelpMe tool can recommend conversations through tags so far.

Currently, in HelpMe tool, we introduced the inconsistency that arises when difference in the evaluation range is observed (divergent voting) and propose a solution by sending invitations to experts until the inconsistency is eliminated, through score formula.

4 Methodology

HelpMe tool is an expertise finder that finds community users according to ULQI and ULCI metrics introduced here for the first time. The methodology behind the tool is expanding in five stages [5]: (a) study of forum visualization so far (b) user

needs about the content of a conversation in a forum (c) design of HelpMe tool (d) implementation of the tool (e) testing and evaluation. Unfortunately, due to the page limitation it is difficult to analyze the methodology in a separate chapter. It is a simple reputation system that computes and publishes reputation scores for a set of labels and users. The opinions are typically passed as ratings to the database and the tool uses a specific algorithm, using ULQI and ULCI to dynamically compute the reputation scores based on the received ratings. Two major things that make HelpMe tool to go beyond comparing to other tools is that (a) it uses content visualization of a conversation, rather than using stack representation like forums do and (b) uses two new metrics for expertise finding and computing the reputation scores based on the received ratings.

One can find several interpretations of inconsistency that are reflected in various definitions. Intuitively, inconsistency appears when a set of sentences (formulas, theorems, beliefs) cannot be true at the same time. When inconsistency arises, the system should automatically and appropriately react [3].

5 Handling Inconsistency

What goes beyond the previous work [5] is to introduce the inconsistency that arises when difference in the evaluation range is observed (divergent voting). In HelpMe tool, an inconsistency range may be defined by the community. For example, the community dealing with a subject can define the *inconsistency range* to vary between x grades from min to max. If inconsistency takes place at any task of the activity, the community should be informed in order to evaluate the existing answer with an "up vote" or "down vote". No other option is available. This will make the user to decide if he agrees with the specific opinion or not. In HelpMe tool we do not want to suppress inconsistency. We want each task of the activity to have the approval of the community. Since each task is approved, in every stage of the activity, the result will be an approved activity (conversation). The invitation must be given to as many experts to the subject as possible. In this way incorporating inconsistency into reasoning is achieved. In HelpMe tool this process is implemented by selecting distinct users of all related conversations that exist in the tool's database. In case of equality or small number of votes, the system should seek for more experts in the community by changing the score formula. *This process can be a loop that keeps inviting experts according to the criteria specified by the community.* Such criteria can be the number of evaluators and the inconsistency range. If the number of evaluators in a specific task is very small (parameter that can be defined by the community), then the criteria will not be met. If the inconsistency range will not drop to acceptable limits (as defined by the community) the criteria will also not be met.

In other words, if the criteria do not meet, the loop will go on until the score formula reaches the lowest point and the tool will invite all members of the community. A lowest point of the score formula may be: $0.5 * \text{ULQI} + 0.5 * \text{ULCI} > 0$.

Another approach to handle inconsistency is that the system should be able to find in its knowledge database related conversations, and propose the decisions made sometime in the past. In that case resources may at a later point be obsolete or characterized as unimportant by the group. The sheer diversity of the resource types requires from individual members of the group to engage into the process of information triage, i.e., sorting the available material, interact with the resources on the space in an attempt to interpret, and recast them as well as organize them into larger structures. Some resources may even have to be filtered out or marked as unimportant [16].

