Mohamedally, D., Zaphiris, P., Petrie, H. A Web Based Tool for HCI-Orientated Massive Asynchronous Linear Card Sorting. In the Proceedings of British HCI Conference. Bath, UK 8-12 September 2003, Vol. 2, pp. 99-103.

A WEB BASED TOOL FOR HCI-ORIENTATED MASSIVELY ASYNCHRONOUS LINEAR CARD SORTING

Dean Mohamedally
Centre for HCI Design
City University
London EC1V 0HB, UK
cp496@soi.city.ac.uk

Panayiotis Zaphiris,
Centre for HCI Design
City University
London EC1V 0HB, UK
zaphiri@soi.city.ac.uk

Helen Petrie
Centre for HCI Design
City University
London EC1V 0HB, UK
h.l.petrie@city.ac.uk

ABSTRACT

This paper presents an initial iterative design and implementation of a web based linear card sorting tool. Card sorting is a participant based knowledge elicitation technique for grouping information into categorical domains. We have constructed a tool that HCI specialists and others can utilise to gather data from demographic groups. This would allow large numbers of participants to take part in card sorting experiments asynchronously (in their own time) via using a web browser. The advantages of using this tool as part of a card sorting technique are firstly the possibility of aggregation of data from particular demographical arenas on a wide scale and secondly the fact that the output data is in a format that is easily parsable by a cluster analysis tool. To better evaluate its design and implementation a pilot test was conducted with HCI specialists as the target group.

Keywords

Card sorting, prototyping, design, Java.

1. INTRODUCTION

Card sorting is a participant based knowledge elicitation technique for grouping information into categorical domains. It is useful to HCI specialists as a technique for analysing categorisations of domains of knowledge [5]. Most current card sorting is done by having several cards of information placed into groups by a participant or groups of participants. These groups are then given a category name based on the experiment's requirements [2]. The demographic characteristics of participants chosen to take part in card sorting experiments influence the output quality of the categorisations. For example expert categorisations of problem domains are often favoured over novice categorisations [4].

Although traditional card sorting is primarily associated with non-linear categorisations of items, in certain cases categorisations without comparison biases are necessary. This is especially important in factoring categories of new and relatively inexperienced domains of knowledge as well as mixing confident expert participants with non-expert participants. For such cases linear card sorting might be a better alternative.

In a linear card sorting experiment, as devised by the authors, the participants' first evaluation of each card they come across is what matters most in order to aggregate the categories that they form without comparison bias. However they can view prior cards in categories already existing to remind them of what they have placed where, so that a history is constructed.

Once a card is sorted into a category, whether into a new category or an existing one, the category can be renamed but it cannot be moved or deleted. Progression through the different cards is done in a purely linear fashion. It is suggested that participants should not undertake the linear card sorting in communicable groups as it is their own initial judgement that is to be considered without bias from others.

Fig 1: a practical non-linear card sorting in progress

On the other hand, in a non-linear card sorting experiment as depicted in Fig. 1, the participants may read any of the cards at any time, before or after being sorted into groupings. They are usually encouraged to participate in small groups and communicate with each other to explain their card groupings. More importantly they can de-allocate cards from one category group and move them dynamically to others. This is functionally more appropriate for quality categorisation in expert domains of knowledge [3]. However this rationale can lead to a tendency for participants to bias their sorting by spending

more time comparing certain cards rather than others for a number of reasons e.g. number of cards, popularity and level of awareness of the categories, group communication, sometimes even position and verboseness of the category.

It is thus evident that both linear and non-linear card sorting techniques have advantages and disadvantages which need to be further investigated.

2. SPECIFICATION REQUIREMENTS

Although the described tool can be used for a variety of domains, the original requirements of this system came from a taxonomy review of the current state of the art in mobile HCI research [6] by reviewing literature from five years of conference proceedings [7] and the tool will be use in subsequent experiments to further refine our findings..

In general however the requirements presented us with an opportunity for providing a technique that would raise the quality of the categorisation process in card sorting by allowing for large numbers of participants to take part. Card sorting is usually undertaken by relatively small groups of participants. The larger the number of participants, the more refined the categories will become [5]. We therefore decided to construct a web based online tool that would allow participants to log into an URL, do the card sorting via a web page applet and then send the results back for analysis.

Since we are interested in non expert categorisation we chose to implement our devised linear card sorting method. In addition to this, our original taxonomy review compared 107 papers. Most non-linear card sorting is done with considerably less cards in order to reduce time constraints and sorting complexity. To facilitate the categorisation of so many cards would be a fruitless exercise if deployed as non-linear card sorting to public non-expert groups. We believe that linear card sorting is easier and faster to complete than non-linear card sorting.

Our specification required us to construct a simple to use tool that would allow HCI specialists and others to aggregate grouping data from large numbers of participants via the internet without them having to be physically present or monitored. The fundamental system requirements of our tool when compared with other existing card-sorting tools was that there should be minimal installation for both the test administrator and the participant users, and also the data received from our tool should be sent in a format ready for cluster analysis in a secondary statistical tool.

