Spreadsheets as User Interfaces

Alan Dix
University of Birmingham,
Birmingham, B15 2TT, UK
and Talis, 48 Frederick Street,
Birmingham B1 3HN, UK
alan@hcibook.com

Rachel Cowgill
University of Huddersfield
Queensgate,
Huddersfield, HD1 3DH, UK
r.e.cowgill@hud.ac.uk

Christina Bashford
University of Illinois at Urbana-Champaign
1114 W. Nevada Street
Urbana, IL 61801, USA
bashford@illinois.edu

Simon McVeigh
Goldsmiths, University of London
New Cross
London SE14 6NW, UK
S.McVeigh@gold.ac.uk

Rupert Ridgewell
British Library
96 Euston Road
London, NW1 2DB, UK
Rupert.Ridgewell@bl.uk

http://alandix.com/academic/papers/avi2016-spreadsheet

ABSTRACT
Spreadsheets are ubiquitous, familiar, often overlooked, and embody vast financial and human investment, not least in their user interface. This paper shows how spreadsheets can be used as an integral part of interactive processes, for activities from simple data entry, to more complex grouping and linking of datasets, both as fully functional prototypes and as part of a final system. They reveal artful digital and physical end-user appropriation; exemplify key design principles including 'appropriate intelligence', ensuring 'smart' technology fits the complete human–computer process; and expose further design issues such as the importance of 'exception' sets.

CCS Concepts
Applied Computing – *performing arts, digital libraries and archives*; Information Systems – *data provenance*; **Human-Centered Computing** – *interaction design*

Keywords
Spreadsheets; appropriation; musicology; digital humanities

1. INTRODUCTION
As researchers of user interfaces we want to create novel interface features, visual, audio, haptic or even bodily interactions that astound and extend the capability of what is possible. Of course, as HCI researchers, we also know that the most effective *interaction* is not necessarily the most innovative interface, and certainly not the most complex.

This paper is about spreadsheets, a technology so ubiquitous and yet so mundane that it is almost invisible, forming a backdrop to day-to-day working life. However, we will see that spreadsheets can be used as part of complex or novel interaction workflows, effectively becoming a part of the 'interface' in the same way that a web form does in a web application.

AVI '16, June 07 - 10, 2016, Bari, Italy

ACM 978-1-4503-4131-8/16/06···$15.00
DOI: http://dx.doi.org/10.1145/2909132.2909271

Although there are innovate uses of the computational nature of a spreadsheet, we concentrate here on its most basic use as an editor of tabular data. This is important as there are users, who, while expert in their own domain, find the numeric or computational aspects complex or intimidating, not without justification.

This work is partly an analysis of existing systems, some of which will be familiar to any academic. It is also partly a reflection of the use of the 'spreadsheet as interface' within the InConcert project (http://inconcert.datatodata.com), which is investigating and re-imagining the digital archive within musicology [13]. However, this is fundamentally a paper about the *design and engineering* of user interaction, particularly in an innovative research setting. This may be as:

(i) a prototype or technology probe, that helps us understand the requirements of a final (possibly table-oriented) bespoke web or desktop system;

(ii) a full solution to a temporary problem, such as during data cleaning and preparation

(iii) a final system that may be deployed long term

Each of the more innovative applications we consider has alternative research or commercial systems that could have been used instead. However, we argue that spreadsheets are not simply 'good enough' tools, but often the *best tools* to use, for a variety of reasons including:

(i) familiarity to users reducing barriers to use and training time

(ii) rapid development and turnaround allowing fully-fledged interactions to be created in low-budget or time-constrained situations, and also reducing premature commitment

(iii) leveraging years of existing user interface development, often better than can be created in a bespoke system

While much has been written about the design and use of spreadsheets, and many spreadsheet-like systems have been created in research and commercial settings. We know of no other work that addresses the issue as a deliberate tool in the design and engineering of interactive systems.

In the rest of this paper, we briefly review related literature, describe some examples and use these to expose issues and design principles for the use of spreadsheets as user interfaces.

Further links, reports and images can be found on this paper's webpage (see above).

2. RELATED WORK

Spreadsheets feature in some of the earliest HCI and end-user programming literature, and VisiCalc is seen as an archetype of successful user-driven software development. In 1984, Alan Kay positioned spreadsheets at the top of the scale of programming languages, as an 'ultra-high level language' above Smalltalk and Prolog [18]. Early ethnographic studies by Nardi and Miller [23] showed that this flexibility and end-user programming ability were used creatively and collaboratively, often more expert users creating formulae and frameworks for colleagues.

