# TAGZILLA: Tag-Based File Storage and Retrieval

Vikram Nair, Vijayanand Banahatti, and Niranjan Pedanekar

Systems Research Lab, Tata Research Development & Design Centre,
Tata Consultancy Services, 54 B Hadapsar Industrial Estate, Pune 411013, India
{vikram.nair,vijayanand.banahatti,n.pedanekar}@tcs.com

**Abstract.** Users have to rely on memory for storing or retrieving data in Hierarchical Folder Organization (HFO) such as the Microsoft Windows Explorer for managing their information. We propose 20 Interface Design Objectives (IDOs) for Personal Information Management (PIM) interfaces. We find IDOs of HFO that need the most improvement using a qualitative survey of 66 users on importance and satisfaction scales. We present an alternate tag-based interface called TAGZILLA based on the concept of the 'Stream of Consciousness'. TAGZILLA provides users with an interface to create tags for storing files and retrieve files based on tags. It also suggests tags during storage and retrieval. We report an increase in satisfaction for all IDOs using a return survey with 20 participants who used TAGZILLA. We also present a preliminary quantitative experimental comparison of TAGZILLA with the Windows Explorer interface for the IDOs needing most improvement.

**Keywords:** Microsoft Windows Explorer, Personal Information Management, Tagging, Human Computer Interfaces, Hierarchical Folder Organization.

## 1 Introduction

A vast majority of users use the tree-based Hierarchical Folder Organization (HFO) such as the Microsoft Windows Explorer to organize information on computers. This would seem natural in the last century when users were used to physically filing paper documents in folders. But even in today's connected world with a much wider variety of information, 56-68% of file retrieval is still done using folder navigation despite advances in technologies such as desktop search [1].

Information can belong to multiple folders as humans naturally associate information with multiple concepts. But HFO has conditioned users to think of information in terms of hierarchies and not in terms of correlation with other concepts. They have made users think of 'where' to look rather than 'what' to look for [2].

Search using location is perhaps natural when information is limited. Let us consider the real world task of storing and retrieving a blue shirt. One would simply have to find a good place to store it, and then remember the location where the shirt was kept in order to retrieve it. The problem is that this works for a limited number of shirts, but would be taxing when the number of shirts and storage locations increase.

S. Yamamoto (Ed.): HIMI/HCII 2013, Part I, LNCS 8016, pp. 505–514, 2013.

Furthermore, a user could describe a blue shirt uniquely by a stream of concepts that appears in her mind such as ‘worn at graduation’ or ‘gifted on Valentine’s Day’ in order to differentiate it from or associate it with other shirts or things. But, the HFO forces the user to take a decision for storing the shirt – whether to choose the ‘Blue’ folder inside the ‘Shirts’ folder OR to make a new folder called ‘Graduation’ inside it OR to choose the ‘Graduation stuff’ folder, even if all are valid storage places. At the time of retrieval of the blue shirt, the user would either have to recall where exactly the shirt was kept or would have to explore several possible paths using partial recall and even brute force.

This need for relying on memory of a path rather than how the user recognizes the object is a major drawback of the HFO. Bloehdorn and Völkel [3] summarize the deficiencies of the HFO as: need to know exact file location, inability to represent orthogonal information as folders, dependence on order of directories, absence of query refinement and lack of navigation aids. Given such limitations, we believe that there is a need to find alternative interfaces which depend on remembering associations rather than recalling its exact location and traversal path.

In this paper, we present TAGZILLA, a tag-based approach to Personal Information Management (PIM) that we have built to complement the way humans think. Our specific contributions are:

1. We present the TAGZILLA interface, which is based on the concept of the ‘Stream of Consciousness’.
2. We define 20 specific interface design objectives for a PIM interface.
3. We find the objectives which need improvement in an HFO interface, specifically Microsoft Windows Explorer, using an importance-satisfaction survey of users.
4. We report an increase in user satisfaction for these objectives in a return survey of users test driving TAGZILLA.
5. We present quantitative results from preliminary experiments comparing TAGZILLA with a traditional HFO, viz. the Microsoft Windows Explorer.

