CUBA – Fostering Artificial Conviviality by a Web-based Feed Framework

Sascha Kaufmann, Christoph Schommer
Dept. of Computer Science and Communication – ILIAS Laboratory
University of Luxembourg, Luxembourg

Abstract

Conviviality is a concept of great depth that plays an important role in any social interaction. A convivial relation between individuals is one that allows the participating individuals to behave and interact with each other following a set of conventions that are shared, commonly agreed upon, or at least understood. This presupposes implicit or an explicit regulation mechanism based on consensus or social contracts and applies to the behaviours and interactions of participating individuals. With respect to an intelligent web-based system, an applicable social contribution is the give of assistance to other users in situations that are unclear and in guiding him to find the right decision whenever a conflict arises. Such a convivial social biotope deeply depends on both implicit and explicit co-operation and collaboration of natural users inside a community. Here, the individual conviviality may benefit from a wisdom of crowd, which fosters a dynamic understanding of the user's behaviour and a strong influence of an individual's well being to another person(s). The web-based system CUBA focus on such a behavioural analysis through profiling and demonstrates a convivial stay within a web-based feed system.

Keywords: Web Intelligence, Conviviality, Customer Profiling.

1. Introduction

We concern with the problem of simulating natural conviviality in an artificial, web-based environment. We believe that such a convivial environment directly influences the acceptance of the web-based system and, consequently, offers a competitive advantage.

However, an exact definition of what conviviality is does not really exist in the literature and there is neither a clear model nor a singular vision of how it can be realised. A usage of the word in a communication environment like the World-Wide Web is often understood as a layout problem.

Moreover, the relationship of conviviality to terms like amiability or comfort ability remains fluent: does conviviality refer to a place or a situation where someone is welcomed and/or feels well? Can conviviality be computed by algorithmic parameters, being adjusted and adapted? May external signal be considered and internal rules be activated such that we can obtain conviviality?

We believe that artificial conviviality is a concept of greater depth that is significant in a social interaction and, moreover, in social systems in general. Convivial relations between individuals allow them to behave and interact among each other following a set of conventions shared, commonly agreed upon or at least understood. This presupposes implicit or explicit mechanisms that are based on consensus or social contracts and applies to the behaviour and interactions of participating individuals [16]. Following [12], a natural conviviality distinguishes between a technological and a sociological meaning of the term. The technological meaning addresses the *quality pertaining to a software or hardware easy and pleasant to use and understand even for a beginner*. The sociological meaning stresses the importance of *positive relations between the people and the groups that form a society, with an emphasis on community life and equality rather than hierarchical functions*.

In Computer Science, a reliable and convincing definition of *artificial conviviality* has less been considered. As mentioned above, it is commonly understood as a user-friendly design or event, being often equated with GUIs (Graphical User Interfaces). *Artificial conviviality* also occurs in conjunction with digital cities and normative agents [4], design processes [13], or more generally in the context of sharing and enjoying a “good time” with other users. However, what a *good time* is remains unclear, since *good* is strictly subjective and, because of his previous experiences, almost each user has a different understanding of it. There are computer-related sources that may influence a “good time” in many ways, which makes it nearly impossible to introduce a single definition of *artificial conviviality* that covers all aspects.

An interesting concept, however, is proposed by [15],

where the term *conviviality* is associated to (software) tools as the result of a conscious user-centric decision: “I am aware that in English *convivial* now seeks the company of tipsy jolliness, which is distinct from that indicated by the Old English Dictionary and opposite to the austere meaning of modern *eutrapelia*, which I intend. By applying the term *convivial* to tools rather than to people, I hope to forestall confusion.” And, in fact, [15] intends to bring the technology to the level of “common” people making it accessible (and hence usable) to everybody. The idea is to enable (potentially all) users to use technique in a better/smoothen way. Instead of certain specifications on how convivial (software) tools should look like or should be used, Illich proposes some characteristics of convivial (software) tools, for example, no users must be excluded, no education should be necessary to use such a software tool, or no too complex features, even if they are implementable, must exist. Unfortunately, these guidelines have not been intended to the World Wide Web.

