FunEuler: an Euler Diagram based Interface Enhanced with Region-based Functionalities

Gennaro Cordasco[1], Rosario De Chiara[2], Andrew Fish[3]⋆, and Vittorio Scarano[2]

[1] Dipartimento di Psicologia - Seconda Università di Napoli - ITALY
[2] ISISLab - Dipartimento di Informatica - Università di Salerno - ITALY
[3] School of Computing, Mathematical and Information Sciences, University of Brighton - UK

Abstract. Euler Diagrams are an accessible means of representing non hierarchical set-based relationships which have recently been used in resource management interfaces to facilitate user categorisation. We develop a novel, extensible Euler diagram based interface, called FunEuler, which integrates the concepts of visual classification, spatial arrangements and functional application, thereby greatly extending the power of such Euler diagram based interfaces by enabling fast application of a collection of predefined functions to collections of categorised resources. To demonstrate the principle, we provide several functionalities such as file zipping or creating playlists within the application, whilst also providing a mechanism to extend the functionality to facilitate end user development. Preliminary user testing suggests that the Euler diagram concept is easily comprehensible for resource categorisation purposes, the concept and application of functions can be understood and applied successfully, and that users perceived the addition of functions increased the usefulness of the application for repetitive tasks.

1 Introduction

User based resource management tasks such as file categorisation are clearly important tasks and yet there are serious limitations within current methodologies to assist users. The traditional hierarchical-based representations are well known to have limitations in the realms of user categorisation, with difficulties occurring when a user wants to categorise resources in more than one place, to change the categorisation structure, or to perform operations on sets of resources that are spread over multiple directories. An alternative methodology is via tagging, and there are methods to handle specific resources such as `mp3`'s or particular types of tagged documents, but they can be problematic if the user wishes to utilise file type based methods for types such as `TeX` or `eps`, for instance; also, tag-based methodologies have the downside that documents with missing or mis-typed tags may be omitted from searches, for example. Some modern operating systems do allow the use of facilities for both tag and file-type based save, search and retrieval methods. As a proposed alternative approach, Euler Diagrams are a means of representing non-hierarchical set-based relationships. They have recently been used in resource management interfaces to facilitate user categorisation, in [15, 5], for instance.

Looking from another perspective, we know that the organisation of desktop files or links into groups can enhance working memory by utilising the spatial information [4, 12, 14]. However, no methodologies exist, to the best of our knowledge, that bring together the avenues of spatial organisation, categorisation and functional application. Doing so enables the application of functions to spatially organised (or categorised) resources, utilising the user's own spatial memory, whilst also providing the powerful feature of functional application to sets of the spatially organised resources. Therefore, we extend the power of an Euler diagram based interface, enabling the application of a collection of pre-defined functions to collections of categorised resources. The interface enables the user to quickly create a diagram which captures the categories of interest, and their intersections, and utilises a drag and drop facility to apply functionalities to the intersection of categories that are of interest (or to a union of such intersections).

Key ideas. One of the key concepts considered in this paper is the extension of the concept of a *palette*, which is an area where collections of resources of interest can be placed and manipulated, to a *spatial palette* in which these areas are arranged into non-hierarchical regions, as defined by an Euler diagram. We have developed an Euler diagram based interface for user-based categorisation within a non-hierarchical structure, which also allows users to apply functionalities to collections of resources placed within that structure. From an application point of view, we provide means to extend the functionalities in order to encourage end user development and aid future uptake of the interface.

To provide a concrete motivational example, we present one of the user based scenarios that we have developed,set within the mp3 application domain. Then, in Section 2, we provide Euler diagram interface background information and describe related works. The application is described in detail in Section 3, and some preliminary user testing is reported in Section 4. Subsequent to the user test, we developed a specialisation of the FunEuler interface within the music domain, providing an indication of how the interface can be specialised to cater for domain specific applications; this is reported in Section 5. Conclusions and future work avenues are suggested in Section 6.

User based scenarios. Let us imagine George, a local student DJ, who often creates music based gifts for his friends as well as constructing larger distributions. Related tasks that he performs include: selecting a collection of mp3 tracks and creating a playlist; naming the playlist and printing out the tracklist as a PDF; zipping a playlist together with a tracklist, or collections of these; emailing the zip file, or uploading it to a music-sharing website; creating multiple zip files of playlists, together with tracklists, plus a global list of the zipped packages in order to distribute more efficiently. By developing a spatial palette (which is a virtual space in which he can arrange his resources and have the facilities to apply the relevant functionalities to them) we will assist George in performing his tasks more efficiently. To gain long term benefits, he would also like to store his music files in an accessible manner, and related tasks he performs include: categorising a collection of mp3 files, building a structure to hold them, storing (or tagging) the created zip files appropriately. These tasks are also facilitated via the use of an Euler diagram based structure and interface.