## 2 What Needs to Improve in PIM Interfaces

Voit et al. [4] suggest broad requirements for PIM interfaces such as compatibility with current user habits, minimal interference and support for multiple contexts under which user plans to retrieve files. Before we set out designing a new interface, we wanted to get a better understanding of which specific PIM functionality needed improvement for a traditional HFO. We divided the PIM functionality in two broad categories, viz. ‘store’ and ‘retrieve’. ‘Store’ had further subcategories, viz. create file, categorize / group files, copy file and delete file. ‘Retrieve’ also had further subcategories, viz. search file, find related / similar files, compare files, view file contents, and filter / sort files. Based on user operations normally carried out in each of the above subcategories, we came up with 20 Interface Design Objectives (IDOs), e.g. ‘reduce navigation path to find a file’, ‘reduce time required to find a file’, ‘increase likelihood of finding related files’.

We followed the Importance-Satisfaction (I-S) model [5] for identifying the functionality needing improvement. An I-S model finds the product or service attributes which need the most improvement by rating each attribute in terms of its perceived importance and the perceived level of satisfaction with it [5-7]. The 'to-be-improved' attributes lie in the high importance and low satisfaction quadrant of an importance-satisfaction graph. We conducted an I-S survey on 66 users consisting of scientists, administrative staff, IT professionals and students. They rated the 20 IDOs on their importance and satisfaction with an HFO interface, viz. Microsoft Windows Explorer. Both scales were 5-point Likert scales ('not at all important' to 'extremely important' and 'not at all satisfactory' to 'extremely satisfactory').

We found the most important 'to-be-improved' IDOs in the high importance and low satisfaction region, viz. 1. Reduce navigation path to find desired location, 2. Reduce time required to find a file, 3. Increase likelihood of finding the most appropriate file, 4. Reduce time required to find related or similar files, and 5. Make PIM operations more natural. The survey results are shown in Fig. 3 in comparison with a return survey using the TAGZILLA interface (See Section 5).

## 3 The Key Idea Behind TAGZILLA

Consider the shirt scenario that we mentioned in Section 1. Now imagine a hypothetical assistant. You tell him the things that come to your mind when you see the shirt. For example, you tell him that 'it was bought at Acme' and 'worn at graduation'. In addition, the assistant himself notices obvious things about the shirt such as color and brand. While retrieving the shirt, you may recall 'the Acme shirt' and tell the assistant about it. If there are two shirts from Acme, the assistant narrows down your quest by suggesting 'graduation' for this shirt, and 'Valentine's day' for another one. If you just remember 'blue', the assistant suggests further options which might include 'graduation'. This helps you quickly find the shirt by remembering things that come to your mind when you think of that shirt.

We present a PIM interface called TAGZILLA that simulates the above process. It relies on capturing the concepts that come to user's mind on seeing a file while storing, and on thinking of a file while retrieving. The Concise Oxford English Dictionary defines a similar concept as the 'Stream of Consciousness' (SOC) as "*a person's thoughts and conscious reactions to events, perceived as a continuous flow*" [8]. In our context, we define it as "*a series of concepts that come to a user's mind one after the other on seeing or thinking of an object*". We use this definition as a basis for our interface design. We capture the SOC as a series of tags as tagging is emerging as a promising alternative to HFOs [9].

In TAGZILLA, the user utilizes her SOC to create tags for a file she encounters. At the time of retrieval, the user remembers a file by tags associated with it and is able to generate the SOC, albeit not in the same order or number. TAGZILLA provides help during storage and retrieval by suggesting tags of files associated with the tag typed in, and narrowing down the search space.

