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. Four vignettes related to musicological data demonstrate 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.

Categories and Subject Descriptors

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

General Terms

Design, Human Factors.

Keywords

Spreadsheets, appropriation, musicology, digital humanities,

1. INTRODUCTION

This paper examines the way simple spreadsheets can be used in a surprisingly wide variety of ways as part of the user interface of relatively complex tasks. We base the paper around four vignettes from the InConcert project, which is studying the processes and tools for digital archives in musicology. However, we also draw from other examples in our own projects and that of others.

There is extensive research in the HCI and related literature focused on the use of spreadsheets, and particularly the potential for errors. However, this paper is focused predominantly on the spreadsheet as a plain table viewer and editor, which, because of its ubiquity and familiarity, can be used in flexible ways.

VisiCalc was a revolution in computing enabling, what would now be called, end-user programming to non-programmers. Critically, it was the result of collaboration between a programmer and a business user, a form of cooperative development, which is still unusual today. While we do not claim to achieve the same level of transformation, we do adopt a similarly participative approach, in this case between technologist and musicologists. The use of spreadsheets has facilitated this approach, as they enable a flexible way to create workflows that actually achieve results, but that are relatively low-cost in terms of implementation effort. This has increased the rate of deployment and reduced the risk of premature commitment.

We will see examples where the spreadsheet is used for input, output, and updating datasets. In some cases it is the canonical 'golden copy' of the data set, in others merely a temporary interaction artefact. In some cases the spreadsheet ends up being a highly functional, early prototype leading to a web-based system, in others the spreadsheet is the final interaction technique. The humble spreadsheet is found to be a flexible, familiar and adaptable resource both digitally, and (unexpectedly) physically.

In the following section we look at some of the related literature and systems on spreadsheets and tabular data. Section 3 gives a brief introduction to the InConcert project before the heart of the paper, section 4, which describes four vignettes of spreadsheet use in the project. Section 5 reflects on some of the lessons learnt from the vignettes and synthesises a taxonomy of uses and issues concerning the use of spreadsheets as user interfaces.

2. RELATED WORK

2.1 User-oriented spreadsheet research

As a major end-user application, spreadsheets have been studied in some of the earliest HCI and end-user programming literature. Indeed, the user-driven nature of the origins of VisiCalc have been an archetype and pattern for many discussions of successful software development. In 1984, Alan Kay placed programming languages on a scale and put spreadsheets in the top group, above Smalltalk and Prolog, as an 'ultra-high level language' [24]. He argued that the 'tissuelike' nature of the aggregation of cells enabled the spreadsheet to be used as a 'simulation tool', a use not

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from Permissions@acm.org.

Further conference information here

foreseen by its creators. He demonstrated the potential for appropriation by making histograms using table cell colouring, before graphing facilities were available. Early ethnographic studies by Nardi and Miller [31] 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, these collaborations were typically not one way, like older program-then-deliver paradigms, but truly co-design (before the term existed).

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. In 1987, one of the earliest formal studies found that 44% of spreadsheets contained errors, despite their expert users feeling "*quite confident that their spreadsheets were accurate*" [8], a pattern repeated in numerous studies since. Nearly ten years later, a study of error detection with over a thousand MBA students, and found that only half of spreadsheet errors were detected, even with alternative presentations, including showing the formulae below each cell [17].

The properties and problems of spreadsheet programming were one of the sources and early applications of Green's cognitive dimensions [20]. Empirical and theoretical analysis of spreadsheet use showed them ranking well on some dimensions such as "secondary notation" (e.g. using layout, formatting and non-calculated text fields to add additional information for users), but poorly on others such as "hidden dependencies" [21].

While there have been various tools and techniques proposed to help with spreadsheet authoring [41,40,35] and some implemented in production spreadsheets, this is by no means a solved problem. Indeed, the European Spreadsheet Risks Interest Group runs an annual conference solely on this topic [16].

2.2 Spreadsheet-like and tabular interfaces

Despite the known problems, spreadsheets have been and are still remarkably successful both for the financial purposes for which they were originally designed, and many other tasks.

There have therefore been many attempts to leverage these strengths, to extend the spreadsheet programming and layout model to improve or extend the range of use. Some of these operate largely within the existing spreadsheet paradigm. For example, Peyton Jones et al. proposed techniques to enable the coding of user-defined Excel functions within the spreadsheet rather than as separate Visual Basic, thus reducing the barriers to learning [35]. Some push the paradigm slightly, for example, Apple's Numbers, which allows multiple tables to exist within the same worksheet, both allowing clearer layout for some applications, but also reducing some causes of error. Some are more radical: for example, the recently released 'Guesstimate' allows cells to include upper and lower estimates, and Monte Carlo probabilistic techniques create outcome distributions rather than single central estimates [19].