2 Euler Diagram interfaces

Venn diagrams [17] are sets of simple closed curves in the plane such that every *zone* (a region that is inside a set of those curves and outside the remaining curves) is a non-empty connected region of the plane. Although there are many variations, especially in terms of *wellformedness condtions* (geometric or topological conditions that one can impose on the diagrams, commonly with the intent of reducing the risks of human miscomprehension), the main difference is that for Euler diagrams not all of these zones have to be present (i.e. non-empty). Commonly these curves are labelled to indicate the set that they represent, and the diagrammatic systems can be extend to incorporate more information content. Currently there are many application areas for Euler diagram based interfaces related to diagrammatic logic and software specification, and set based data visualisation. Here we present only those interfaces which are directly relevant to our work, being related to enhancing searching and categorisation tasks. The basic architecture and insights into FunEuler were described in [3]. In this paper we specialize FunEuler to a specific context (managing mp3 files, see Section 5). This provide a simple scenario of use inspired by real life, with motivation easily comprehended by users without explanation.

Related Work. A clickable Venn diagram interface was developed in [13] with the idea of improving web search transparency (the ability to see how sub-queries contribute to the entire result set of a search). It showed the number of results displayed by each term or group of terms in the diagram, but was limited to three search terms and the queries performed upon clicking each region of the diagram were just the conjunction of the terms of the containing curves. Euler diagram based query interfaces have also been used in Traditional Library Environments [11] where the curves represent the query terms, and numbers are used to indicate the number of search term results in the database returned from the Boolean expression associated to the region, but with a slightly different meaning: a region which is outside a curve also means that that curves' search term does not occur in the search results.

Venn diagrams have been used to represent non-hierarchical directories, replacing the traditional hierarchical structure of file systems [15], where diagrams could be drawn with curves representing categories (or tags) and files could be placed within more than one directory by utilising a region of overlap of the contours. In [5], an Euler diagram interface was developed, enabling more general resource management, together with efficient interpretation algorithms to detect the underlying meaning of the regions of the diagram. A reification of an Euler diagram based categorisation structure was integrated with Flickr in [16], utilising the non-hierarchical categorisation structure.

There has been relatively little actual user testing of the Euler diagram concept, with the idea generally being taken for granted as being beneficial. In [2], the comprehension of basic Euler diagrams (without items) that had the same zone sets, but differed in terms of properties such as jaggedness of curves, was examined, whilst in [6], an investigation of the effects of varying wellformedness conditions (e.g. of typical conditions are: no concurrency, no more than two curves crossing at a point, etc.) imposed

on user comprehension and preference was performed. In terms of human reasoning processes with Euler diagrams, in [18], eye-tracking experiments to investigate user's focus changes during reasoning tasks were performed. In [7] an in-depth study was presented that aims at measuring the influence of the wellformedness conditions on the comprehension of a diagram; the tests utilised paper based materials.

Background. We view the zones of an Euler diagram as a repository in which to place resources, such as `files` and `urls`, and so the basic notion of Euler diagrams is extended to capture the placement of items in the diagram; this is similar to unitary alpha Spider diagrams in the diagrammatic logic context [9]. However, in terms of the semantics assigned within this application area, we simply assign the set of tags in a zone's description to that zone, and extend this assignation to any items placed in that zone. Thus an Euler diagram provides a means to build a non-hierarchical categorisation structure, and to use it to categorise resources, as in [5].

3   Application description

We describe the FunEuler interface, together with justifications for the design decisions adopted. In Figure 1 a screenshot of the new FunEuler application developed is shown. The application is available at [1]. The main portion of the window real estate contains the diagram itself and this is where the main user-interaction occurs. At the top of the left hand pane, a list of icons representing the possible operations that the user may apply (to regions of the diagram) is displayed. Below this operations list there is a special area displaying a simple diagram which indicates the *Results* set. This essentially depicts the output from the application of the user-selected operations (see Subsection 3.2 for more details). At the top of the interface, a standard menu bar enables the user to select different interaction modes: modify a curve (Arrow icon), draw new curve (Pencil icon), query the diagram (Eyedropper icon), reset the zoom level (Magnifying glass icon), tag selected items (Tag icon) and save the diagram (Camera icon).

3.1   Diagram construction and interaction

We allow users to construct a diagram by adding curves, in the form of ellipses, one at a time by a simple mechanism of left click and drag to specify one axis whilst using the mouse scroll wheel to specify the other axis (cf. Figure 2 top-left). This allows fast user construction of the diagrams, as well as a fast interpretation of the zones of the diagram [5]. The downside is that it slightly restricts the freedom of users who may wish to draw other types of curves, but the use of more general curves can be considered as a future extension where the associated trade-offs will be investigated. However, we emphasise the important point that in this application domain users only need to construct diagrams which have a superset of their required collection of set intersections that they need to represent and so concerns relating to the complexity of diagram construction [8] are reduced.