Cluster analysis is a powerful technique for methodically processing the strategic groupings of information from a number of sources (e.g. participants' data) by calculation of perceptual strengths between pairs of cards. Many statistical packages such as SPSS™ and Statistica™ will convert category datasets into matrices of similarity or distance scores. However this is not an automated process, and if we wish to gather large amounts of participant data as in our case it would require extreme hours of manual input labour. The cluster analysis format we chose to utilise was the EZSort package (EZCalc utility in particular) from IBM [1]. EZCalc is a particularly effective and simple to operate utility for the generation of cluster analysis trees once card sorted datasets are imported in the correct format. EZSort comes complete with its own card sorting implementation as a standalone tool named USort. However like some other card sorting tools, it is not a web based system and therefore requires full user installation with administrator privileges to attain its user data, which many potential participants may not find appealing. Dealing with such a potentially large amount of participant's data our specification was determined to be the best solution.

3. SYSTEM DESIGN AND IMPLEMENTATION

We decided to implement our card sorting tool in a language that would be as compatible as possible to the majority of Internet users whilst still maintaining enough cross platform facilities to ensure successful functionality upon deployment. In a crunch, it came down to Java 1.1 applets Vs Macromedia Flash. We felt that Macromedia Flash is a truly versatile authoring environment with advanced graphics and animation support and we could have accomplished the majority of requirements eventually through Flash scripting. However we believe that for the first fully functional prototype Java would enable us to better manage our resources and speed up development time by utilising a Java based rapid application development (RAD) tool. It also enabled us to incorporate functionality directly such as the transmission of participants' data without utilizing third party plugins or scripts and outputting the required format for EZSort's EZCalc Cluster Analysis utility. However Macromedia Flash may still yet be considered for future implementations as its technology grows more robust with every incarnation.

Java 1.2+ was not preferred, even though advanced drag and drop facilities via its Swing libraries could have been useful in a card sorting simulation scenario. Deployment issues with participants having to download upgrades to their browser to run the applet was not an option for a proposed simple applet test. We wanted the most readily available edition and since current 1.2 enabled browsers are backwards compatible, 1.1 was seen as the best option. Other languages will be evaluated for subsequent iterations of the system and it is already proposed that a server-side language be utilised such as Java Servlets to offload end user technical requirements.

The cards data themselves needed a format that would be easily maintainable by all foreseeable administrators of the experiment for any scenario they wish. As such, we have kept the cards in a simple text formatted file. It was suggested that XML should be utilised as the cards data file format so that other software utilities could be designed to securely interoperate with it but due to time constraints this is on the recommendations for future iterations.

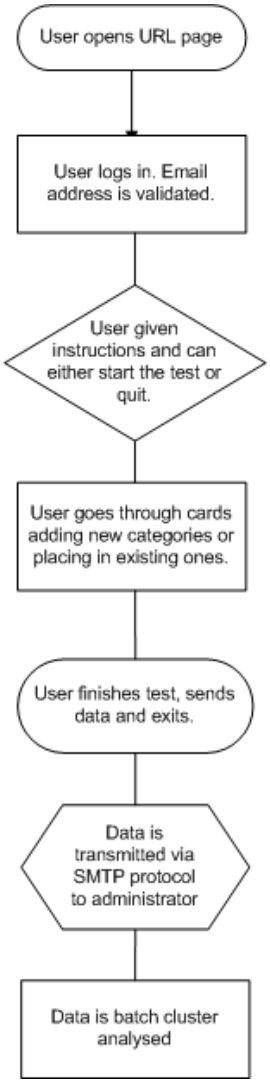

Fig 2: Simplified process sequence of operation

A simplified view of the operational architecture is shown in Fig 2. As you can see from Fig 2 an SMTP transfer mechanism is incorporated directly in the applet allowing it to email the users data itself as a micro email client rather than rely on third party email scripts or plugins. This allows future experiment administrators of our card sorting tool to specify direct mail servers for processing incoming test data as well as minimize the maintenance requirements of reconfiguring the tool for other card sorting tests.

Performance and user operations were also of a high concern to the authors. It was important to construct a robust architecture that could facilitate potentially large card sets but yet be operated on relatively low specification end user machines and also be used responsively fast with a relatively quick learning curve. For example:

- String optimisations were included throughout the system to increase the performance of GUI text display responsiveness.
- The card sorting algorithms were designed to utilise data vectors as linked lists for processing dynamic allocations of objects in memory and enabling the processing of potentially memory-finite amounts of cards to be sorted rather than any fixed amount of cards.
- The user interface of the main card viewing screen was designed with HCI principles in mind. For example minimising distance between user information and user actions, confirmation and error recognition as well as available help, colour coding important actions so as to be as comfortable as possible for as many possible demographics of participant users.