## 4 User Interface Design

We developed the prototype of TAGZILLA as a web interface that opens in a browser on Windows XP and 7 machines. A Windows service monitors addition of a new file or changes to a file. Before we describe the interface flow, we define the types of tagging that TAGZILLA provides:

**Automated Tags:** A set of pre-defined tags are automatically assigned to a file based on parameters such as extension, type and date. These are extracted from the ID3 metadata associated with the file. If a file ‘Vijay.jpeg’ is introduced to TAGZILLA, it automatically tags it with the tags such as ‘Pictures’, ‘jpeg’ and ‘23/01/2012’ based on the ID3 metadata.

**Personalized Tags:** When a user uses her SOC for a file, she comes up with unique tags that might help her identify the file in future. Also, TAGZILLA suggests questions such as ‘Who gave the file(s)?’, ‘On what occasion?’, ‘Where will you use it?’ to aid the user’s SOC. For the example mentioned above, let us assume that Vijay is wearing a blue shirt in the picture ‘Vijay.jpeg’ at the time of his graduation. The above questions might prompt him to create personalized tags such as ‘Me’, ‘Blue Shirt’ and ‘Graduation’. He could also add other tags such as ‘happy’ that come to his mind on seeing this picture.

**Tag Suggestions:** TAGZILLA also aids the SOC by providing tag suggestions during file storage and retrieval. It does so by finding files with the tag given by the user, and suggesting the tags associated with these files as ‘related tags’. The related tags are extracted using their proximity in the tag file graph and other factors such as most recent usage and file extensions. For the example mentioned in Fig. 1, if the user thinks of the tag ‘ACME’, TAGZILLA aids the SOC by suggesting ‘Development’, ‘Testing’, and ‘Research’ based on the tag file graph.

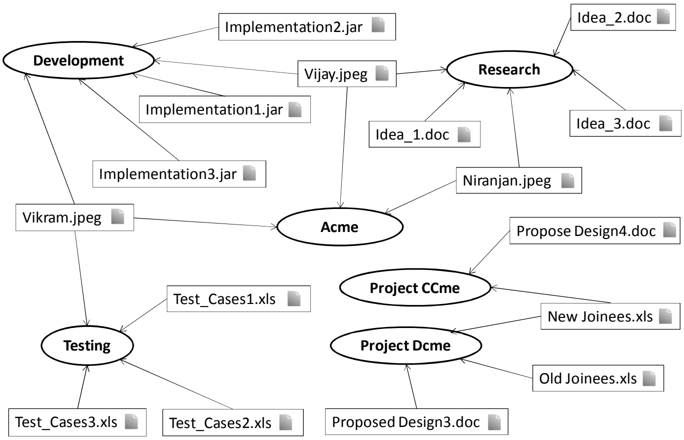
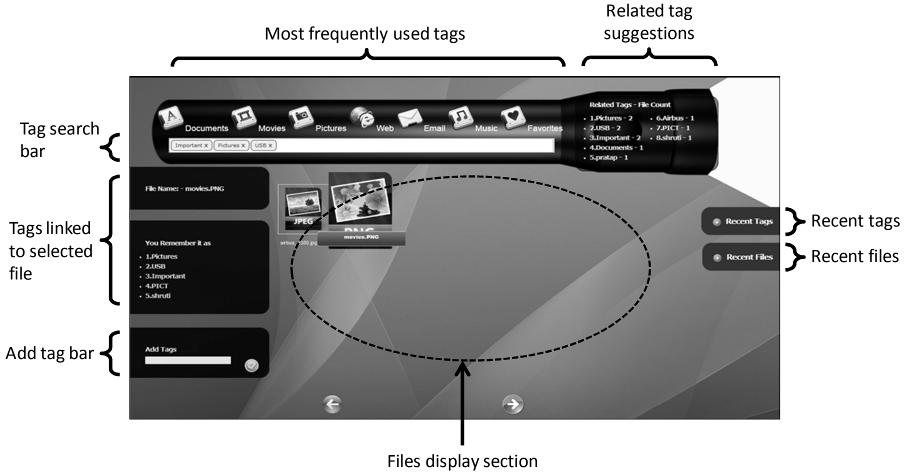