At the moment of creation of a curve, the system automatically assigns a random colour to the curves, as well as a choice of set name (according to a predetermined sequence), but these names and colours can be then edited by the user. This facilitates
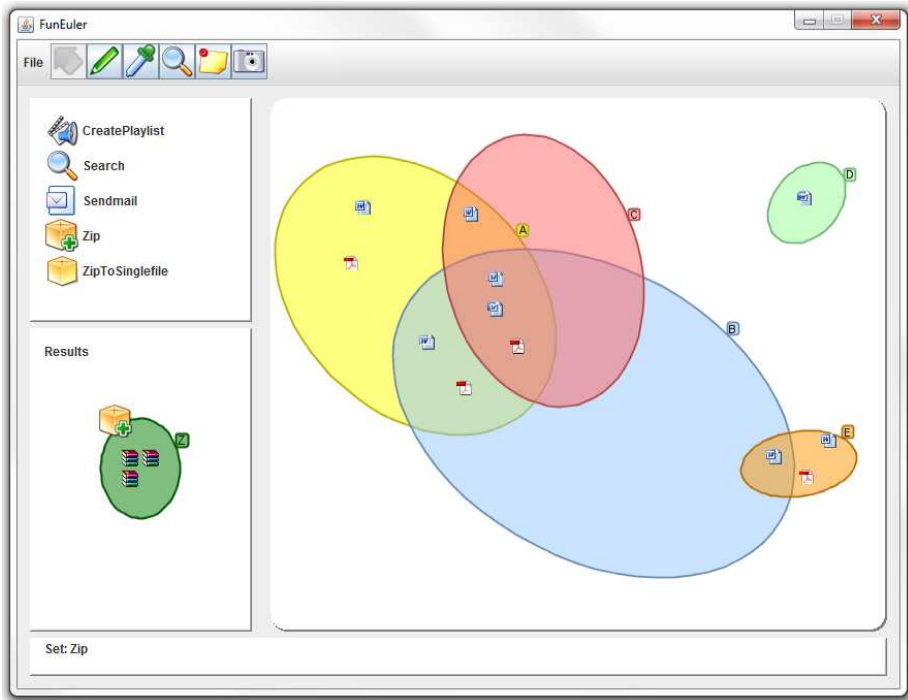
Fig. 1. A screenshot of the FunEuler interface.

rapid user construction of diagrams via ellipse addition without the interruption of naming and choosing a colour for each curve, which could interrupt the flow of creation. The collection of parameters associated to each ellipse are easily accessible through a pop-up menu via a right-click action (cf. Figure 2 top-middle). Existing ellipses can be quickly modified by translation, rotation or by altering the size of its axes.

Although the general advice would be that users constructing categorisation diagrams should keep them as simple as possible, human readers of diagrams may have difficulties identifying zones correctly if complex diagrams which have small zones with a lot of bounding curves, or if zones are complex spatial regions (e.g. in the general case they can be disconnected or non-simple regions) are constructed. Therefore, we assist in helping a reader to understand a categorisation by providing the possibility of querying a zone: by selecting a point p within any zone using the eyedropper the entire zone containing p is highlighted, and the zone descriptor displayed in the status bar (cf. Figure 2 top-right).

To aid user navigation within larger diagrams, we allow the usual scrollbars for horizontal and vertical directions, but we also provide spatial zooming: this is accessed via a select zoom option and mouse wheel scroll in and out, providing a larger working area when needed. This facility is likely to be useful in cases when users are working with a lot of categories, or when there are a large number of items in an intersection of

categories and one needs to enlarge the working space (i.e. to enlarge the region that is the spatial palette under consideration).

Item management. We utilise a drag and drop protocol for resource classification, simply placing items within the relevant region of the diagram. Item properties are visualized through tool tip text while the status bar, at the bottom of the interface, provides its current categorization (cf. Figure 2 bottom-left). Each item can be interactively repositioned and consequently re-classified. In order to not restrict the selection (and hence the functional application) of items to those sets of items corresponding to entire zones we also allow the selection of multiple items via a common box-selection (cf. Figure 2 bottom-middle/right).
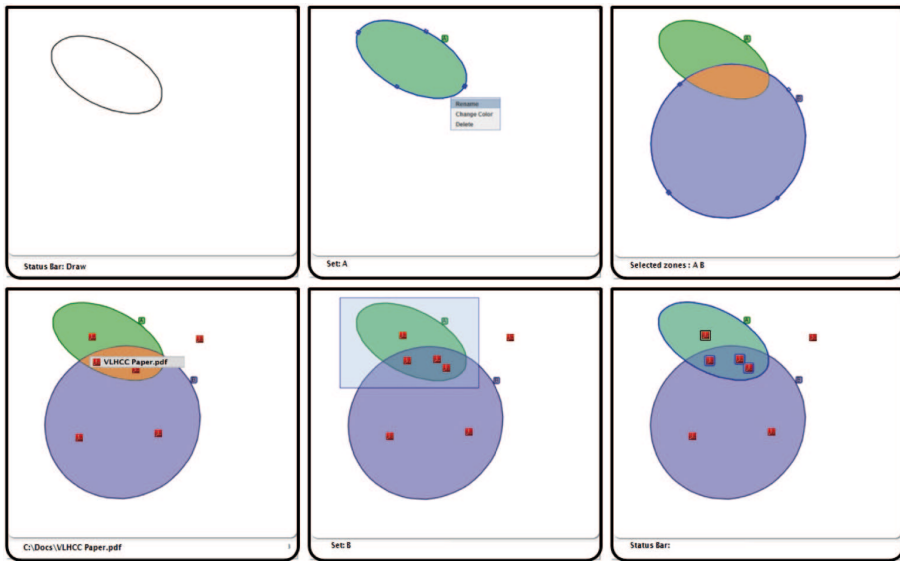

Fig. 2. A sequence of screenshots demonstrating FunEuler basic functionalities.

