Eulerview with projections: non hierarchical visualisation

Rosario De Chiara[1] and Andrew Fish[2][†]

[1]ISISLab - Dipartimento di Informatica ed Applicazioni "R.M. Capocelli" Università di Salerno - ITALY `dechiara@dia.unisa.it`
[2]School of Computing, Mathematical and Information Sciences, University of Brighton - UK `Andrew.Fish@brighton.ac.uk`

Abstract

EulerView is a novel resource management tool, enabling the representation of non-hierarchical categorisation structures within which to place resources. Since the initial incarnation for use in file-system management, it has been integrated with other systems, such as Flickr to produce Eulr, which assists in user manipulation of photo tags. Another system currently under development is Eulicious, a url management tool integrated with delicious the social bookmarking site. We bring these areas together, enabling the management of different types of resources, such as photo and url's, for user convenience. User feedback obtained from Eulr testing indicated that certain extra facilities would assist in their development of EulerView categorisation structures. To address this user request, we have introduced projections which capture information, within our broader non-hierarchical structure, in a similar manner to symbolic links in hierarchical structures. This will enable an easy transfer of user knowledge, whilst still providing the greater power of a non-hierarchical structure to be utilised when desired.

Categories and Subject Descriptors (according to ACM CCS): H.5.2 User Interfaces: Graphical user interfaces (GUI)
H.5.3 Group and Organization Interfaces: Web-based interaction
H.3.2 Information Storage
H.3.3 Information Search and Retrieval

1. Introduction and Background

Resource management is a difficult task. There now are large numbers of sites devoted to the managements of resources (for recording, searching, sharing, etc). Some of these target specific types of resources (eg Flickr for photos, delicious for bookmarks), and try to provide assistance with their management, often in a collaborative setting across the web; see Section 4 for details of a few of these. The development of a single interface for the systematic management of resources would be greatly beneficial.

In this paper, we discuss the evolution of the EulerView concept and its integration with various resource management tools. In section 1.1 we discuss hierarchies and their limitations, and then we talk about metadata and tagging in section 1.2. The limitations of the hierarchy and the flatness of tag-space leads us on to considering non-hierarchical classification systems in section 1.3. The EulerView solution is presented in section 2, together with the functionalities currently developed to aid in user tagging, searching and browsing tasks. Outcomes of user-testing is described in section 3, which has lead us to update EulerView to include projections and depict them visually. Section 4 demonstrates the integration of EulerView with various applications. Conclusions and future work are discussed in section 5.

1.1. Hierarchies

Hierarchical visualization seems a perfectly natural way of organizing and presenting information, both in real life and in computer applications. The main reason for this is, of course, that in real life *objects can be just in one place*. A real office room is the inspiration for the user interface metaphor that has been implemented, in various flavours, in computer operating systems for decades. This is a well-known mechanism [CC98] in user-interface design: the human mind organizes external world information spatially in a natural way, so that a cognitive map can be seen as the internalized analogy, in the human mind, of the physical layout of the environment. By mimicking a real office where a document,

---

a mail message, or a sheet of paper, can be placed in only one folder, a computer interface allows a file to be placed exclusively in one directory by using a mechanism called *single inheritance*. This is actually felt to also be a limitation in a real world setting, as shown by the results of a study undertaken in a real office [Mal83]. In this study, office workers were interviewed in order to analyze the organization of the sheets, book and documents on their desks which needed to be categorized. The interviews indicate that users had difficulties in categorizing, with quotes such as "...*the hardest problem for me organizationally is deciding what the categories are and what category something is in*" and "...[referring to a pile of documents kept on his desktop] *It's interesting how hard it is to characterize these things - it's almost like leaving them out means that I don't have to characterize them...*".

The Treeview control is the traditional way of visualizing hierarchical information in user interfaces. It has a standard look and feel and it is integrated in modern programming language environments. Although the Treeview control has its limitations, it has been available since the character based terminal era and is still an effective visualization technique which is suitable for a wide range of different applications. However there are many different techniques to visualize hierarchies, including: the 2D node-link diagram [Joh92], the horizontal family tree diagram [Knu97] and the radial tree diagram [LR94]. Within the last decade, novel visualization methods have been developed for displaying large hierarchies, including the Treemap [Joh92], cone tree [RMC91], disc tree [JP98], hyperbolic tree [LR94], and 3D hyperbolic tree [Mun98] visualizations.