A promising approach is to foster the principles of *Wisdom of Crowd*. This principle is not new and there had been a lot of experiments and research work in the past. The term has been populated in [19]: here, it is argued that certain situations exist where a group of humans (= crowd) come up with a better solution (regarding a given problem) than the group’s smartest individual if a certain number of conditions exist:

- the *diversity of the existing points of view*
- a *decentralisation* and the *independence* of the participants
- a form of *aggregation*

The main idea behind the *diversity of points of view* is that knowledge – being unavailable for experts (the so called private or local knowledge) – must be collected when it is used for the final solution. To ensure that other group members do not influence such knowledge, there exist certain requirements of *decentralisation* and *independence*. At the end, an independent instance *aggregates* the different knowledge to *the wisdom of crowd*. This can be obtained in different ways, which means that there is no well-defined way of coming up with the perfect solution. Generally, there exist several types of problems where wisdom of crowd is very beneficial and applicable. For example, *cognition problems* require a particular solution. This can be a judgment of a market or optimisation problems. *Co-operation problems* involve trust between the members for their interaction. And *co-ordination problems* occurs when there is a need to arrange resources in relation with time.

In practice, it may happen that experts are smarter than the “crowd”. Generally, this may become crucial in case that an expert(s) continuously dominates such a group. An

aggregation of the wisdom may not be guaranteed in an acceptable way. However, there are also situations, where the crowds can fail with their solution, for example, when the composition of a group is too centralised, emotional, homogeneous, and/or imitative. Some of these reasons are only the opposite characteristics of the conditions discussed above (e. g. *low vs. high centralised, diversity vs. homogeneity or imitation*). In any case the most important thing is that the members of the group are independent. The goal is to let everybody in the group share their knowledge and experience; a little bit cheating won’t do any harm. The most dangerous situations are that groups are too small – group leaders can grow up and influence the opinion of other members – or that groups with nearly the same background knowledge exist (no diversity). In a small group, the avoidance of an imitative behaviour may lead to a cascading effect over the whole group. On the other hand, this does not heavily influence the result as long as it happens in a small environment.

2. An Artificial Framework

We follow the presented concepts in [15], [19] and focus on aspects that concern with *usability* and *content awareness*. We foster on the presentation of the right information at the right time in a direct way through the principle of personalisation and demonstrate this on web-based feeds: these correspond to a data format used for providing users with frequently updated content, thereby allowing users to subscribe to it. A collection of web feeds that is accessible in one spot is known as an aggregation feed. One advantage is that users do not have to send their address at subscription time and therefore not increasing their exposure to threats associated with spam or phishing emails. We believe that this successfully influences the level of conviviality during a user’s presence on the web page. The aim is to allow the user to use the environment in a free way and to recommend him content, which he might be interested in. This is accomplished by an analysis

- on how the user organises his content
- on how he acts during his stay at the web system
- to which extent the crowd may reasonable contribute.

Our understanding is that the combination of all these points supports the user towards a higher level of conviviality. We primarily take advantage of the non-obvious profile approach, which has been presented in [3]. The main idea is to define a set of topics Tp_i that describes the content of a web site in a proper way with

$$Topics = \{Tp_i\} \quad (1)$$

and $i = 1, \dots, n$. With respect to this, a topic Tp_i also corresponds to a certain area of interest. A weight indicates the relative importance of a topic in respect to the content, having a value between

$$0 \leq Tp_i(content_h) \leq 1 \quad (2)$$

which also corresponds to an interest between *not relevant* and *very relevant* ($i = 1, \dots, n$ and $h = 1, \dots, m$). Both topics and weights are then assigned to each web page $Page_j(Tp_i)$ that reflects its content in an accurate way with

$$0 \leq Page_j(Tp_i) \leq 1 \quad (3)$$

and $i = 1, \dots, n$ and $j = 1, \dots, m$. It is further possible to divide pages into different areas or zones $Zone_k(Tp_i)$, and to combine them with topics as well:

$$0 \leq Zone_k(Tp_i) \leq 1 \quad (4)$$

with $i = 1, \dots, n$ and $k = 1, \dots, m$. Alternatively, we may assign a topic-weight combination to such actions $Action_l(Tp_i)$ that let us draw conclusions about the user's interests and be supported by the web page:

$$0 \leq Action_l(Tp_i) \leq 1 \quad (5)$$

with $i = 1, \dots, n$ and $l = 1, \dots, m$. The leading concept is to concern with non-obvious user (interest) profiles called $N_U(Tp_i)$ for each user U with

$$0 \leq N_U(Tp_i) \leq 1 \quad (6)$$

on the basis of the user's U individual interests $IP_U(Tp_i)$ with

$$0 \leq IP_U(Tp_i) \leq 1 \quad (7)$$

and $i = 1, \dots, n$. Beside the duration of a session, all performed actions are taken into account. Because the system is able to operate on a fine granularity (i.e. parts of a web page are considered), the preparation of a personalised web site is generally very extensive.