3.2 Operational functionality

At the core of the FunEuler functionalities is the ability to select and apply one, or more, *operations* to regions of the diagram. Performing operations on zones uses the familiar drag and drop: the user simply drags the operation icon onto the desired zone of the diagram. To perform operations on multiple zones (or similarly for other selected sets of items), the user must first perform a selection (by using the eyedropper) on all of the zones that he/she wishes to apply the operation to, and then drag the operation icon onto one of the selected zones.

To demonstrate the principal of functional application within the Euler diagram structure we developed several operations including: zip (which individually zips all

files in a region, creating a collection of zip files), ZipToSingle (which zips all files in a region to a single zip file), createplaylist (which creates a playlist when applied to a collection of mp3 files) and search (described below).

Tagging and Searching. The Euler diagram based interfaces commonly have a means to export to a tree based file system using conjunction of categories as node labels. However, we provide some specialised tag-based support for this interface. In general item tagging can be performed by adding information "inside" the item (e.g. utilising the genre field for mp3 files or tag data for office documents) or "outside" by utilising an item tagging database.

Now, if diagrams contain a large number of items then the user's ability to find and use specific items may be adversely affected. Therefore we offer the user the option of hiding tagged items, so that a user can hide items classified within the zone, thereby controlling what is actually visualised. Then, users can display tagged items by using a specific Search functionality, which displays all of the items whose tags correspond to the appropriate intersection of categories. We have developed a simple yet powerful Search functionality that relies on the Windows Search feature (which was introduced with Windows Vista/Seven) which allows the user to query, using an SQL-like language, the SistemIndex catalogue of Windows machines; since we offer support for this kind of operation we have access to functions for searching and modifying tags.

Developing new functions. FunEuler is designed to let the user to program to extend its functionalities. The very essence of the operational application within the interface is allowing users to assemble operations and this could be viewed as building a simple program or macro to support their tasks. The interface allows programmers to develop new operations by using their favourite scripting language (e.g. php, python, etc). Indeed, FunEuler provides access to two lists for the items and the zones that the operation is to be applied to: one list containing the pathnames of the items and the zones they belong to, and another list reporting the zones and the items that each zone contains. Each operation can produce zero, one or several files as result. As an example, the zip file operation produces a zipped file for each pathname in the input list, whilst the zip to single file operation produces one single zip file containing all of the files of the input list. On the other hand, the send by email operation does not produce any result files. Independent of the number of files produced the Results pane supports their representation in a special single-curve ED, where file icons are automatically placed within the curve representing the results set.

Closure. The application of an operation to a region of a diagram causes the generation of a new diagram in the Results part of the application window, and since the Results set is a diagram in its own right, operations can be applied to it generating a new Results diagram arising from the sequential application of the selected operations. Thus we can view the class of diagrams in FunEuler as being *closed under the application of operations*. Figure 3 shows the operations' icons within the Results set; in the left hand side the items are the result of one operation (create playlist on mp3s) whilst on the right they are the result of two operations (create playlist followed by zip). Subsequent operation icons are added consecutively, above the previous operations' icon. Due to

the closure we allow the user to drag the Results set onto the main diagram pane. This permits the user to categorise items which are the result of one of more operations alongside the original items that were already displayed in their diagram.
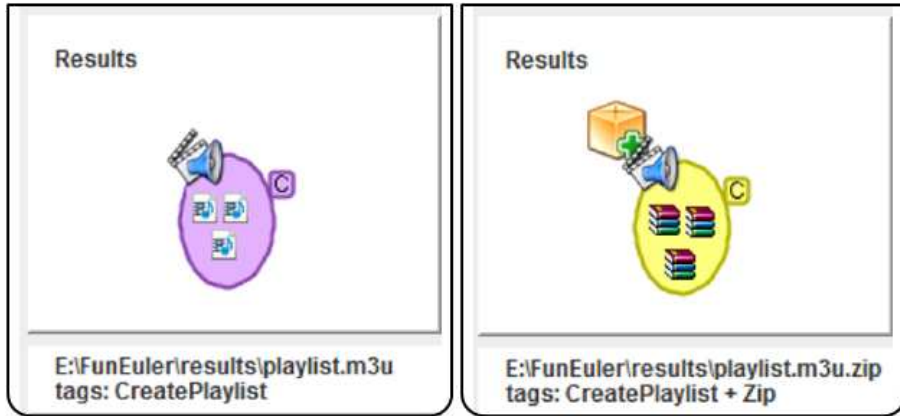

Fig. 3. Two screenshots showing the results of operations: (left) A single operation, *Create Playlist*; (right) Two sequential operations, *Create Playlist* and *Zip*.