However, despite (or because of) the power and flexibility of spreadsheets, they also have many problems, and an extensive literature has developed around these, particularly the potential for hidden errors. One study found that 44% of spreadsheets contained errors, despite expert users feeling "*quite confident that their spreadsheets were accurate*" [8], and another study found that only half of spreadsheet errors were detected [15]. The properties and problems of spreadsheet programming were also one of the early applications of Green's cognitive dimensions [17]. While various tools and techniques have been proposed to help with spreadsheet authoring [30, 25], this is by no means a solved problem. Indeed, the European Spreadsheet Risks Interest Group runs an annual conference solely on this topic (http://www.eusprig.org).

In addition to extensions focusing on the programming power of spreadsheets, there are many spreadsheet-like table editing and visualising interfaces. As well as web spreadsheets, many data organisation applications have table-based interfaces that are spreadsheet-like. This includes classic PC databases such as Access; web tools, such as Google Fusion Tables; and many visualisation and data analysis tools. Table-based views are included as one of, or even the main, visualisation even where data is clearly graph-based, such as ontologies (e.g. Protégé, http://protege.stanford.edu) or RDF (e.g. Tabulator [4]).

In the information systems literature, the use of spreadsheets alongside 'official' systems has traditionally been seen as negative, undermining the integrity of centralised data. However, more recent literature has begin to recognise the value of these, so called, 'shadow systems' or 'shadow tools' [7]

3. SPREADSHEETS FOR UPDATE

Because of their programming power, spreadsheets or spreadsheet extensions have been used for prototypes and full application building; for example action-effect rules, a form of user-interface specification [22], or, more recently, Gneiss, a spreadsheet tool for building streaming web data applications [9]. More prosaically, CSV has become the *lingua franca* of data generally and of Open Data in particular – so much so that the Open Data Institute blog declared 2014 the "year of CSV" [29].

CSV or Excel spreadsheets are also used as the means for updating data. Many university management systems create *pro forma* spreadsheets listing students on a course, so that academics or administrators can fill in the marks and then re-upload the data into the central system. The same type of system was used as part of REF, the UK's periodic evaluation of university research. Assessors downloaded personalised Excel spreadsheets listing the papers allocated to the reviewer with summary information (title, venue, etc.) and blanks for the assessments and comments. These were periodically uploaded to keep the central computer system up-to-date and allow grades from all reviewers to be combined.

At first these uses of the spreadsheet for update may seem crude or even suggest laziness on the part of the developers. However, anyone who has used an online university mark-entry system will know this is far from the case. Bespoke interfaces, however well designed, need to be documented and learnt. This is especially problematic when they are only used occasionally. In contrast spreadsheets are familiar and relatively simple.

Furthermore, while far from perfect, there has been enormous financial investment in the user interface of major commercial spreadsheets such as Excel, Numbers and Google Sheets, and similar levels of human investment in open-source equivalents such as Open Office. Using the spreadsheet as the user interface leverages that investment and can create a far better user experience than would otherwise be possible within budget.

design lessons: leverage past UI development, simple can be best

4. SPREADSHEETS IN INCONCERT

InConcert is reimagining the process of digital archive creation and use in the humanities, focusing on a number of specific musicology datasets, in a variety of formats, but all concerning concerts in London from 1750 to the early 20th Century [13, 21, 3, 10].

4.1 Primary Data

The oldest data set had been through several legacy database since the early 1990s and its current form consists of a number of CSV files. An online version was to be created using a NoSQL database with the aim of creating both a web user interface and also data APIs in JSON and RDF. The normal approach would be to tidy the CSV with a data-cleaning tool, if necessary, and then import into the data base; effectively discarding the initial CSV,

However, while it would be possible to re-export a CSV version, the current CSVs are 'owned' by the musicologist. The authoritative nature of the data is a core value of the historical scholar; so it was crucial that, as the data was imported, the original files were preserved as the 'golden copy'. Furthermore, if there were updates they would happen to this authoritative copy, and so the updated data would need to be merged into the online copy. For some data, unique ids had to be added to the CSV files, but otherwise they were left as developed by the domain expert.