References

1. Ackerman, M.S., McDonald, D.W.: Answer Garden 2: merging organizational memory with collaborative help. In: Ackerman, M.S. (ed.) Proceedings of the ACM Conference on Computer Supported Cooperative Work, Computer Supported Cooperative Work, pp. 97–105. Boston, ACM Press, New York (1996)
2. Ackerman, M.S., Wulf, V., Pipek, V.: Sharing Expertise: Beyond Knowledge Management. MIT Press, Cambridge, MA (2002)
3. Adrian, W.T., Nalepa, J.G., Ligeza, A.: On potential usefulness of inconsistency in collaborative knowledge engineering. In: Skulimowski, A.M.J. (ed.) Proceedings of KICSS'2013. Progress & Business Publishers, Kraków, pp. 14–25 (2013)
4. Aouiche, K., Lemire, D., Godin, R.: Collaborative OLAP with Tag Clouds: Web 2.0 OLAP Formalism and Experimental Evaluation, WEBIST, Madeira, Portugal (2008)
5. Assimakopoulos, D., Tzagarakis, M., Garofalakis, J.: Communities of practice for developers: HelpMe tool. In: Andrzej M.J. Skulimowski (ed.), Proceedings of KICSS'2013. Progress & Business Publishers, Kraków, pp. 26–37 (2013)
6. Atkinson, K., Bench-Capon, T., McBurney, P.: PARMENIDES: facilitating deliberation in democracies. Artif. Intell. Law **14**(4), 261–275 (2006)
7. Bielenberg, K., Zacher, M.: Groups in social software: utilizing tagging to integrate individual contexts for social navigation. Masters Thesis submitted to the Program of Digital Media, Universität Bremen (2006)
8. Buckingham Shum, S.: Cohere: Towards Web 2.0 Argumentation. Knowledge Media Institute, The Open University, UK (2008)
9. Foner, L.N.: Yenta: a multi-agent, referral based matchmaking system. In: Johnson, W.L., Hayes-Roth, B. (eds.) Proceedings of the First International Conference on Autonomous Agents, pp. 301–307. ACM Press, New York (1997)
10. Golder, S., Huberman, B.A.: Usage patterns of collaborative tagging systems. J. Inf. Sci. **32**(2), 198–208 (2006)
11. Groza, T., Handschuh, S., Breslin, J.G.: Adding provenance and evolution information to modularized argumentation models. In: WI-IAT'08. IEEE/WIC/ACM International Conference, vol. 1 (2008)
12. Kautz, H., Selman, B., Shah, M.: Referral Web: combining social networks and collaborative filtering. Commun. ACM **40**(3), 63–65 (1997)
13. Krulwich, B., Burkey, C.: The ContactFinder agent: answering bulletin board questions with referrals. In: Proceedings of the 13th National Conference on Artificial Intelligence and Eighth Innovative Applications of Artificial Intelligence Conference, AAAI 96, Portland, Oregon, vol. 1. AAAI Press, Menlo Park, CA, pp. 10–15 (1996)

14. Ricci, F., Rokach, L., Shapira, B.: Introduction to recommender systems handbook. In: Ricci, F., Rokach, F., Shapira, L., Kantor, B. (eds.) Recommender Systems Handbook, pp. 1–35. Springer, Heidelberg (2011)
15. Streeter, L., Lochbaum, K.: Who knows: a system based on automatic representation of semantic structure. In: Proceedings of the Conference on Computer-Assisted Information Retrieval, RIAO'88 Program Conference, Cambridge, MA, 21–24 March, pp. 380–388. CID, Paris (1998)
16. Tzagarakis, M., Karousos, N., Gkotsis, G., Kallistros, V., Christodoulou, S., Mettouris, C., Kyriakou, P., Nousia, D.: From 'Collecting' to 'Deciding': facilitating the emergence of decisions in argumentative collaboration. Research Academic Computer Technology Institute, N. Kazantzaki, University of Patras, 26500 Rion, Greece (2010)
17. Wikipedia Article Stack Overflow. http://en.wikipedia.org/wiki/Stack_Overflow (2014)
18. Wikipedia Article Reddit. http://en.wikipedia.org/wiki/Reddit (2014)
19. Zhang, J., Ackerman, M., Adamic, L.: Expertise networks in online communities: structure and algorithms. In: Proceedings of the 16th International Conference on World Wide Web, International World Wide Web Conference, Banff, Canada, 8–12 May, pp. 221–230. ACM Press, New York (2007)