3.3 Comparison with other tools

We decided not to try to compare the Euler diagram concept directly with other existing methodologies, because it is difficult to determine a suitable comparator and it also seems likely that tasks set would have a good chance of being biased towards one methodology. For example, a comparison of the Euler diagram concept for categorisation/resource management versus a hierarchical structure is likely to favour the Euler diagram concept for tasks that utilise overlapping categories, providing the underlying concept of Euler diagrams is not challenging for users. Furthermore, knowledge of users' level of comprehension and abilities with this underlying concept is a necessary precursor to allow us to accurately examine their abilities with the concept of the application of functions to regions of the diagram.

4   User testing

We decided to test users' conceptual understanding of the basic concept of Euler diagrams within this domain, together with their perception of its utility. Since we also wish to know if users can understand and apply functions, we also perform a basic test of this, keeping the tasks fairly straightforward in order to test the basic concept, with significantly more complex operational tasks being planned for future studies (as well as a more direct comparison within a more specific domain application that we have subsequently developed and is described in Section 5).

We test the concepts within the FunEuler tool we developed, which means we cannot really separate the conceptual effects of the actual representation from the effects of the particular tool, since the tool may restrict a user's actions and thus alter their behaviour; to acknowledge this, at the end of the tests, we asked users if they were aware of anything they would have liked to have been able to do but could not, either because of the representation or because of the tool. However, we believe these effect of the tool on the judgement of the representation will be minor in this instance due to the simplicity of the notation and the interface, whilst the tool based approach brings many benefits in that we can trace with the behaviour of the users in detail (e.g. the number of clicks, the number of corrections made during the test, etc. . .).

We describe our experimental design, some of the actual tasks set to the users, and a summary of the results of the user tests. We provide very high-level broad hypotheses below, but note that we are actually testing a restricted class of fairly small simple diagrams (no more than five curves or twelve zones), and as always any statement of results must really be restricted to the diagrams used and the tasks set within the experiment; we take this restriction for granted in the following. However, we are provided with an indication of the potential usefulness of the interface.

Hypotheses.

H1 Users can easily comprehend, create and modify Euler diagram based categorisations using FunEuler.
H2 Users are able to utilise the application of functions using FunEuler effectively.
H3 Users perceive FunEuler as an application they would find useful to improve their efficiency of operations.

The intention of the user testing is to investigate Hypotheses $H1$ and $H2$ using quantitative methods, by the recording of correctness of task completion, and efficiency in terms of timing, whilst we investigate the subjective questions related to Hypotheses $H3$ using qualitative methods. In order to test these hypothesis we carried out two different test sessions: test session 1 focused on $H1$ and $H3$, whilst test session 2 focussed on $H2$ and $H3$.

Participants. We recruited 19 volunteers, twelve of whom were male, aged between 21 and 31, who were a mix of undergraduate Computer Science students and Ph.D. students at the ISISLab, Università di Salerno. Three of these participated in pilot testing of successive versions of the experimental materials (e.g. the tasks and questionnaire). This allowed us to thoroughly check the experimental set-up, altering the language and providing more clarification of the tasks where necessary. The remaining sixteen participants were equally divided between two tests sessions (each of which were of approximately 25 minutes duration).

Test setting. In terms of the test conditions, both tests were carried out in a laboratory setting where each user had a dedicated laptop. Each test was explained orally and guided via a printed handout offering a short description of each of the stages. At the end of each step the user was asked to take a *snapshot* of the diagram, by clicking a special button in the application, in order to record both the duration of the last task and

the current state of the diagram. At the end of each session, the users were asked to fill in a questionnaire, consisting of 26 questions divided into 6 sections: general application, set creation, set manipulation, item categorisation, executing operations and perceived effectiveness of FunEuler in accelerating daily activities. These were measured on a Likert scale with score range $[-3, 3]$.

Experimental design. In the first test session each user was provided with a sequence of five stages, with each stage describing a basic concept or functionality of FunEuler. A task was presented to the user which required the utilisation of that functionality in order to modify the diagram, for instance. Thus we could check the users' comprehension of the induced modifications. Sample tasks included: reproduce a given diagram presented on paper utilising the tool, and to count the number of zones into which a given set was split. For each stage we recorded the completion time and we measured the percentage of the tasks performed correctly (e.g. the number of sets that were correctly represented, or the number of zone sets that were correctly counted). The second test session was aimed at assessing users' ease of comprehension and their application of operations to zones containing items. This test session consisted of three stages. The first stage was to perform item tagging in the similar manner to that of the first test, whilst the next two stages focussed on the application of a zip operation to certain zones of the diagram. We provide a brief description of the tasks below.