Fig 3 shows the main card viewing screen of our tool in action.

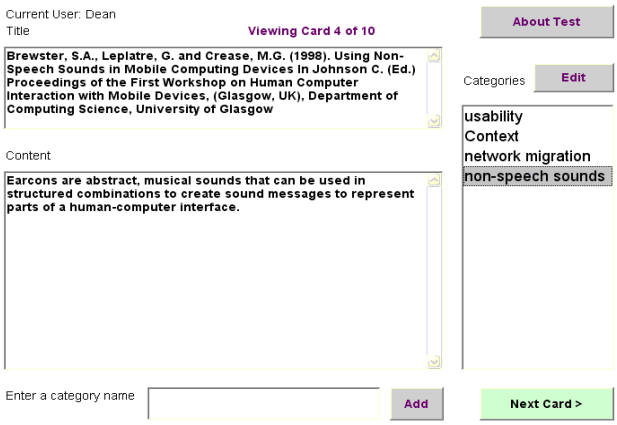

Fig 3: Card viewing screenshot

4. PILOT TEST

To prepare the web tool for deployment we have undertaken a pilot test with six HCI specialists in our department and one external engineering specialist.

In this pilot experiment participants were asked to categorize mobile HCI literature.

First a pre-test questionnaire was presented to the participants in order to:

- Evaluate their current understanding of card sorting
- Present them with our linear card sorting technique
- Engage them in initial thoughts on categorisation of the field under test

Then, we gave them a printed short description which referred to the same material as found in the web tool's online help documentation to verify whether they did require assistance beyond that which the actual tool could provide.

After answering any clarification questions they raised, they were asked to use the tool to perform a card sorting task. Throughout the experiment they could raise any points of interest or complaints. Finally a post-test questionnaire was given to collect information about their general impressions about the tool and any modifications they thought were necessary. A cluster analysis graph of their combined categorical formations of ten sample cards is shown here in Fig 4.

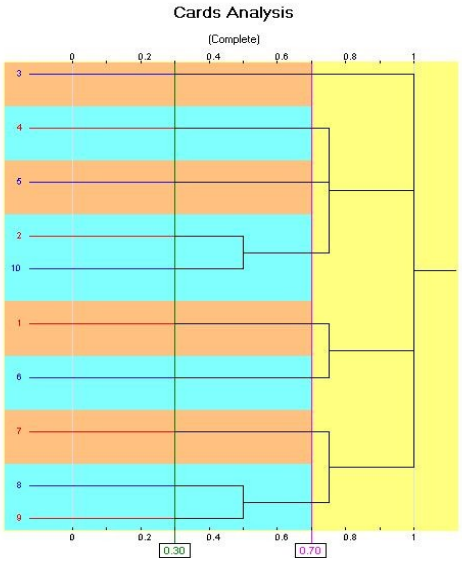

Fig 4: Cluster Analysis of the pilot test's data with IBM EZSort

5. FINDINGS AND CONCLUSIONS

The pilot study was very effective at pointing out minor GUI arrangement issues, which was consistent to all of the HCI expert participants. Other than that, most of the participants completed the test faster than expected. It was originally planned to take them 12-15 minutes to complete it but they ranged from 6-13 minutes. Aside from their comments and commendations on its design and simplicity of use, the only issue some of them had was that they were expecting non-linear card sorting and not this particular technique of card sorting regardless of their expertise. Thus subsequent versions may intend to develop both techniques of online card sorting further, but for this first version it has been found to be functionally accurate and almost ready to deploy in its first scenario.

The ability to utilise IBM's EZSort cluster analysis tool has shown to be very effective and it is anticipated that the results from the deployed tool will provide interesting insights into the categorisation process in card sorting.

6. REFERENCES

[1] IBM EZSort (USort and EZCalc). www-3.ibm.com/ibm/easy/eou_ext.nsf/Publish/410

[2] Kurniawan, S. & Zaphiris, P. (2003) Web Health Information Architecture For Older Users. IT & Society, 1 (3) pp. 42-63

[3] Maiden, N.A.M & Hare, M. (1998) Problem domain categories in requirements engineering. Article No. hc980206 International Journal of Human Computer Studies. Issue 49, pp281-304.

[4] Maiden, N.A.M. Mistry, P, & Sutcliffe, A.G. (1995).How people categorise requirements for reuse: a natural approach. In Proceedings of the 2nd IEEE Symposium on Requirements Engineering, pp148-155.

[5] Martin, S. (1999) Cluster Analysis for Web Site Organization. Internetworking 2(3), www.internettg.org/newsletter/dec99/cluster_analysis.html.

[6] Mohamedally, D., Zaphiris, P., Petrie, H.L. (2003). Recent research in mobile computing: A review and taxonomy of HCI issues. In Proceedings of HCI International 2003, Crete.

[7] Proceedings of the ACM Mobile HCI (1998-2002), ACM Press.