**Fig. 1.** An example of a tag-file graph in TAGZILLA

The TAGZILLA interface consists of two screens: one for storing (or tagging) files, and one for retrieval. The interface flow is as follows:

### 4.1 File Storage

1. When a file(s) is copied or created into the file system or in a central folder, a Microsoft Windows service registers a change in the file system.
2. At this point, TAGZILLA creates automatic tags for the file(s).
3. TAGZILLA interface prompts the user to create personalized tags. It also provides an autocomplete feature to quickly recreate existing tags.
4. TAGZILLA also gives tag suggestions aiding the user's SOC.
5. Personalized tag(s) created by the user, selected from suggestions, and automated tags created by TAGZILLA are stored against the file in the TAGZILLA database.
6. User can also assign tags to untagged files listed under the 'Untagged files'.



Related Tags - File Count

- 1.Pictures - 2
- 2.USB - 2
- 3.Important - 2
- 4.Documents - 1
- 5.pratap - 1
- 6.Airbus - 1
- 7.PICT - 1
- 8.shruti - 1

File Name: - movies.PNG

You Remember it as

- 1.Pictures
- 2.USB
- 3.Important
- 4.PICT
- 5.shruti

**Fig. 2.** File Retrieval Page

### 4.2 File Retrieval

1. At the time of retrieval, the user uses the retrieval screen shown in Fig. 2.
2. The frequently used tags are provided at the top for quicker retrieval.
3. The user enters a tag that comes to her mind about a file(s) she wants to retrieve in the search bar. An autocomplete feature is provided if the tag already exists to avoid minor variations in tags such as 'Shirt' and 'Shirts'.
4. TAGZILLA displays thumbnails of all the files tagged with this tag in the display section below the search bar.
5. Based on the current tag, TAGZILLA suggests further tags as 'related tags' on the right hand of the search bar to aid user's SOC.
6. As the user adds another tag in the search bar, the search results in the display section are updated by showing the files also having this tag.

7. A tag in the search bar can be deleted by clicking a cross provided on the tag. The search results are updated accordingly.
8. On hovering over the file thumbnail, one can see the tags associated with the file on the left hand side of the display section. One can also add a new tag to a file using the ‘Add tag’ box on the bottom left.
9. The ‘Recent tags’ and ‘Recent files’ shortcuts on the right hand side allow quick access to recent activity.
10. The file can be launched by double clicking the thumbnail within the browser.

## 5 Results and Discussion

In order to evaluate the effectiveness of TAGZILLA, we first conducted an I-S survey-based evaluation similar to the one mentioned in section 2. We allowed 20 users to interact with TAGZILLA for a period of 30 minutes each. Users rated the IDOs on 5-point scales for importance and satisfaction. We found that while the IDOs largely retained their importance, the satisfaction levels increased by 1 point on an average as compared to the earlier survey (See Fig. 3). We found that most IDOs shifted from the ‘To-be-improved’ quadrant to the ‘Excellent’ quadrant, especially the ones that needed the most improvement.