User Tasks. We consider user tasks divided into the following categories: Interpretation (given a diagram interpret its meaning), Development (given some categories and items construct a diagram) and Functional (utilise diagrams, applying functions). Note that, as mentioned earlier, we had decided it was a priority to test the fundamental aspects of the concept and the application and to leave the complex task categories like end-user development to a future date.

Table 1 show the basic results related to the tasks that comprised the two test sessions. We briefly describe the tasks and the criteria that we adopted to evaluate their execution, where necessary:

Diagram Construction To create a diagram containing certain specified zones. The presence of the given zones is checked.

Diagram Modification To modify a diagram by altering the sets represented (i.e modifying the curves which changes the set of zones present).

Zone Counting To count the number of zones contained belonging to certain set (i.e contained within a certain region).

Diagram Reproduction To reproduce a printed diagram. The presence of the given zones is checked.

Item Tagging To place eleven files onto zones of a given diagram. Each file is named according to the zone it should be placed in. The reason for this was to mimic the pre-existing knowledge of file content that a user would normally have at the time at which he/she is about to categorise a file. The correctness of the placement is checked.

Zip Items in one zone To apply the zip operation to files belonging to one zone which represented the intersection of two sets.

Zip Items in multiple zones To apply the zip operation to files belonging to multiple zones from a single set.
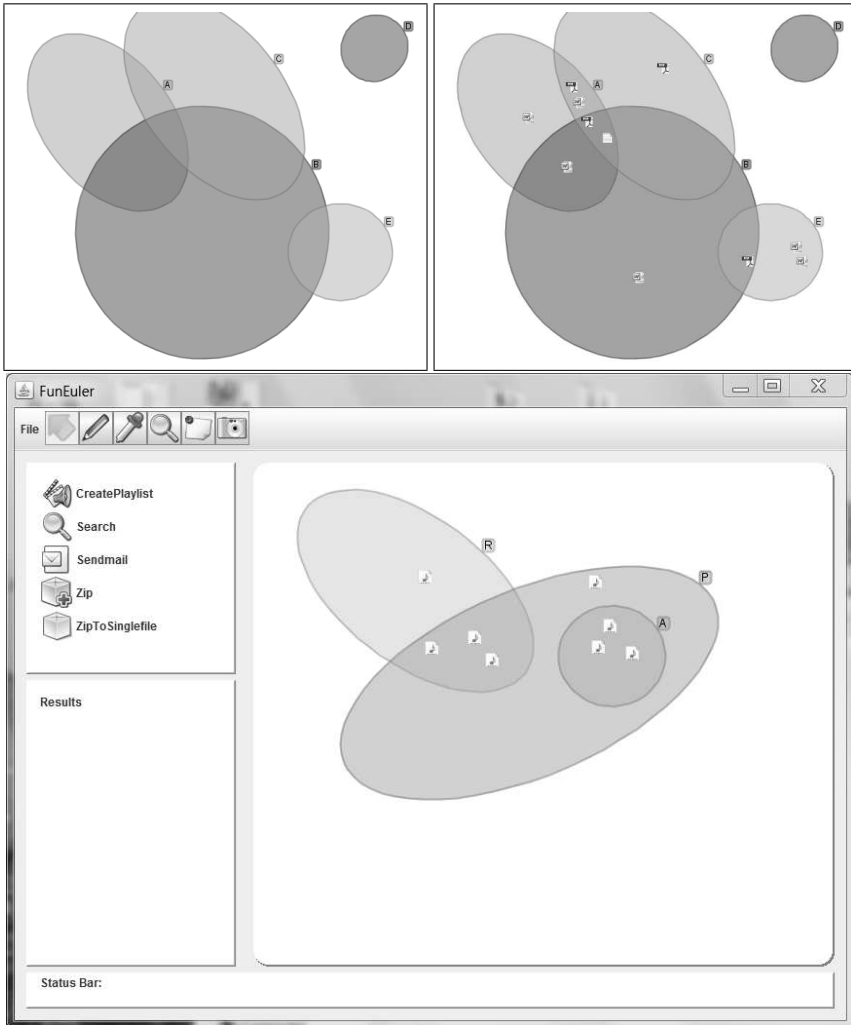

Fig. 4. Screenshots from one of the tests

In Figure 4 we present screenshots of diagrams constructed by users of the test. In the upper part the user has reproduced the diagram in Figure 1: on the left the diagram has no items, whilst on the right she had categorized 12 files, as requested, and a single file was misplaced (placed in E instead of D). In the lower part of the same figure we show a diagram used to categorize mp3 files.

4.1 Results

In Table 1 we present a summary of the results of the two test session that were carried out. Within the context of the diagrams and tasks used, Hypothesis $H1$ is supported by a high score on the tasks of Diagram Construction, Diagram Modification and Diagram Reproduction. Hypothesis $H2$ has been tested in tasks Zip Items in one zone and Zip items in multiple zones which also report $\approx 100\%$ success rates.