In addition, some of the fields were not 'normalised'. For example, the 'advert' field could contain a single newspaper abbreviation (e.g. 'Ti' for 'The Times'), but could contain more details (e.g. 'PA 3 Apr'). Most complex was a 'programme' field, which describes the complete programme of the concert including works and performers. These semi-structured fields need to be parsed, but importantly in a way that could be reliably recreated if the original data was updated. Traditional data cleaning would lose this connection, so instead a number of 'exception lists' were created to deal with elements that did not parse easily.

design lessons: understanding user values , preserve the 'golden copy' as live source; value of complex fields and exception lists

4.2 Matching and Grouping

Each dataset has its own 'authority files' listing names of people and places. In order to be able to cross-link the different datasets, the names had to be matched. Of course, the same person's name my be expressed differently (e.g. 'Bach', 'Johann Sebastian Bach', Bach, J.S.') and the same name might refer to several people or places (e.g. 'Royal Theatre'). That is, we needed to do a level of entity matching *between* datasets. In addition, one of the datasets

contained records relating to sources: adverts or notices about concerts. Several sources might refer to the same concert. This was not simple entity matching as these were not the same source, but did refer to the same concert. That is, here we needed to do a form of entity matching *within* a dataset. Crucially, the identification of these historical entities required musicological expertise to ensure the authoritative nature of the relationships.

There is a substantial literature on entity/object identification dating back from paper records [14], to the early days of databases [1] and now semantic web applications [24]. Techniques vary from simple similarity measures to complex machine learning and structural relationships [26, 5, 16]. There is also tool support including OpenRefine (https://github.com/OpenRefine/), RELAIS [27], Trifacta (http://trifacta.com) and D-Dupe [6]. However, where these tools are publically available, they are highly functional, but complex, designed for the data scientist. For the domain experts to use these would require for more extensive training than was possible given the time limits of the project.

Instead, batch scripts performed relatively simple algorithms to determine *candidate* matches, which were then verified by the musicologists (*authoritative*). The precise form of automatic matching differed slightly, for person names, surnames were particularly important, but again with *exception lists* to deal with honorifics, non-human agents (e.g. orchestras), etc. For the concert source matching a liberal rule was used to do candidate grouping (ignoring time of day), with a conservative rule to create warnings of potential problems (e.g. 'evening' vs. '7pm').

The 'intelligent' algorithms followed the principles of *appropriate intelligence* [11], providing just sufficient cleverness to aid the complete human–computer system, maximising user expertise. Often 'human computation' [2] treats the human as a (sometimes unwitting) cog in the machine.

Expert verification was performed using specially exported spreadsheets. For cross-dataset name matching this consisted of spreadsheets each row of which listed a pair of names, one from each dataset, with summary details of each and a blank 'match' column for the expert judgement. The verification job required the musicologists simply to scan down the list filling in 'Y', 'N', or 'P' (for 'possibly'). For the within-dataset grouping, the spreadsheet consisted of a row for each source with blank rows between groups. It was decided that moving rows was potentially 'fragile' (too easy to select part of a row, or overwrite), so instead a verification column was used, but with extra *user-defined codes* for when the candidate groups was not correct ('Y1', 'Y2', etc.). In both cases the spreadsheets were re-imported to create online data.

design lessons: rich use of simple spreadsheets, user expertise, appropriate intelligence, fragility, user-defined codes

4.3 Anxiety and appropriation

Overall the process worked well, but one musicologist initially expressed concern about using the spreadsheets. It turned out that this was because of university financial spreadsheets in the past, which were very 'fragile'. Such spreadsheet anxiety has been described elsewhere [28] and is amply justified by the literature on spreadsheet errors. Happily, the initial misgivings were dispelled when it was explained that the spreadsheet was only being used as a table of data, not for formula calculations.

The musicologist who completed the grouping task liked the spreadsheet view, noting that it was reminiscent of a Paradox database used many years before (*familiarity*). The spreadsheet itself was updated as expected, with URL links used to interrogate the full online data when needed. However, the musicologist also printed out the spreadsheets, sticking them together, spreading them across a large table (fig. 1), and covering the paper copies in copious notes. While some notes were transcribed into a 'notes' field in the spreadsheet, others were left only on the paper copy as a record of the process that led to the decisions.