In addition to extensions focusing on the programming power of spreadsheets, there are perhaps even more spreadsheet-like table editing and visualising interfaces. As well as direct web spreadsheets such as Google Sheets and online versions of Excel and Numbers, many data organisation applications have table-based interfaces that are at least spreadsheet-like, if not directly based on spreadsheets. This includes classic PC databases such as Access; web tools, such as Google Fusion Tables; and many visualisation and data analysis tools. Even where data is clearly graph-based, such as ontologies (e.g. Protégé [34]) or RDF (e.g. Tabulator [4]), table-based views are included as one of, or even the main, visualisation.

2.3 Using spreadsheets as the user interface

Because of their programming power, spreadsheets or spreadsheet extensions have been used for prototypes and full application building; for example Monk's spreadsheet simulator for action-effect rules, a form of user-interface specification [30], or, more recently, Gneiss, a spreadsheet tool for building streaming web data applications [10].

More prosaically, spreadsheets and CSV files are used extensively to import and export of data from databases or other applications. While not as standard as at first appears, CSV has become the *lingua franca* of data generally and of Open Data in particular. While there are clear arguments for more semantic formats such as RDF, and JSON has become ubiquitous in web APIs, the majority of open data is in CSV format – so much so that the Open Data Institute blog declared 2014 the "year of CSV" [39].

CSV or Excel spreadsheets are also used as the means for updating data. One example that many academics will have experienced is for the upload of marks. 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 were allocated a substantial number (from several hundred to over a thousand) of articles to read and assess, giving them scores and comments. This was accomplished by downloading personalised Excel spreadsheets, which listed all 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 as an update mechanism 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.

3. BACKGROUND –INCONCERT

The vignettes in the following section are all drawn from *In Concert: Towards a Collaborative Digital Archive of Musical Ephemera* [22], a sub-project of the AHRC funded Transforming Musicology programme [43]. The InConcert project has investigated the changing nature of digital humanities focusing on a number of inter-related datasets about concerts in London from the mid 18th century to early 20th century.

The aim of the project is partly to develop these particular datasets as digital assets, and partly to use them as a case study to understand and more fundamentally re-imagine the process of digital archiving in the humanities [14]. We hope to create a more lightweight, incremental archiving process akin to what would be seen as an 'agile' process in software development.

Methodologically, we have approached this as a usage and case study driven process, constrained by what is possible technically, but as far as possible addressing real musicological problems at a deep level. That is, while the technologies and processes may be transformed, we aim to do so in a way that preserves the underlying values and concerns of the musicologists.

In this paper we will mention the following datasets:

- LC18 – *Calendar of London Concerts 1750–1800* [29] – This was obtained through an exhaustive trawl of sources relating to concerts in the second half of the 18th Century.
- LC19 – *Concert Life in Nineteenth-Century London* [3] – By the 19th century the number of concerts grew to such an extent that exhaustive collation is not possible; instead, sample years at 20-year intervals were exhaustively studied, using newspaper archives and other sources.
- CPP – *Concert Programmes Project* [11] – This project, administered at the British Library, collates meta-information about archives; it does not contain programmes or programme text itself, but lists archives and collections (most offline) where such ephemera can be found and the venues they cover.
- BMB – *British Musical Biography 1897* [9] – As part of the InConcert project we created a digital version of this 400-page volume, which includes nearly 4000 entries for British musicians and composers.

In addition, we had access to a large body of OCR text from the *Konzertprogramm Austausch* (‘Concert Programme Exchange’) and various external digital resources, but these do not figure in the examples in this paper.

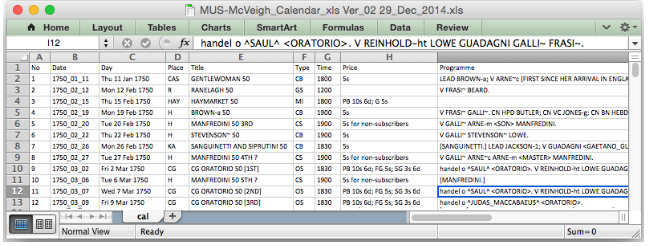
4. VIGNETTES OF SPREADSHEET USE

We now look at four vignettes from the InConcert project, where spreadsheets and tabular input/output were used in different ways.