Each feed k is then displayed in its own zone $Zone_k$ with assigned topic weights $Tp_i(feed_k)$, reflecting the content of the feed. The user may perform some simple actions, like subscribing and cancelling feeds, reading previews of articles, click links to reach the whole article or arrange zones at the web page.

We support web pages that consist of an individual layout of sets of zones, but do not assign static topics and values to such web pages. To calculate the $Page_j(Tp_i)$ we therefore come up with the following concept: in general, a page reflects the interests of an user, but we support the (re-) arrangement of feeds that will usually lead to a placement of interesting feeds at the top of the page. We then calculate $Page_j(Tp_i)$ by targeting all zones $Zone_k(Tp_i)$ of $Page_j$

Figure 1. Example of possible value assignment strategies for a layout with three rows: a) Dovetailing, b) Coating, and c) Waving). The boxes represent the feeds with their (relative) importance to the user. Arrows represent the way of calculating the corresponding importance.

weighting each zone in respect of the importance for the user. For this, a diverse number of strategies has been considered, where some of them are presented in Figure 1:

- **Dovetailing:** this strategy follows the psychological assumption that the more a user is interested in content the higher the assigned value will be.
- **Coating:** here we argue that the left-most/top-most content receives the highest weight again, but that – in contrast to *Dovetailing* – each following inverse coat, which is identified by the diagonal, is assigned to the same weight. We therefore assume that content with the same distance to the top content (left-most, top-most) should have the same weight.
- **Waving:** with this strategy, we perform a weighting following the radius around the top content.

We may then observe an *Interest Profile (IP)* as a way of modelling the level of the users' interests in topics Tp_i . Within a session, the visiting time and all actions on each page are recorded. When the user quits the system, the *Non-Obvious Profile (N)* algorithm automatically computes the interest profile of the user, i.e. the level of interest in each topic concerning the session. This is done in two steps, each considering a different approach on determine an interest profile: in the first step, the *Duration Profile DurP(i)* for each topic i is calculated. We hereby take into account the

duration of viewing $Page_j(Tp_i)$ in relation to the total time of viewing all pages. This part reflects the users' common interests, because it considers the page layout and how long the user has used a page:

$$DurP(i) = \frac{\sum_j (duration(Page_j) * Page_j(Tp_i))}{\sum_l duration(Page_l)} \quad (8)$$

Then, we determine the *Action Profile* $ActP(i)$ for all topics i . We include all actions involved with Tp_i and multiply this value with the number of topics Tp_i of the zone where they occurred. The result is also set in relation to the total sum of all actions that are involved with Tp_i during the whole session. This part takes the current interests of a user into account, by handle the actions that were performed in a zone during the session. It is as well possible to model different kind of actions (e.g. open an article may be a stronger indicator than reading the article's preview):

$$ActP(i) = \frac{\sum_k (\sum_l Action_l(Tp_i) * Zone_k(Tp_i))}{\sum_s Action_s(Tp_i)} \quad (9)$$

Finally, we combine both profiles, which is done by a weighted summarisation of action and duration:

$$N_{Session}(Tp_i) = \alpha * ActP(i) + \beta * DurP(i) \quad (10)$$

having $\alpha + \beta = 1$ and $i = 1, \dots, n$ in order to determine the $N_{Session}$ for this session. Such a strategy has been selected in order to involve the web site owner in the decision to determine the ratio between the duration profile and the action profile and how strong they influence $N_{Session}$. For example, a value of $\alpha = \beta = 0.5$ results in an equipollent influence of both profiles. The new N then replaces the old one (if existing), which is done with

$$N_{new}(Tp_i) = \frac{\sigma * N_{cur}(Tp_i) + \gamma * N_{Session}(Tp_i)}{\sigma + \gamma} \quad (11)$$

and with $i = 1, \dots, n$. We multiply the current non-obvious profile N_{cur} with the number of sessions σ and adding it to $N_{Session}$ by a factor γ . We finally divide it with $\sigma + \gamma$, where, γ signalises how strong the impact of $N_{Session}$ to the new profile N_{new} will probably be. We inform the user explicitly about his current interest profile. In case a user updates his interest profile IP_U we use

$$N_{U_{new}}(Tp_i) = \frac{N_{U_{cur}}(Tp_i) + IP_{U_{new}}(Tp_i)}{\kappa} \quad (12)$$

with $i = 1, \dots, n$ and $\kappa = 2$ to re-calibrate the user's current non-obvious profile for further usage.