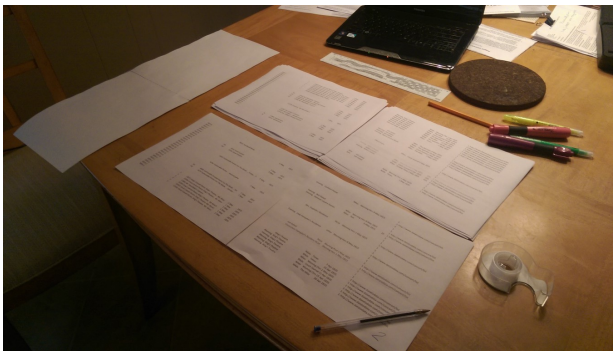

Figure 1. Printed spreadsheet for working (© C. Bashford)

The musicologist intended to archive the paper copies after the critical information and decisions had been transcribed into the electronic form. This is partly because of the additional paper notes tracing and evidencing the processes that led to the authoritative version, maintaining scholarly rigour, which is at the heart of the discipline of historical musicology However, there may also be a slight and reasonable distrust of electronic storage.

Within the project several datasets had to be retrieved from legacy and unsupported systems including old versions of Paradox and Access. Even SQL dumps were problematic as SQL has many proprietary variations. In contrast, paper notes from the same period were easily available in archive boxes. The musicologists are not alone in facing this problem, sometimes referred to as the 'digital dark ages' [20] and major archival institutions have projects to restore and future-proof past digital materials [19]. From an electronic storage point of view, this emphasises the importance of having archival copies in long-lasting formats. Despite some minor issues, CSV certainly has this property.

design lessons: familiarity, fragility, spreadsheet anxiety, physicality, longevity of format

4.4 Additional Lessons

As well as the physical appropriation in fig. 1, the user-defined methods for grouping emphasised the way appropriation design principles [12] apply to spreadsheets. Indeed, the use as a user interface component is a form of appropriation and by providing spreadsheets to people, they can perform their own appropriation.

During the project a presentation had to be prepared and to create graphs various CSV exports were created and combined. Crucially, this included adding fields, For example, venues were geocoded to allow spatial graphs and composers' countries of residence were added for the relevant period. Of course, such additional fields should ideally be fed back into the online datasets, both for the dataset owners and also third party users of the data. However, this form of crowd-sourcing raises issues of authority and requires methods to record and visualise the origin of updated or additional data.

design lessons: appropriation, extensible fields, crowdsourcing and provenance

5. SUMMARY

We have seen that the spreadsheet is far from inconsequential, functioning as a rich tabular interaction 'widget' in complex data manipulation workflows. While it would be possible to design dedicated matching, linking and data-update interfaces, the export and import of well-designed spreadsheets offers a familiar and flexible way to achieve high production quality in a timely and cost-effective way. We have summarised some of the technical and user interface lessons learnt, and seen a range of potential applications from simple update to data reconciliation. However, given the rich history of appropriation and invention, uses of the humble spreadsheet will continue to evolve and surprise.

Note that the complete development effort for InConcert (not just the elements reported here) was approximately 30 days. If the facets reported were built completely bespoke, each could have taken that long. The use of the spreadsheet as a user interface component was critical in making it possible to achieve our musicological and technological research goals.

6. ACKNOWLEDGEMENTS

In Concert (http://inconcert.datatodata.com) is a mini-project of *Transforming Musicology* (http://transforming-musicology.org), funded by the AHRC Digital Transformations in the Arts and Humanities scheme.

7. REFERENCES

[1] Ahmed, E. Ipeirotis, P. and Verykios, V. (2007). Duplicate Record Detection: A Survey. *IEEE Transactions on Knowledge and Data Engineering* 19 (1):1–16.

[2] von Ahn, L., Maurer, B., McMillen, C., Abraham, D. and M. Blum, M. (2008). reCAPTCHA: Human-Based Character Recognition via Web Security Measures. *Science*, 321(5895):1465–1468

[3] Bashford, C., Cowgill, R. and McVeigh, S. (2000). The Concert Life in Nineteenth-Century London Database, in *Nineteenth-Century British Music Studies*, 2, ed. by J. Dibble and B. Zon (Aldershot: Ashgate, 2000), 1-12.

[4] Berners-Lee, T., Chen, Y., Chilton, L., Connolly, D., Dhanaraj, R., Hollenbach, J., Lerer, A. and Sheets, D. (2006). Tabulator: Exploring and Analyzing linked data on the Semantic Web, *Proc. SWUI06*

[5] Bhattacharya, I. and Getoor, L. (2007). Collective entity resolution in relational data. *ACM Trans. Knowl. Discov. Data*, 1(1):5

[6] Bilgic, M., Licamele, L., Getoor, L. and Shneiderman, B. (2006). D-Dupe: An Interactive Tool for Entity Resolution in Social Networks. *Proc. IEEE VAST '06*.