4.1 Input of existing data

The first vignette concerns the simple import of CSV data. The 1750–1800 century concert data (LC18) was most well established, and had previously been in database format. However, as the original 1990s database had been 'retired', the data was now available only as Excel and CSV files. This consisted of a main file listing concerts (see fig 1), two 'authority' files for people and places with abbreviations for each (acting as unique identifiers), and a number of other abbreviations such as newspapers and sources (e.g. 'TI' for 'The Times'), and abbreviations used in concert descriptions (e.g. 'VN' for 'Violin').

Although this data had originally been in a relational database, the structure had been predominantly to aid human interrogation, not automatic processing. While the authority and abbreviation files had unique identifiers, few mapped uniquely to fields in the main concerts file. The 'place' field is always an abbreviation from the places authority file (the venue of the concert), a classic foreign key. However, the 'advert' field (denoting the source in which the advertisement for the concert appeared), contains newspaper title abbreviations, but it can include additional information (e.g. "PA 3 Apr"), or indeed multiple sources if there were several advertisements for the same concert. More complex still, name and other abbreviations can occur in multiple fields, but never as the sole item: the 'title' field (main composer or player in the concert), the semi-structured 'programme' field (describing the players and pieces) and within free text fields.

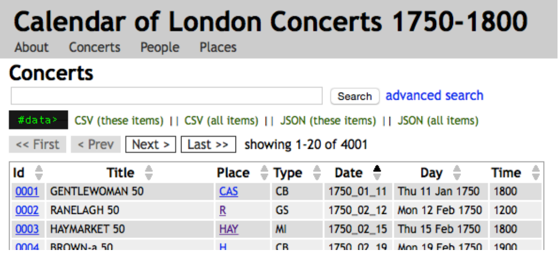


I12 handel o ^SAUL^ <ORATORIO>. V REINHOLD–ht LOWE GUADAGNI GALLI~ FRASI~.

	A	B	C	D	E	F	G	H	
1	No	Date	Day	Place	Title	Type	Time	Price	Programme
2	1	1750_01_11	Thu 11 Jan 1750	CAS	GENTLEWOMAN 50	CB	1800	5s	LEAD BROWN-a; V ARNE~c (FIRST SINCE HER ARRIVAL IN ENGLA
3	2	1750_02_12	Mon 12 Feb 1750	R	RANELAGH 50	GS	1200		V FRASI~ BEARD.
4	3	1750_02_15	Thu 15 Feb 1750	HAY	HAYMARKET 50	MI	1800	PB 10s 6d; G 5s	
5	4	1750_02_19	Mon 19 Feb 1750	H	BROWN-a 50	CB	1900	5s	V FRASI~ GALLI~. CN HPD BUTLER; CN VC JONES-g; CN BN HEBD
6	5	1750_02_20	Tue 20 Feb 1750	H	MANFREDINI 50 3RD	CS	1900	5s for non-subscribers	V GALLI~ ARNE-m <SON> MANFREDINI.
7	6	1750_02_22	Thu 22 Feb 1750	H	STEVENSON~ 50	CB	1900	5s	V GALLI~ STEVENSON~ LOWE.
8	7	1750_02_26	Mon 26 Feb 1750	KA	SANGUINETTI AND SIPRUTINI 50	CB	1830	5s	[SANGUINETTI.] LEAD JACKSON-1; V GUADAGNI <GAETANO_GU
9	8	1750_02_27	Tue 27 Feb 1750	H	MANFREDINI 50 4TH ?	CS	1900	5s for non-subscribers	V GALLI~ ARNE~c ARNE-m <MASTER> MANFREDINI.
10	9	1750_03_02	Fri 2 Mar 1750	CG	CG ORATORIO 50 [1ST]	OS	1830	PB 10s 6d; FG 5s; SG 3s 6d	handel o ^SAUL^ <ORATORIO>. V REINHOLD-ht LOWE GUADAG
11	10	1750_03_06	Tue 6 Mar 1750	H	MANFREDINI 50 5TH ?	CS	1900	5s for non-subscribers	[MANFREDINI.]
12	11	1750_03_07	Wed 7 Mar 1750	CG	CG ORATORIO 50 [2ND]	OS	1830	PB 10s 6d; FG 5s; SG 3s 6d	handel o ^SAUL^ <ORATORIO>. V REINHOLD-ht LOWE GUADAG
13	12	1750_03_09	Fri 9 Mar 1750	CG	CG ORATORIO 50 [3RD]	OS	1830	PB 10s 6d; FG 5s; SG 3s 6d	handel o ^JUDAS_MACCABAEUS^ <ORATORIO>.