The limitations of the usefulness of hierarchies in categorizing information are demonstrated by the categorization of web sites within a huge ontology. The Open Directory Project (ODP) is the most comprehensive human-edited directory of the Web. It is constructed and maintained by a vast community of authors. Open Directory is a *web directory*, which is a directory of categorized web links. This differs from a search engine because it is built by users; *editors* look for the most appropriate subtree in the ODP directory in which to place a particular link to a website, and they provided a short resume of the contents of the corresponding website. The ODP goal is to provide a structure capable of containing and categorizing all of the websites that are available on the Internet. The ODP solution to the categorization problem is to use *symbolic links* between categories in order to allow the content of a subtree, corresponding to a category, to *appear* under an unlimited number of other subtrees. Symbolic links are an ad-hoc solution which can loose some of their appeal if the structure is much more complex than a simple hierarchy. For example, in the ODP hierarchy, the category "Art" contains a multitude of symbolic links to other subtrees spread along the hierarchy. In fact, the ODP hierarchy contains more than $630,000$ different topics and more than $700,000$ symbolic links. This suggests that the

use of symbolic links is not a method which is used to manage a few complex categories, but their deployment is actually *systematic* and is necessary in order to describe the general categorization structure. We conclude that *categorizing information requires something more expressive than a hierarchy*.

The categorization process can be viewed as the process of assigning *tags* to items. Before we move on to broader classification system than hierarchies we describe the notions of metadata and tagging, so that we can refer to the use of the classification structure in the tagging process.

1.2. Metadata and tagging

Metadata refers to data about data, and typical usage involves data about resources such as documents, books, articles or photos. Metadata are specific to the particular purpose they are designed for (e.g. metadata that can be associated to books). In the file system context, metadata are particular fields of information kept in a structure that permits querying, in order to assemble "virtual directories" of files on-the-fly, for example. The metadata approach is implemented in various solutions [spo]. They are used to facilitate the retrieval of those specific resources. The importance of Metadata is increasing, especially in web-related activities, because many web sites allow users to add metadata to resources such as photo, URLs and blog entries. In this particular context the operation of adding metadata has been dubbed *tagging*.

Tags are an effective way for authors to categorize their resources: they facilitate future retrieval of information using tags as keywords. Similarly, tags are useful for readers who are browsing for resources. Although an author may be the best person to assign tags to his/her resources, his/her resources may mean something different to other people, or they may just perceive the tags used in a different manner. For this reason web sites like Del.icio.us allow users to tag other people's resources. This *collaborative tagging* has been dubbed "Folksonomy" [fol, Mat04], joining the term *folk* and the term *taxonomy*; "specifically it refers to subject indexing systems created within Internet communities". Tagging tasks can be described as the process of adding, or altering, tags to resources, where this might involve adding multiple tags to multiple resources. Whilst free-form user tagging allows quick single resource tagging they do not give an idea of the related categorisation structure, which would aid user awareness and facilitate quick tagging of multiple related resources.

Relating tags to hierarchies A hierarchy is used to represent the directories of a file system (or categories in ODP) and can be represented by a single rooted tree. Consider the directory names as the tags which are used to categorize information kept in files. In a hierarchical file system, only one such tag can be assigned to a file, because a file can be just in
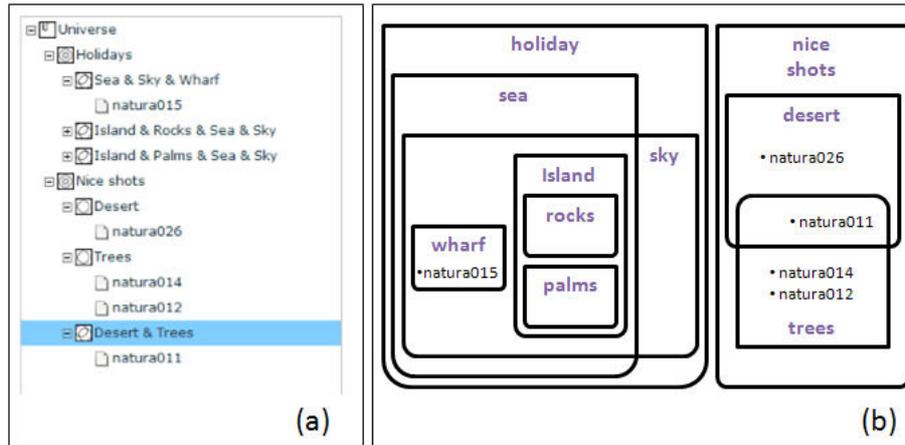
Figure 1: *(a) EulerView and (b) an associated Euler diagram.*

one node of the tree. For many realistic categorisation tasks hierarchies are not sufficiently rich.