Task description	Correct answers	Completion time avg (min)	Task description	Correct answers	Completion time avg (min)
Diagram Construction	100%	6.8	Item Tagging	99%	3.8
Diagram Modification	100%	4.4	Zip Items in one zone	100%	4.3
Zone Counting	88%	3.3	Zip Items in multiple zones	97%	2.6
Diagram Reproduction	80%	3.0			
Item Tagging	90%	0.9			

Table 1. Tests results: (left) first test; (right) second test.

By comparing the two tests session results we could evaluate how the use of operations influenced the users' perceived usefulness of FunEuler: the average score was 0.13 for participants of the first test session but 1.88 for participants of the second test session (the score range is $[-3, 3]$). A one-way ANOVA was calculated on participants' ratings of the question "is FunEuler useful for repetitive tasks ?". The analysis was significant, with $F_{1,14} = 5.0$, $p = 0.04$.

The accurate responses to the tasks indicate that the underlying conceptual use of Euler diagrams for categorisation was easily grasped by users. The functional application utilising this structure did not appear to be a large conceptual step for them. The perceived usefulness of the FunEuler application was significantly enhanced by the utility for functional application. Users also reported that they would use the application in the future, suggesting areas where it would be useful, and some even stated that they would be interested in developing the associated functionality themselves.

5 FunEuler specialisation with mp3 files

FunEuler can be viewed as a flexible framework on which to host services based upon an Euler Diagram representation of information. As an example of this flexibility, and as an extension of FunEuler following the experiment, we developed a prototypical mp3 files management system. Since the music domain is of interest to the student participants, and since there are applications available to deal with mp3 files, we focussed on this specialisation area.

The diffusion of digital music has radically changed the manner of listening to music for the average user, and one fundamental step toward this evolution is the widespread adoption of the mp3 file format [10]. mp3 files are commonly created by the so-called *ripping* of CDs: CD audio tracks are digitally recorded by using a computer and compressing them into an mp3 file. This process is straightforward and a number of applications can perform it as well as attempt to fill the ID3 fields (identification for mp3) by accessing online archives (e.g. http://www.gracenote.com). A second, and perhaps the most relevant, method to obtain mp3 files is by buying them from online music stores (e.g. iTunes, Amazon), which usually provide correctly filled in ID3 fields. Despite these two widespread methods to obtain music, there is a large number of mp3 files that are not tagged at all: we sampled more than 8000 mp3 files available on our students' accounts, by asking them to run a small application that calculates a statistic on the content of the field Genre in each mp3, and we found that around 75% of them had empty or bad (containing random characters) Genre tags. Thus we inferred that the Genre field would be a suitable testbed for FunEuler tagging capabilities.

New functionalities. We implemented the operations of searching mp3 files by genre and tagging mp3 files. By combining the use of these operations it is possible to categorise large archives of mp3 files using few mouse clicks. The idea is to implement a cycle of *search-categorise-tag*: search for untagged files, categorise them by dragging and dropping their icons onto the zone representing the correct genre, and tag files by applying the tagging operation. The search for untagged mp3 files can be obtained by applying the Search operation to the "Universe" zone, and in this specialisation, the number of untagged mp3 files is displayed within the zone; clicking on the number causes a window containing the list of files to be opened. It is now possible to drag and drop the mp3 file icons onto the desired zone of the diagram in order to categorise them. The tagging operation assigns the set of tags associated with the zone containing a file to the Genre ID3 field.

6   Conclusion & Future works

 We have brought together concepts of visual non-hierarchical classification, spatial arrangements and functional application in the development of a novel Euler diagram based interface called FunEuler. The prototype application developed was utilised in user testing, which investigated if the Euler diagram concept was easily comprehensible for resource categorisation purposes, if the application of functions could be performed successfully, and if users perceived this idea to be useful either in its current state or together with future functionalities. The results were extremely encouraging.

The FunEuler interface also provides a firm base for future developments in many directions. For example, we could extend the power of the labels of curves, allowing a mixture of user defined categories (or tags) such as "Computational Geometry" and pre-determined types such as TeX files, or to allow functional labels such as "Before this year", which would filter for files based on the timestamp. Allowing any Boolean expressions on labels would assist with scalability of the diagrams since more information could be encapsulated in the labels instead of via the addition of more curves. A useful option could be to allow users to select from a set of curves in a library (e.g. files

modified yesterday, PDF files, etc). In terms of repetitive tasks, we observe that choosing specific groups of contacts for sending email to is a task that a user often needs to perform several times a day, and so enabling a fast mechanism to realise group selection as well as to facilitate repeated usage of related group selections could be of great assistance to end users, and we will extend an Euler diagram interface for such tasks.

Furthermore, there are many potential application areas within which the developed concept and interface could be applied. For instance, we briefly reported on the specialisation for `mp3` files and we plan to run future user testing within this domain. Secondly, in the workflow application area (e.g. see Alfresco), we could allow the drag and drop of a `TeX` document into a particular region, automate the running of macros such as PDFTeX and observe the results appear automatically elsewhere as an item in the PDF category. Users could then understand and track workflow involving multiple people on different systems. Thirdly, we will consider Euler Diagram Programming, where we will assign behaviour to curves instead of to the zones. For example one might have "Hotel" and "Food" categories with different "print" functions but their intersection might make use of both functionalities, although one must deal with the usual multiple inheritance problems, of course (e.g. we do not want to print off a date twice due to its appearance in each categories' behaviour).