Assume, a user starts a web-based session, an interest profile is created regarding actions and duration. In case

of a user request, the current interest profile is compared to (existing) similar profiles inside the community; an adequate content is then recommended. Currently, we use

$$d(U_j, U_k) = \left(\sum_{i=0}^n (N_{U_j}(Tp_i) - N_{U_k}(Tp_i))^2 \right)^{\frac{1}{2}} \quad (13)$$

to perform the matching. This allows us to identify users with similar interest profiles. As an alternative, we consider the *Pearson correlation*

$$r = \frac{1}{n-1} \sum_{i=0}^n \left(\frac{X_i - \bar{X}}{s_X} \right) \left(\frac{Y_i - \bar{Y}}{s_Y} \right) \quad (14)$$

since this supports the identification of users of similar differences in their chosen topics, respectively. As a consequence, this leads to different recommendations, in case that the users do not share subscribed feeds. And the acceptance or rejection to the given recommendations influences the user's profile again: if the user accepts the recommendation, the selected feeds become part of his web environment. As an result, the environment is considered as new; the topic weights are re-calculated, becoming part of the duration profile (8). Since there exists a profile of interest for each user, we identify users by the profile distance and take advantage of it if there is a request for recommendation. Then, we examine similar profiles and suggest channels or articles from similar users that are not in focus of the requester.

In case that each user's profile is updated, this may become very expensive. We therefore cluster the profiles of the community on a regular basis to decrease the complexity. And In case that a profile is modified, we assign the best fitting cluster to it. A second advantage is that a cluster comes up with typical interest profiles that describe the common user interests.

In order to measure the corresponding level of conviviality, our concept is to explicitly contact the user about the feelings and attitudes. Additionally, the individual behaviour is included. Furthermore, an implicit determination of the level of conviviality ends up with a diverse number of web-analytic concepts [18] as for example the duration within the web sites, the time to come back to the web environment, or the number of accepted recommendation. For the duration within the web environment, we assume that a "longer" stay indicates interest and is therefore a higher-level concerning conviviality (8). This practice is also applied to the duration of reading, for example the summary of an article. The question on how many actions the user performs during his visit can be answered quite easily: a high number of actions may indicate some kind of interest/satisfaction (9) (especially in combinations with the previous point), whereas a low number represents disinterest. Concerning the time of returning, a regular return may indicate a basic interest in the content provided by the web

Figure 2. CUBA (www.cuba.lu) fosters on Illich's concept of conviviality. (1) shows a preview of a message, (2) points to a currently closed channel. In (3), possible actions for a feed are presented, which are – from left to right – the updating of a channel (i.e. check for new messages), the opening and closing of a channel, and the cancellation of a channel subscription (i.e. removes a channel from Web site).

site. Finally, the number of accepted recommendations is considered as an indicator of conviviality because we derive the quality of the recommendation algorithm. It is also of interest, how long the visitor keeps subscribing the chosen feeds to get a feeling how good it covers his needs.

3. Implementation

The architecture of CUBA is composed of a Model-View-Controller (MVC) system that has been introduced by [1]. The advantage of this model is that the logic model can be separated from the preparation of the web site while the application stays maintainable. We can therefore separate both the computation of the different profiles and the aggregation of the wisdom of crowd (for a recommendation) from the preparation of the content for the user's individual web site.

From a programming point of view, CUBA is implemented by PHP and takes advantage of the Zend Framework [11]; it supports the MVC paradigm on PHP. The dynamic behaviour of CUBA is realized by applying the Ajax approach [2] and jQuery [6]. This allows us to support the user with a dynamic environment where he can drag and drop the feeds or read an article's preview. The later is using an asynchronous data pattern to serve the preview's data to the user's browser. On the other hand it allows to trace the user's actions and to store the result in a database without any interruption in the user's experience. This is a very important part, because being able to trace the user's actions and to influence the recommendations for the user are

one of the main parts of CUBA. The disadvantage is, however, that CUBA heavily depends on JavaScript to provide a good time: which means that in case the user switches it off, CUBA is no longer able to trace the actions. On the other hand, CUBA offers a comfortable environment which allows to individualize a Web site and to get results within a very short time. In other words, CUBA is trying to seduce the user to switch JavaScript on. On the background, CUBA uses ROME [8] and the ROME Fetcher module [9] to collect and prepare the feeds in a periodical interval. While ROME is responsible for parsing and converting the different feed formats into a common one, the ROME Fetcher module is loading the feeds in a intelligent way by caching them and using the "last modified" and "ETag" information to reducing bandwidth. For the storage of the profiles and feeds, CUBA maintains a PostgreSQL database [7].