1.3. Beyond Hierarchies

Polyarchy More complex underlying structures can be modelled by Polyarchies which are multiple intersecting hierarchies. In [RCCR02], Polyarchies are defined and visualization methods to try to raise the user's awareness of the structure of the polyarchy are provided. Considering a Polyarchy as a categorization structure we can easily see how an element can have more than one tag attached to it – such elements will be part of various overlapping hierarchies, each of which assigns a tag to the element. In [MS04], polyarchies are viewed as edge coloured multigraphs; effectively they can be thought of as the union of trees, each corresponding to a single hierarchy. Traditionally, the representational abilities of hierarchical visualizations are extended using symbolic links, and although you could view a polyarchy as an instance of a hierarchy extended with symbolic links, it seems more appropriate to accept and incorporate the complexity of the categorization problem into one's underlying model, and work with the more general structures.

Euler diagrams Euler diagrams are a well known, intuitive method of representing certain relationships between sets. Traditionally, simple closed curves in the plane (usually labelled) are used to represent sets and the spatial relationships between curves (intersection, containment and disjointness) are used to represent the corresponding set-theoretic relationships. One of the reasons that they are said to be an effective representation is that these spatial relationships are well-matched to the corresponding domain relationships they represent [Gur99]. In the file-system context, the sets are the categories and one can represent sub categories, intersections between categories, and disjointness of

categories. Thus they allow a user to categorize into categories that are not necessarily disjoint (i.e. they allow multiple inheritance). As well as enabling easier initial categorization, if a user has two separate classifications and the desire to merge them (e.g. merging favourite lists from different browsers or different computers) then it becomes feasible to do this easily, since categories from one classification do not need to be disjoint from categories in the other. One difficulty with Euler diagram based languages is the common visual langauge problem of *overspecificity* [Shi04]: if you wish to represent information about a set then you have to specify its relationship with all of the other sets which are already represented. If a large number of curves (or a large number of items densely packed together within a region) are present in a diagram then this can also lead to the diagram being cluttered (see [JFHT06] for a formal definition of clutter of Spider diagrams and empirical evaluation justifying the measure). One approach to try to address this visualisation problem is to introduce projections, as in [GHT01, Joh04], which enable the specification of relationships within a specific context.

2. The EulerView solution

We wished to enhance the user experience of tagging resources by providing them with the facility to systematically create and manipulate their own categorisation structure to update tags for resources. Since hierarchical structures were deemed not sufficient, a non-hierarchical structure was required. Furthermore, we wished to make the new structure easily accessible. To this end, we have developed EulerView which is an extension of Treeview and has a similar look and feel, addressing the potential migration issues from TreeView and the potential navigation issues arising from 2D representations, but with familiar characteristics, and an underlying model, based on Euler diagrams to enable

a more general representation than hierarchies. EulerView could replace Treeview in any application where a hierarchical visualization can be felt as a limitation, such as managing bookmarks in browsers (or e-mail messages). In the future we intend to investigate the advantages of EulerView over other resource management tools and also to establish if it is advantageous to add further functionality to facilitate other user related tasks such as searching and browsing.

EulerView enables the systematic management of tags. This brings many advantages such as enhancing the users' understanding of the categorisation structure (allowing user construction and manipulation of such a structure which displays relationships between resources); this can be a difficult comprehension task using "flat" tags. It has also been developed within a constraint-based framework, where the level of constraint could be varied, according the the application domain or user preferences, for instance. A user can build a certain view of an Euler diagram using the EulerView. Actually what they are really doing is building a set of paths in the transitive closure of the directed dual graph of an Euler diagram. Figure 1(a) shows the EulerView and Figure 1(b) shows a corresponding Euler Diagram.

2.1. EulerView functionalities

We describe functionalities that have been developed with the evolution of EulerView, showing how it extends Treeview. Some of these techniques were added to assist users who wish to tag, to search or to browse through resources using a large EulerView; they should ease the burden of these tasks by reducing the amount of information displayed according to user setting of context.