[7] Boudreau, M. and Robey, D. (2005). Enacting Integrated Information Technology: A Human Agency Perspective. *Organization Science*, 16(1):3-18.

[8] Brown, P. and Gould, J. (1987). An experimental study of people creating spreadsheets. *ACM Trans. Inf. Syst*. 5, 3 (July 1987), 258-272.

[9] Chang, K. and Myers, B. (2015). A Spreadsheet Model for Handling Streaming Data. *CHI '15*, 3399-3402.

[10] *Concert Programmes online database*. accessed 3/1/2016. http://www.concertprogrammes.org.uk/html/about

[11] [DB00] Dix, A., Beale, R. and Wood, A. (2000). Architectures to make Simple Visualisations using Simple Systems. *Proc. AVI2000*, ACM, pp. 51-60.

[12] Dix, A. (2007). Designing for appropriation. In *Proc. BCS-HCI '07* Vol. 2. BCS, UK, pp.27-30.

[13] Dix, A., Cowgill, R., Bashford, C., McVeigh, S. and Ridgewell, R. (2014). Authority and Judgement in the Digital Archive. In *1st International Digital Libraries for Musicology workshop (DLfM 2014),* ACM/IEEE Digital Libraries 2014.

[14] Dunn, H. (1946). Record Linkage. *American Journal of Public Health* 36 (12): pp. 1412–1416.

[15] Galletta, D., Hartzel, K., Johnson, S., Joseph, J. and Rustagi, S. (1996). Spreadsheet presentation and error detection: an experimental study. *J. Manage. Inf. Syst.* 13(3):45-63.

[16] Di Gioia, M., Scannapieco, M. and Beneventano, D. (2010). Object Identification across Multiple Sources. *Proc. 18th Italian Symp. on Adv. Database Systems, SEBD 2010*,.

[17] Hendry, D. and Green, T. (1994). Creating, comprehending and explaining spreadsheets. *Int. J. Hum.-Comput. Stud*. 40, 6 (June 1994), 1033-1065.

[18] Kay, A. (1984) Computer Software. *Scientific American* 251, 52–59.

[19] Kennedy, M. (2007). National Archive project to avert digital dark age. *The Guardian*, 4 July 2007.

[20] Kuny, T. (1997). A Digital Dark Ages? Challenges in the of Electronic Prevention Information. 63rd IFLA Council and General Conference.

[21] McVeigh, S. (1992–2014) *Calendar of London Concerts 1750–1800*. (Dataset) Goldsmiths, University of London. http://research.gold.ac.uk/10342/

[22] Monk, A. (1990). Action-effect rules: a technique for evaluating an informal specification against principles. *Behaviour & Information Technology*. 9(2):147–155.

[23] Nardi, B. and Miller, J. (1990). An ethnographic study of distributed problem solving in spreadsheet development. *Proc CSCW '90*. ACM, 197-208.

[24] Nikolov, A., d'Aquin, M. and Motta, E. (2012). Unsupervised learning of link discovery configuration. In *Proc. ESWC'12*, Springer-Verlag, 119-133.

[25] Peyton Jones, S., Blackwell, A. and Burnett, M. (2003). A user-centred approach to functions in Excel. In *Proc. ICFP '03*, ACM, 165-176.

[26] Rendle, S. and Schmidt-Thieme. L. (2006). Object identification with constraints. *Data Mining 2006*, 1026–1031.

[27] Scannapieco, M., Tosco, L., Valentino, L., Mancini, L., Cibella, N., Tuoto T. and Fortini, M. (2015). Relais User's Guide - Version 3.0. Technical Report, Istat, Italy. July 2015

[28] Singh, A., Bhadauria, V., Jain, A. and Gurung, A. (2013). Role of gender, self-efficacy, anxiety and testing formats in learning spreadsheets. *Comput. Hum. Behav*. 29(3):739-746.

[29] Tennison, J. (2014). *2014: The Year of CSV*. Open Data Institute. http://theodi.org/blog/2014-the-year-of-csv

[30] Thorne, S. (2009). A review of spreadsheet error reduction techniques. *Comm. Assoc. Inf. Sys* 25 (1), 34.