One weakness of the representation is the need for continuous navigation in order to find specific sets/regions, particularly in larger diagrams. Whilst the representation may be an effective way of leveraging the spatial memory of the user, performance of actions by hand without specific automated support from the application may be annoying. A deeper investigation into the advantages and disadvantages of the representation, and exploration of potential tasks and application areas is a next step.

A wider investigation of the usability of EDs in a general context is required. There are many unanswered questions, such as: how large (in terms of the number of zones and number of curves) can a diagram be to be effectively manipulated by an average user; how well does the representation scale with the increasing of the complexity of the diagram; what effect does the representation have on user behaviour? Such questions require specifically targeted testing.

References

1. Funeuler. `http://www.isislab.it/projects/FunEuler/`.

2. F. Benoy and P. Rodgers. Evaluating the comprehension of euler diagrams. In *Proceedings of IV 2007*, pages 771–780, 2007.

3. Paolo Bottoni, Gennaro Cordasco, Rosario De Chiara, Andrew Fish, and Vittorio Scarano. Personalised resource categorisation using euler diagrams. In Maria Costabile, Yvonne Dittrich, Gerhard Fischer, and Antonio Piccinno, editors, *End-User Development*, volume 6654 of *Lecture Notes in Computer Science*, pages 251–257. Springer Berlin / Heidelberg, 2011.

4. Andy Cockburn. Revisiting 2d vs 3d implications on spatial memory. In *AUIC '04: Proceedings of the fifth conference on Australasian user interface*, pages 25–31, Darlinghurst, Australia, Australia, 2004. Australian Computer Society, Inc.

5. G. Cordasco, R.De Chiara, and A. Fish. Interactive visual classification with euler diagrams. In *Proceedings of the IEEE Symposium on Visual Languages and Human-Centric Computing VL/HCC 2009*, pages 185–192. IEEE Press, 2009.

6. A Fish, B. Khazaei, and C. Roast. Exploring human factors in formal diagram usage. In *Proceedings of Engineering Interactive Systems conference 2007*, LNCS 4940, pages 432–448, 2008.

7. Andrew Fish, Babak Khazaei, and Chris Roast. User-comprehension of euler diagrams. *Journal of Visual Languages & Computing*, 22(5):340 – 354, 2011.

8. J. Flower, A. Fish, and J. Howse. Euler diagram generation. *Journal of Visual Languages and Computing*, 19:675–694, 2008.

9. J. Howse, G. Stapleton, and J. Taylor. Spider diagrams. *LMS Journal of Computation and Mathematics*, 8:145–194, 2005.

10. ISO/IEC 11172-3:1994. *Information technology – Coding of moving pictures and associated audio for digital storage media at up to about 1,5 Mbit/s – Part 3: Audio.* ISO, Geneva, Switzerland.

11. A. Verroust-Blondet. J. Thièvre, M. Viaud. Using euler diagrams in traditional library environments. In *Proceedings of Euler Diagrams 2004*, pages 189–202. Electronic Notes in Theoretical Computer Science, 2005.

12. Akrivi Katifori, George Lepouras, Alan Dix, and Azrina Kamaruddin. Evaluating the significance of the desktop area in everyday computer use. *International Conference on Advances in Computer-Human Interaction*, pages 31–38, 2008.

13. L. Langer and E. Frokjaer. Improving web search transparency by using a venn diagram interface. In *Proceedings of NordiCHI 2008*, pages 249–256. ACM Press, 2008.

14. Maarten, Mary P. Czerwinski, Maarten Van Dantzich, George Robertson, and Hunter Hoffman. The contribution of thumbnail image, mouse-over text and spatial location memory to web page retrieval in 3d. pages 163–170. Press, 1999.

15. R.De Chiara, U. Erra, and V. Scarano. VennFS: a Venn Diagram File Manager. In *Proc. of the Seventh International Conference on Information Visualisation, IV 2003, 16-18 July 2003, London, UK*, pages 120–126. IEEE Computer Society Press, 2003.

16. R.De Chiara, A. Fish, and S. Ruocco. Eulr: a novel resource tagging facility integrated with Flickr. In *Proceedings of the AVI 2008 Advanced Visual Interfaces Conference*, pages 326–330. ACM Press, 2008.

17. F. Ruskey. A survey of Venn diagrams. *Electronic Journal of Combinatorics*, 2005. www.combinatorics.org/Surveys/ds5/VennEJC.html.

18. A. Shimojima and Y. Katagiri. An eye-tracking study of exploitations of spatial constraints in diagrammatic reasoning. In *Diagrams 2008*, LNAI, pages 74–88, 2008.