- **Iconicity:** is used to give consistent information about the relationship of the set with other sets in the data structure. In Figure 1(a) the standard icon used is a single circle icon shown beside `Trees` indicates that there are no subcategories in the EulerView, whilst the icon beside `Desert & Trees` indicates an intersection between two (or more) sets. The two concentric circles icon beside `Nice shots` indicates that there is at least one subset "inside `Nice shots`" within the data structure, and a single circle inside an intersection icon means that there is at least one subset in the data structure which is completely within this intersection.

- **Set manipulation:** the drag and drop of set labels enables the user to choose whether to create a subset or an intersection. A strict runtime control is performed to check the consistency of the diagram by disallowing certain node move operations that would violate the underlying Euler diagram model. For instance, `Set A` cannot be placed inside the intersection `Set A & Set B`.

- **Item manipulation:** dragging an item into an intersection coherently creates symbolic links to the item in the relevant places.

- **Renaming:** renaming a set invokes an automatic rename

of all of the relevant set intersections, but in order to keep consistency, one cannot rename an intersection itself (whose label is determined by the labels of the sets involved in the intersection).

- **Coloured items:** one may colour categories in order to provide a visual classification. On the creation of an intersection among two categories, the two colours are coherently blended in the intersection label.

- **Import/Export and wizards:** we enable the import and export of multiple resources from various applications such as Flickr and the ACM CCS, the main criteria for allowing this functionality is the ability to interchange XML.

- **Breadcrumbs:** *Breadcrumbs* [Ber88] are well known technique used in web pages to convey to user the information about his location within the website structure. The information takes the form of a trail of crumbs from the homepage to the page the user is currently visiting. From a structural point of view this trail is just one of the possible paths from the root-node within the website graph. We exploit this representation in EulerView in order to provide the same kind of information to user. An added value of the breadcrumbs use in EulerView is that it shows a "trails of categories" that is the current *context* which the current selected node of EulerView refers to. In Figure 2 the breadcrumbs for the selected node `Graphical User Interface(GUI)` are shown.

- **Branch compression**
  We allow a user to select a path in the EulerView display and compress it to a single vertex. This utilises the facility of allowing multiple labels on a single vertex. For example, in Figure 2, we show the effects of branch compression: the whole branch from Information Systems to Graphical User Interfaces shown in the top of the figure is compressed to the single vertex whose label is the conjunction of the labels in this path. Considering the breadcrumbs of a selected node, the branch compression feature enables the user to create a category that is *the intersection of the categories currently shown in the breadcrumbs*.

- **Zoomable Universe** We incorporate a facility called a *zoomable universe*: a user can zoom in on a category vertex of the EulerView display, displaying that vertex in the place of the Universe vertex, thereby setting that node to be the local context. While zooming the user is aware of the context via the breadcrumbs which always refer to the Universe-rooted display.