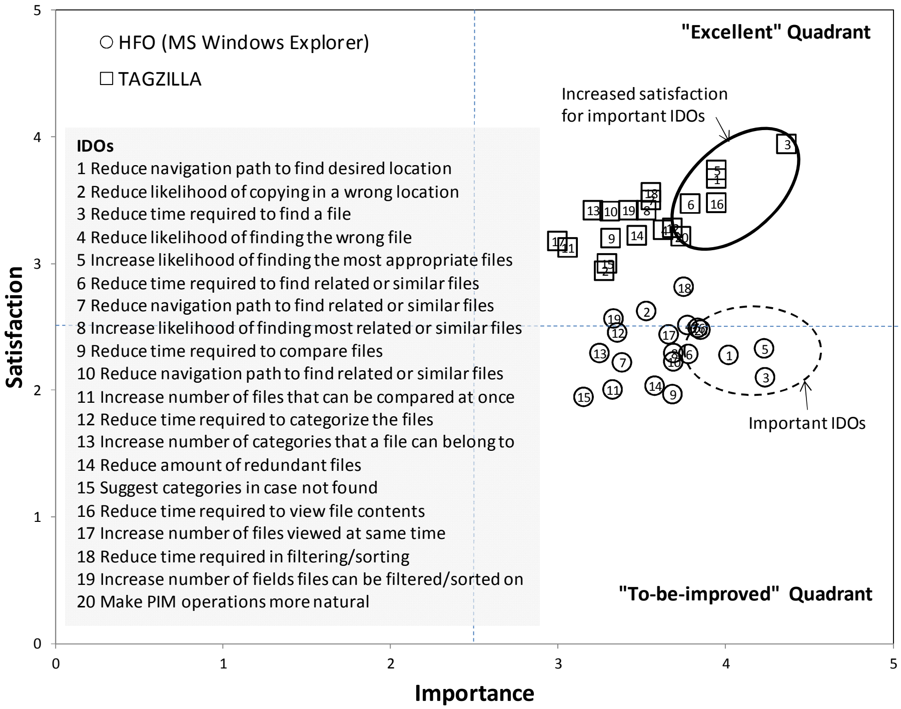


**Fig. 3.** I-S survey results for HFO and TAGZILLA interfaces

We then conducted a preliminary controlled experiment on 18 test subjects to compare TAGZILLA to the Windows Explorer interface. The test subjects were given 30 files along with their descriptions. Group A consisted of 9 test subjects. Each subject was asked to form a directory structure to store these files. Group B consisted of 9 test subjects. Each subject was asked to create tags for each file. After a week, the test subjects were asked to retrieve files for 10 practical scenarios (e.g. 'My home interior decoration plan need to be shared with my wife', 'Need to apply for home loan, so need the latest soft copy of my pay slip', 'I need to refer some old project architectures document as reference'). Group A used the Microsoft Windows Explorer for this task, while Group B used TAGZILLA. We recorded the screen activity as each test subject performed the given tasks. We then analyzed the recordings, and measured the number of clicks and the time required to complete each task. We present a discussion on the results for each important IDO:

1. **Reduction in navigation path to find desired location:** For TAGZILLA, we counted a text entry, a tag deletion and a tag selection as a click. We found that the number of clicks required for retrieving files reduced by an average of 26% with TAGZILLA. We attribute the larger number of clicks to the brute force directory traversal behavior shown by the HFO users, where a user selects a folder, browses content and repeats the process if the file is not found [10].
2. **Reduction in time required to find a file:** We found that the time required to retrieve a file increased by an average of 27% with TAGZILLA. We attribute this to the inclination of TAGZILLA subjects to start typing in the first 'tag-like' term in the task description rather than understanding its context as we normally do in case of Windows Explorer. We believe that this behavior will reduce with training and prolonged real-life use of the interface. Also, as number of files and tags increase, the suggestions in TAGZILLA will be able to provide more SOC help to users.
3. **Increase in likelihood of finding the most appropriate files:** We found that relevant files were not found in 3 cases by HFO users. All TAGZILLA users were able to find relevant files eventually. We attribute this to the reluctance of some HFO users to traverse all directories using brute force. Instead, the TAGZILLA users found it easier to try out multiple tags in case they chose the wrong tags.
4. **Reduction in time required to find related or similar files:** We did not have specific test cases to evaluate this IDO and will take it up in the next phase.
5. **Make PIM operations more natural:** We did not have measurable data on this IDO, though the I-S survey reports an increase of an average of 0.73 points.

We did not separately measure the storage times for HFO and TAGZILLA. But we found that qualitatively, storage for TAGZILLA was much quicker as the users had to use their SOC rather than think about where a file would go in an HFO.