The system CUBA is a first prototype on the basis of web feeds that simulates *artificial conviviality* in a web-based system. CUBA is easy to use to configure, it is continuously filled with the right information such that the user's conviviality can be influenced. We carry out the position of the instance that combines the local knowledge of the different members and provide a solution to the member's current problem. Individual knowledge is received by collecting information about the users following their interest profiles. Assuming that the user is experienced with the World-Wide Web, our approach bases on the following points. First, CUBA involves the subscribed channels, because a subscription indicates a strong interest in the topics covered by the

Figure 3. The Feed Concept in CUBA (right) and the Profile Visualisation/Feed subscription (left): once a user subscribes to a feed, it is directly added to the list of existing feeds. It is placed at the bottom of the page but can be manually moved to another location. Moving the mouse over a web link enables a preview window whereas a click on a web link opens a new window. The user profile is visualised and permanently available using pre-defined categories like for example *Business and Finance*, *Entertainment and Culture*, or *Sports*.

Figure 4. Preview (left) and Session Profile (right): the preview refers to recommended web feeds whereas the session profile corresponds to the user behaviour in this session.

channel. Second, CUBA deals with the duration of a session (session time) as an indication of how strong a user is interested in the content. Moreover, CUBA offers a self-configuration of the content. Third, CUBA traces the actions of the user: each action potentially gives a hint of the user's current interests. As for example, clicking on the link or printing an article are strong indications of the user's interest in the topic in question. Fourth, CUBA considers the locations of the different channels at the page: if a user is able to position the channels at the page, then he would do it in a way that reflects his current interests. This means that if a user is interested in the contents of a channel, then he properly puts this channel at a higher position at the page as a less interesting one. Fifth, CUBA considers the current status of the channel: a closed channel implies a certain but temporal interest to this channel. By collecting and clustering this kind of data and applying a clustering to it, CUBA comes up with a bunch of user profiles in the sense of a collaborative filtering and recommend interesting feeds or articles. Furthermore, CUBA determines the most popular feeds among similar users that are not subscribed by the visitor. This is an important topic as the users benefit from the knowledge of all other users. CUBA therefore supports a "get to know": this is an essential aspect of the traditional definition of conviviality of having a good time together.

In general, the visitor may subscribe, re-subscribe, and arrange feeds on the personal page. He is allowed to update, open and close them, to read a preview of the selected con-

Figure 5. Example of modifying an interest profile, where the user decides its interest within 6 categories ranging from 0 (no interest) to 5 (very high interest).

tent and to access it directly. While tracing these actions, a user performs on feeds and headlines CUBA builds an interest profile belonging to the user. This is achieved as follows: a cancellation of a feed is understood as a non-interest in its related topics, whereas other actions like reading a preview or refreshing a channel are acceptable indicators for an interest in a channel and its topics. In addition, a closed channel is understood as a “basic, but no current interests in the feed”. Another indication might be the recording of the time with respect to the articles’ previews. Even the position of the feed can be taken into account, where a “top-feed” (a feed that is at the top position) may be more important to the visitor than others. This is, because the user can read it immediately and without scrolling, even after the personal page is accessed. In Figure 2, a cutout with the areas of the subscribed feeds is presented.

Figure 6. Example of the graphical representation (right) of a computed Non-Obvious Profile: each node $T_1, \dots, T_{11}$ represents an own (pre-defined) category. The topic values vary between 0 (non interests/center) and 1 (very high interest/outer edge of the wheel). For example, $T_5 = 0$ while $T_4 = T_9 = 1$.

Following [13], it is important to inform the user why CUBA performs an action in order to achieve conviviality. This is to avoid the user's impression to be controlled by CUBA. This is respected by the allowance of examining the current non-obvious profile and of modifying the interest profile (Figure 6). As a future concern, CUBA will present information about the community to the user as well.

Figure 7 demonstrates the generation of clusters. CUBA merges the users U_1 , U_2 , and U_4 to a cluster because of their strong interest in topics T_3 , T_4 , T_7 and T_9 . Their interest in T_1 , T_2 , T_5 and T_{10} , however, is low. The User U_3 , however, is assigned to another cluster because of different interests.

Figure 7. Example of graphical representations of user profiles. CUBA merges users U_1 , U_2 and U_4 into one cluster while U_3 belongs to another cluster.