- **Assisted search facilities** Users are used to being able to perform text searches, and we have included this functionality. The search facility allows the user to automatically zoom in on the first category matching the string. Repeated searching for the same string cycles through all the occurrences of the string.
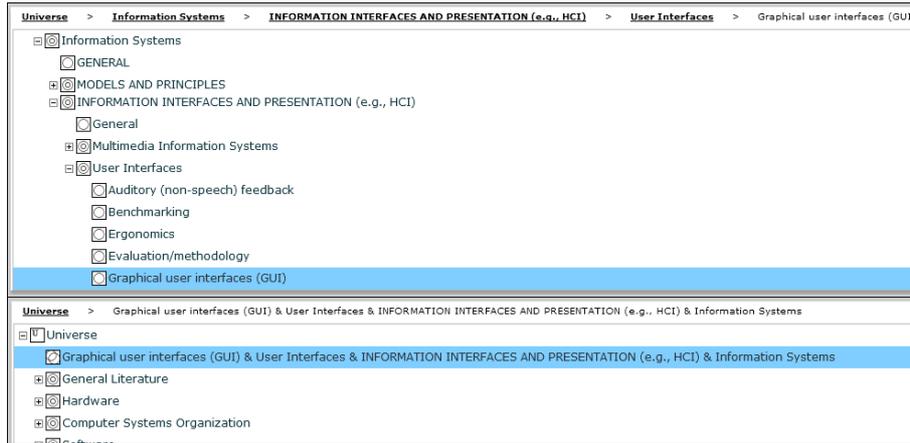
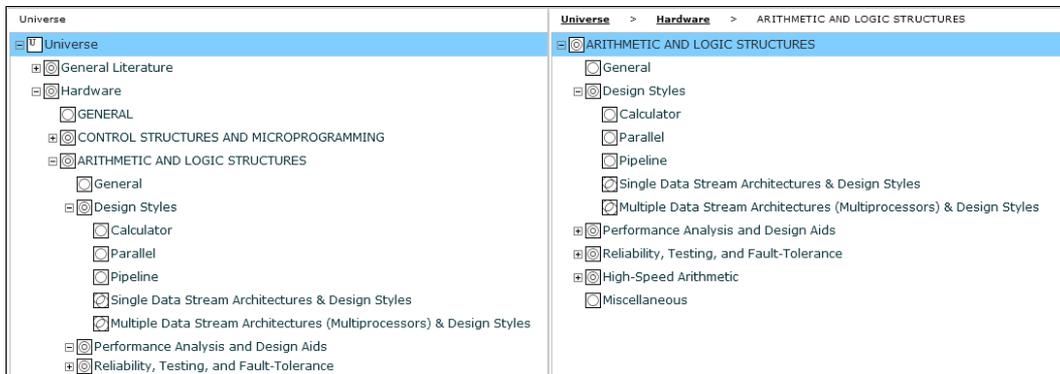**Figure 2:** *Branches of the EulerView can be compressed to simplify the view*

Figure 3: *A zoomable universe enables the user to set the local context.*

3. Usability tests

In [DF07], we reported on a experimental studies performed to give an indication of the usefulness of the EulerView component for the purpose of quick single-person bookmarking (by comparison with the slow multiple-person bookmarking in the ODP), as well as to investigate the necessity of offering 2D representations together with the EulerView control. Results and user feedback suggest that EulerView is perceived as an easy to use extension of the Treeview control, and that users seem to be able to perform accurate quick bookmarking, whilst storing a fairly accurate internal representation of the underlying Euler diagram model. One future avenue for assessing the practical usefulness of EulerView will be by user-comparison of the EulerView performance with other categorization methods (such as Treeview with symbolic links, polyarchies, etc). In [DF07] a test of user perception of Eulr and its features was reported and user feedback indicated that the Eulr representation is intuitive and that the assimilation of concepts into a governing framework is excellent. User feedback also indicated that cer-

tain extra facilities, corresponding the use of symbolic links, would assist in their development of EulerView categorisation structures. We have included projections in EulerView, and a method of depiction to make the user aware of their presence, to address this request. Furthermore, this has lead on to current work which relates to the import of the ACM classification into an EulerView display: one of the translations uses exactly this feature.

Figure 4 shows the use of projections depicted clearly within the EulerView display. On the left there is an Eulerview showing sets named Italy, Swiss, Holy See and San Marino, whilst on the right is a corresponding Euler diagram with projections (note that the usual Euler diagram semantics are modified here to relax the disjointness of sets constraint). This example shows that San Marino and the Holy See territories are totally contained by Italy territory. The use of projections enables us to depict that there is a part of Italian territory that is totally contained within the Swiss boundaries, and this territory, which is an *enclave*, contains a city named Campione d'Italia.
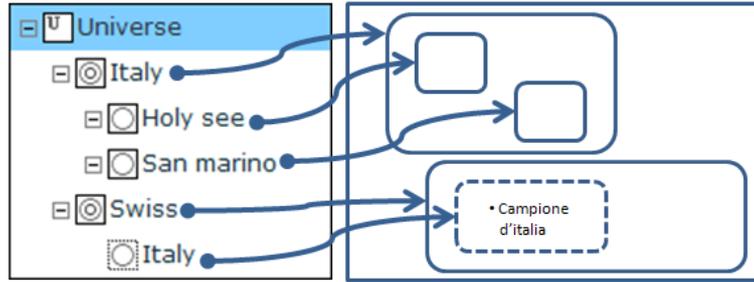
Figure 4: *The use of projections: on the left an EulerView containing some sets and a projection; on the right a corresponding (modified) Euler diagram with projections.*

4. Scenarios of use

We describe some application areas that EulerView has, or is currently being, integrated with.