We made an interesting observation with the correlation between the number of clicks and the time it took to retrieve a file (See Fig. 4). The median number of clicks for HFO was 4, while that for TAGZILLA was 3. We considered two groups X (<=3 clicks) and Y (>3 clicks). For group X, we found that the average time per click was 6.24 second for HFO and 5.47 second for TAGZILLA. For group Y, the same metric was for 3.09 second for HFO and 6.57 second for TAGZILLA. We explain this by some HFO users using brute force directory traversal when they cannot find a file. Since they are experienced in using the HFO interface, they do it efficiently. But when a user is able to find a file quickly, the TAGZILLA interface is more or as efficient as HFO. We believe that the brute force traversal is not practical with a larger number of folders and deeper trees. We also believe that as the number of files increase, the associations among files also increase enabling TAGZILLA to suggest related tags more often further increasing the efficiency of retrieval.

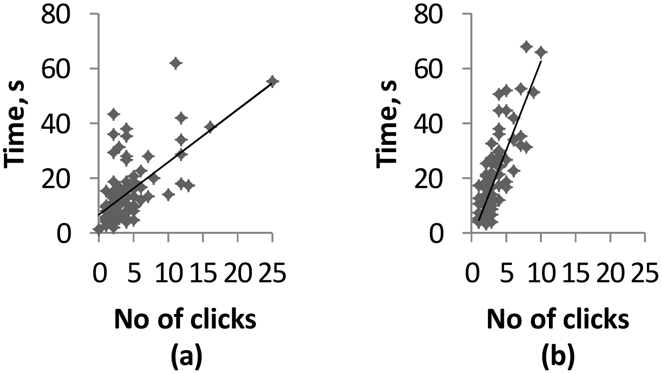


**Fig. 4.** Correlation between number of clicks and time of retrieval for (a) HFO and (b) TAGZILLA interfaces

## 6 Related Work

Researchers have proposed a variety of approaches to overcome deficiencies of HFOs: predictive systems such as FolderPredictor [11] which uses machine learning algorithms to predict folders a user might want to access, ‘Stuff I’ve seen’ [12] which indexes information from multiple sources that a user has ‘seen’ before; semantic systems such as the work by Faubel and Kuschel [2] which expands the metadata about a file into a folder path; and tagging-based solutions such as TagFS [3] and tagstore [9].

As tagging is a key concept in TAGZILLA, we discuss some of the tagging approaches developed to overcome the limitations of HFO.

Bloehdorn and Völkel [3] present TagFS that converts a folder structure into tags and adds semantic data such as name, user and a tag label. This makes the retrieval independent of the order of the folder structure. Unlike in TAGZILLA, TagFS does not provide tag suggestions and does not automatically tag files with their metadata.

The most relevant study to our work is the one by Voit et al. [9]. They present a comprehensive study using their tagging framework ‘tagstore’. Tagstore allows user to tag files and expands the tags into trees called ‘TagTrees’. So, a tagged file can be found under each permutation of trees created using its tags. Users use these HFO structures to locate their files. In doing so, they increase the likelihood of coming across a directory named after a tag, which houses symbolic links for other tags as well the file. A main differences between TAGZILLA and tagstore are that the former does not require the user to depend on HFO and it provides additional tags as suggestions based on their proximity in the tag graph.

In the same paper, Voit et al. present an experimental study comparing Microsoft Windows Explorer and tagstore. They report no noticeable difference in retrieval times between tagstore and HFO, while we report an increase of 27% in retrieval using TAGZILLA. This could be attributed to the users using a more familiar HFO for retrieval in case of tagstore. The same study also reports a decrease in number of clicks (27%) similar to that reported for TAGZILLA (26%). These two comparisons are interesting given that TAGZILLA did not use a familiar HFO for retrieval, had a larger time interval (1 week as opposed to 15 minutes) between storage and retrieval tasks, and did not have any test subjects familiar with tag-based interfaces. The same study also finds that 'fast' users performed well in tagging. While we second that, we report in our study that the users of HFOs who spent their time in brute force directory navigation were doing it 'fast', but not efficiently.