4. Discussion

In this work, we have concerned with *artificial conviviality* and have presented a software solution called CUBA that focuses on the principles of *wisdom of crowd* and that faces up to the intellectual concern by supporting web-feeds. We have taken advantage of the concept of *Wisdom of Crowds* and the characterisation of web users by non-obvious profiles. CUBA has been evaluated by a diverse number of test sessions that have been focused on the usability to raise the further acceptance of the visitors. We have got valuable comments ranging from enthusiastic feedback to important clues and from technical problems to psychological tips.

Some problems, however, remain open. For example, if we assume that the Wisdom of Crowd is a group of k members, what happens if the group's size changes? Up to which point is the crowd still a crowd and when do the decisions of the *wisdom of crowd* come into affect? Secondly, can we identify the “weight” of the individuals inside the crowd? It is obviously that not all group members share the same weight and that some kind of roles between the members potentially exists as for example *leaders*, *followers*, or *disturbers*. Third, the question of how reliable the state of the individuals is another open question. Can we really assume that users live in a perfect/optimal world where the crowd is perfect or does a noise inside the crowd's wisdom significantly increase too strongly?

Acknowledgement

The current work is funded by the Fonds National de la Recherche (FNR) and is conducted at the University of Luxembourg. The interested reader may register to CUBA by <http://www.cuba.lu/auth/register/>.

References

- [1] T. M. H. Reenskaug. MVC XEROX PARC 1978-79. <http://heim.ifi.uio.no/~trygver/themes/mvc/mvc-index.html>, 1979.
- [2] J. J. Garrett. Ajax: A New Approach to Web Applications. <http://www.adaptivepath.com/ideas/essays/archives/000385.php>, 2005.
- [3] N. Hoebel, S. Kaufmann, K. Tolle, and R. Zicari. The design of gugubarra 2.0: A tool for building and managing profiles of web users. In *IEEE/WIC/ACM International Conference on Web Intelligence*. IEEE, 2006.
- [4] P. Caire. A normative multi-agent systems approach to the use of conviviality for digital cities. In *International Conference and Research Center for Computer Science, Schloss Dagstuhl, Wadern, Germany*, 2007.
- [5] S. Kaufmann and C. Schommer. e-conviviality in web systems by the wisdom of crowds. In *Proceedings of the 4th ICITST09 - International Conference for Internet Technology and Secured Transactions*, 2009.
- [6] The jQuery Project. jquery 1.32. <http://jquery.com>, 2009.
- [7] PostgreSQL Global Development Group. Postgresql database 8.4.1. <http://www.postgresql.org/>, 2009.
- [8] A. Abdelnur, P. Chanezon, and E. Chien. Project rome 1.0. <https://rome.dev.java.net/>, 2009.
- [9] Unknown. Rome fetcher 1.0. <http://wiki.java.net/bin/view/Javawsxml/RomeFetcher>, 2009.
- [10] S. Kaufmann and C. Schommer. Towards e-conviviality in web-based systems by considering the wisdom of crowds. In *2nd Conference on Agents and Artificial Intelligence (ICAART 2010), Valencia, Spain.*, 2010.
- [11] Zend Technologies Ltd. Zendframework 1.10. <http://www.zendframework.com>, 2010.
- [12] E. Britannica. *Encyclopaedia Britannica – Miriam Webster’s Dictionary and Thesaurus*. Encyclopaedia Britannica, 2008.
- [13] G. Fischer and A. Lemke. *Constrained Design Processes: Steps Towards Convivial Computing*. In R. Guindon, ed., ‘Cognitive Science and its Application for Human-Computer Interaction’, Lawrence Erlbaum Associates, Inc., Hillsdale, NJ, USA, pp. 1-58., 1988.
- [14] J. J. Garrett. *The Elements of User Experience – User-Centered Design for the Web*. New Riders (2003). Berkeley, CA, USA., 2003.
- [15] I. Illich. *Tools for Conviviality*. Harper & Row, New York, USA., 1980.
- [16] J. J. Macionis. *Sociology*. Pearson Education, UpperSaddleRiver, New Jersey, USA, 2008.
- [17] C. Shaw and J. Ivens. *Building great customer experiences*. Palgrave MacMillan. New York, NY, USA., 2002.
- [18] J. Sterne. *Web Metrics – Proven Methods for Measuring Web Site Success*. Wiley Publishing, New York, NY, USA., 2002.
- [19] J. Surowiecki. *The Wisdom of Crowds – Why the many are smarter than the few*. Abacus. London, UK., 2005.