4.1. Delicious

Delicious [del] is a social bookmarking service for storing, sharing, and discovering web bookmarks. Bookmarks are stored in a flat space (which is not hierarchical): every user can attach any number of tags to their bookmarks. The social nature of bookmarking is emphasized by the fact that Delicious allows users to view bookmarks added by similar-minded users: users can browse through bookmarks using the tags they or others have attached. The service, at time of writing, boasts more than three million users and 100 millions bookmarks. It is worth noting that all the features available on the website are also available through an effective API. Figure 6 shows Eulicious [DF08], with the EulerView pane on the left and the right pane showing delicious.

4.2. ACM

The ACM Computing Classification System (ACM CCS) [acm] is a standard classification method for computer research articles; a useful description of the ACM CCS can be found at [VRG05] where they actually propose a unified classification system for papers related to computer science, software engineering and information systems. Figure 6 shows the importation of EulerView with the ACM CCS. Authors are required to find the appropriate place within the ACM classification structure to classify it.

4.3. Flickr

Flickr is a popular photo-sharing site that exploits tags as a core element to the sharing, retrieval, navigation, and discovery of user-contributed images. Every Flickr user can upload his/her photos to be stored online and they can choose

to allow these photos to be publicly viewable and therefore easily discoverable. Making photos accessible to the public, together with strong emphasis on the tagging facility [MMN*06] has facilitated the expansion of the site; currently it boasts more than 700,000 registered users. A functionality of Flickr which is greatly appreciated is the support of social interactions: in addition to uploading photos, users can create networks of friends, join groups, send messages to other users, comment on photos, tag photos, choose their favourite photos, and so on. Figure 5 shows the integration of EulerView with Flickr, showing the EulerView on the left and right pane being Flickr; see [DFR08] for more details.

5. Conclusion and Future Works

The problem of how to facilitate user categorization for bookmark or file management is a difficult one. The traditional categorization using hierarchies can be prone to limitations due to single inheritance problems (that is the constraint of having just one tag associated to every item). When a more complex categorization is essential, such as in the ODP, hierarchies can be extended using symbolic links, or polyarchies are used to help visualize multiple intersecting hierarchies. Alternative visualization methods are available [Spo93] to help with the exploration of multidimensional (that is, multiple tags) information. A limitation of these techniques is that they need a complex interface that is not likely to be easily embedded in traditional applications (such as a web browser), and they will be unfamiliar and may require a relatively long training period of time for new users. EulerView has been developed with the purpose of facilitating the visualization and management of information by general users. It extends the hierarchical visualization of Treeview, keeping a non-invasive interface that can be easily embedded in an Internet browser. Using an underlying Euler diagram based model enables the use of overlapping categories, whilst keeping a similar look and navigational structure of the standard Treeview aids in migration.