## 7 Conclusion and Outlook

We developed TAGZILLA, a tag-based PIM interface, as an alternative to HFO. The main contribution of TAGZILLA is that it provides a means to capture a user's SOC during file storage and retrieval as tags. It uses association among tags to suggest tags to the user and aid her SOC. Our other contribution is definition of 20 IDOs for PIM interfaces and determination of PIM functionality that needs to be improved in a traditional HFO using a user survey. We observed a 1 point increase on a 5-point satisfaction scale for users who took a test drive on TAGZILLA. We conducted a preliminary controlled experiment to compare the performance of TAGZILLA with a traditional HFO such as Microsoft Windows Explorer. We found that the number of clicks required for file retrieval reduced by an average of 26% with TAGZILLA, while the time required increased by an average of 27%. We found that TAGZILLA was comparable to HFO in retrieving files when files could be found in less number of clicks.

We propose to conduct more experiments in a realistic setting where users use TAGZILLA as an alternative PIM for longer periods of time. We also plan to improve tag suggestions with semantic association using WordNet and Wikipedia. We also plan to include the 'hard-won understanding of information' by users [13] using a tag-file visualization in the interface.

**Acknowledgments.** We would like to thank the associates of TRDDC for participating in user surveys, Yahya Poonawala and Natasha Shah for help in conducting surveys, and Maitreya Natu for review and useful comments. We also thank Sachin Lodha and Prof. Harrick Vin for their support in this activity.

## References

1. Bergman, O., et al.: Improved search engines and navigation preference in personal information management. ACM Transactions on Information Systems (TOIS) 26(4), 20 (2008)
2. Faubel, S., Kuschel, C.: Towards Semantic File System Interfaces. In: Proceedings of the Poster and Demonstration Session at the 7th International Semantic Web Conference, ISWC 2008, vol. 401 (2008)
3. Bloehdorn, S., et al.: TagFS -tag semantics for hierarchical file systems. In: Proceedings of the 6th International Conference on Knowledge Management, I-KNOW 2006 (2006)
4. Voit, K., Andrews, K., Slany, W.: Why personal information management (PIM) technologies are not widespread. In: ASIS&T 2009 Workshop on Personal Information Management, PIM 2009 (2009), http://pimworkshop.org/2009/papers/voit-pim2009.PDF.2009
5. Yang, C.-C.: Establishment and applications of the integrated model of service quality measurement. Managing Service Quality 13(4), 310–324 (2003)
6. Matzler, K., Hinterhuber, H.H.: How to make product development projects more successful by integrating Kano's model of customer satisfaction into quality function deployment. Hinterhuber 18(1), 25–38 (1998)
7. Ulwick, A.W.: Turn customer input into innovation. Harvard Business Review 80(1), 91 (2002)
8. Soanes, C., Stevenson, A. (eds.): Dictionary, Oxford English. Concise Oxford English Dictionary. Oxford University Press, Oxford (2006) (revised)
9. Voit, K., Andrews, K., Slany, W.: Tagging might not be slower than filing in folders. In: Proceedings of the 2012 ACM Annual Conference Extended Abstracts on Human Factors in Computing Systems Extended Abstracts. ACM (2012)
10. Barreau, D., Nardi, B.A.: Finding and reminding: file organization from the desktop. ACM SigChi. Bulletin 27(3), 39–43 (1995)
11. Bao, X., Dietterich, T.G.: FolderPredictor: Reducing the cost of reaching the right folder. ACM Transactions on Intelligent Systems and Technology (TIST) 2(1), 8 (2011)
12. Dumais, S., et al.: Stuff I've seen: a system for personal information retrieval and re-use. In: Proceedings of the 26th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval. ACM (2003)
13. Jones, W., et al.: Don't take my folders away!: organizing personal information to get things done. In: CHI 2005 Extended Abstracts on Human Factors in Computing Systems. ACM (2005)