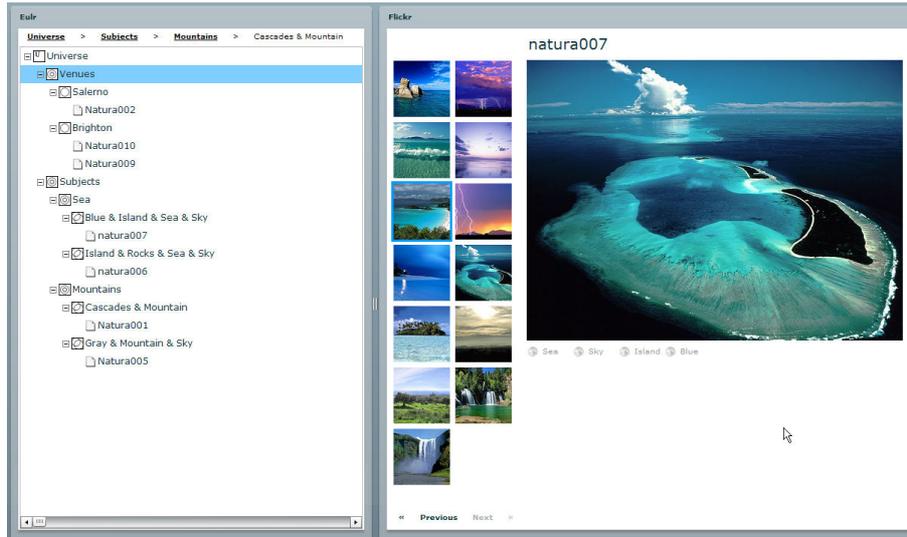In this paper we have brought together and surveyed various incarnations of the EulerView concept in different re-
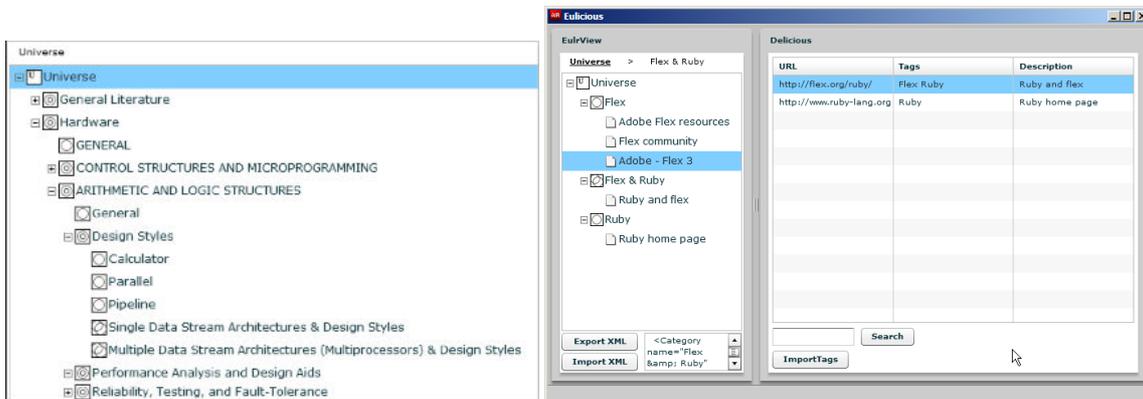
Figure 5: *Eulr*

Figure 6: *Left: EulerView with ACM CCS imported. Right: Eulicious.*

source management scenarios. A major advantage of Eu-lerView is that it can be used in any such context, for the management of any resources, and so a user would avoid the necessity of learning to use several different resource management tools for different types of resources. As a result of earlier user feedback, we have introduced the notion of projections into EulerView. This effectively enables the users to mimic the creation of use of symbolic links together with the normal non-hierarchical EulerView structure, thus providing more flexibility to the user and helping with migration from the use of the common TreeView with symbolic links.

Techniques for improve focus and visual emphasis for TreeViewers were described in [WS97], and some of these features may be incorporated into EulerView in the future. Investigating the potential for integration of EulerView with the ideas of stepping stones and pathways [FNY*06] may

also prove fruitful when considering searching and browsing tasks.

References

[acm] ACM CCS: ACM Computing Classification System [1998 version]. http://www.acm.org/class/1998.

[Ber88] BERNSTEIN M.: The bookmark and the compass: orientation tools for hypertext users. *SIGOIS Bull. 9*, 4 (1988), 34–45.

[CC98] CHEN C., CZERWINSKI M.: From latent semantics to spatial hypertext - an integrated approach. In *Proceedings of the Ninth ACM Conference on Hypertext and Hypermedia (Hypertext '98)* (1998), ACM Press, pp. 77–86.

[DF07] DE CHIARA R., FISH A.: Eulerview: a non-hierarchical visualization component. In *VLHCC '07: Proceedings of the IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC 2007)*, IEEE Computer Society, pp. 145–152.

[DF08] DE CHIARA R., FISH A.: eul.icio.us: Euler diagrams for del.icio.us. In *IV '08: Proceedings of the 12ᵗʰ International Conference Information Visualisation (IV 2008)*, (to appear).

[DFR08] DE CHIARA R., FISH A., RUOCCO S.: Eulr: a Novel Resource Tagging Facility Integrated with Flickr. In *AVI '08: Proceedings of Advanced Visual Interfaces (AVI 2008)*, (to appear).

[del] Del.icio.us Social Bookmarking. http://del.icio.us.

[FNY*06] FOX E., NEVES F., YU X., SHAN R., KIM S., FAN W.: Exploring the computing literature with visualization and stepping stones & pathways. *Communications of the ACM*, 4 (2006), 53–58.

[fol] Wikipedia: Folksonomy. http://en.wikipedia.org/wiki/Folksonomy.

[RMC91] ROBERTSON G. AND MACKINLAY J. D. AND CARD S. K.: Cone Trees: Animated 3D Visualizations of Hierarchical Information. In *Proc. of CHI-91* (New Orleans, LA, 1991), pp. 189–194.

[GHT01] GIL J., HOWSE J., TULCHINSKY E.: Positive semantics of projections. *Journal of Visual Languages and Computing 13*, 2 (April 2001), 197–227.

[Gur99] GURR C.: Effective diagrammatic communication: Syntactic, semantic and pragmatic issues. *Journal of Visual Languages and Computing 10*, 4 (1999), 317–342.

[JFHT06] JOHN C., FISH A., HOWSE J., TAYLOR J.: Exploring the notion of clutter in Euler diagrams. In *Proceedings of the 4th International Conference on the Theory and Application of Diagrams* (Stanford, USA, 2006), Springer, pp. 267–282.

[Joh92] JOHNSON B.: TreeViz: treemap visualization of hierarchically structured information. In *Proceedings of the SIGCHI conference on Human factors in computing systems* (1992), ACM Press, pp. 369–370.

[Joh04] JOHN C.: Reasoning with projected contours. In *Proceedings of 3rd International Conference on the Theory and Application of Diagrams* (Cambridge, UK, 2004), vol. 2980 of *LNAI*, Springer, pp. 147–150.

[JP98] JEONG C.-S., PANG A.: Reconfigurable disc trees for visualizing large hierarchical information space. In *Proceedings of the 1998 IEEE Symposium on Information Visualization* (1998), IEEE Computer Society, pp. 19–25.

[Knu97] KNUTH D. E.: *Fundamental Algorithms*, Third ed., vol. 1 of *The Art of Computer Programming*. Addison-Wesley, 1997, pp. 308–316.

[LR94] LAMPING J., RAO R.: Laying out and visualizing large trees using a hyperbolic space. In *ACM Symposium on User Interface Software and Technology* (1994), pp. 13–14.

[Mat04] MATHES A.: *Folksonomies - Cooperative Classification and Communication Through Shared Metadata*. Technical report, 2004. http://www.adammathes.com/academic/computer-mediated-communication/folksonomies.html.

[MMN*06] MARLOW C., MARLOW C., NAAMAN M., BOYD D., DAVIS M.: Ht06, tagging paper, taxonomy, flickr, academic article, to read. In *HYPERTEXT '06: Proceedings of the seventeenth conference on Hypertext and hypermedia* (New York, NY, USA, 2006), ACM, pp. 31–40.

[MS04] MCGUFFIN M. J., SCHRAEFEL M.: A comparison of hyperstructures: Zzstructures, mspaces, and polyarchies. In *Proceedings of the fifteenth ACM conference on Hypertext and hypermedia* (2004), ACM Press, pp. 153–162.

[RCCR02] ROBERTSON G., CAMERON K., CZERWINSKI M., ROBBINS D.: Polyarchy visualization: visualizing multiple intersecting hierarchies. In *Proceedings of the SIGCHI conference on Human factors in computing systems* (2002), ACM Press, pp. 423–430.

[Shi04] SHIMOJIMA A.: Inferential and expressive capacities of graphical representations: Survey and some generalizations. In *Diagrammatic Representation and Inference: proceedings of Diagrams 2004* (2004), vol. 2980 of *Lecture Notes in Computer Science*, Springer, pp. 18–21.

[spo] Apple Spotlight Technology. http://www.apple.com/macosx/features/spotlight.

[Spo93] SPOERRI A.: Infocrystal: a visual tool for information retrieval & management. In *"CIKM '93: Proceedings of the second international conference on Information and knowledge management"* (1993), ACM Press, pp. 11–20.

[Mal83] MALONE T. : How do people organize their desks?: Implications for the design of office information systems. *ACM Trans. Inf. Syst. 1*, 1 (1983), 99–112.

[Mun98] MUNZNER T.: Exploring Large Graphs in 3D Hyperbolic Space. *IEEE Comput. Graph. Appl. 18*, 4 (1998), 18–23.

[VRG05] VESSEY I., RAMESH V., GLASS R.: A unified classification for research in the computing disciplines. *Information and Systems Technology* (2005), 245–255.

[WS97] WITTENBURG K., SIGMAN E.: Visual focusing and transition techniques in a treeviewer for web information access. In *Visual Languages* (1997), pp. 20–27.