Figure 1. Original Excel file for LC18

The musicologist's spreadsheet files are well established and used widely by the community; they are therefore regarded as the 'golden copy', not simply data to be input and 'cleaned'. The only modification was to add unique identifiers to the main concert spreadsheet. While this dataset is stable, it is possible that new sources and scholarship require minor additions or modifications, and so, following the 'golden copy' principle, this should be a re-import, rather than parallel updates. A batch process was used to import the CSV version of the data files, matching abbreviations in the relevant fields to create linkage data and create JSON datasets that were imported into a NoSQL database (nosqlite), allowing the creation of a web interface (see fig. 2) and cross-linking to other datasets (below).


Calendar of London Concerts 1750-1800
About
Concerts
People
Places
Concerts
Search
advanced search
#data>
CSV (these items) | | CSV (all items) | | JSON (these items) | | JSON (all items)
<< First
< Prev
Next >
Last >>
showing 1-20 of 4001

Id	Title	Place	Type	Date	Day	Time
0001	GENTLEWOMAN 50	CAS	CB	1750_01_11	Thu 11 Jan 1750	1800
0002	RANELAGH 50	R	GS	1750_02_12	Mon 12 Feb 1750	1200
0003	HAYMARKET 50	HAY	MI	1750_02_15	Thu 15 Feb 1750	1800
0004	BROWN-a 50	H	CB	1750_02_19	Mon 19 Feb 1750	1900

Figure 2. Web Interface for LC18

4.2 Matching and linking between datasets

This second vignette concerns entity identification between major datasets. We had name and place information from four sources. Both the 1750–1800 and 19th Century London Concert datasets (LC18 & LC19) have authority files for people (composers and performers) and places (venues). The Concert Programmes Project (CPP) has large authority files for places and agents (people, groups and organisations), including some geo-referencing and planned VIAF links. British Musical Biography (BMB) has people’s names only, but is comprehensive.

Automatic matching was used to create candidate matches followed by a hand verification stage. The latter was crucial as the *authoritative* nature of the data was a key academic value for the humanities researchers [14]; automatic matching, whilst useful, is bound to be inaccurate, yielding both false positives and false negatives. Following the principles of 'appropriate intelligence' [12], the automatic algorithms were not designed to be as clever as possible, but instead to be part of a human–computer system that *as a whole* yields reliable results.

4.2.1 Automatic matching

Places were simplest using plain word matching and permuted word indexes for efficiency. There are fewer place names than people's names and they tended to be more standardised; so simple matching was sufficient for candidate identification

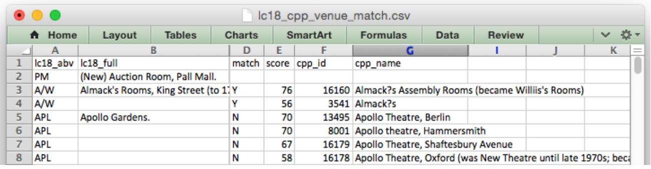
People names were more complex. First, this was because the data sources needed an element of cleaning/normalisation. In the LC18 dataset, the ids included an encoding of the surname, gender and possible disambiguation; for example "KNEISEL~" for the female (trailing tilde) "Henriette Kneisel", or "TURNER-2" for one of two "Turner"s. This was relatively straightforward pattern matching. More complex was the CPP data, which included groups and organisations as well as people and also was itself garnered from multiple sources. Some people names had the forename as a separate field, some were in 'surname, first name' format, and some were more complex, including honorifics. In the spirit of maintaining the original source as 'golden copy', this task was managed through a combination of keywords for terms in organisations (e.g. 'orchestra, 'Staatstheater'), extensive lists of honorifics (e.g. 'Prince', 'Mlle', 'Duke of'), and explicit *exceptions* (e.g. that record id '2173' named 'Tate Britain' is an organisation not someone with surname 'Britain').

Having normalised names as much as possible, the automatic algorithm matched between datasets using a similar word match measure to the places. Fuzzy matches were not used, as this led to too many false positives and the point of the algorithm was to aid not replace human matching. Note that while crude whole word matching was used for the batch processing for names, fast fuzzy search is enabled in online datasets using both Soundex and 'drop one character' indexes. The latter stores every combination of each name with single characters dropped; by doing the same for retrieval terms one can obtain a good triage pass before more sophisticated edit distance measures are calculated.

4.2.2 Human processing

Having obtained automatic 'candidate matches', these were then exported as CSV tables showing names from one data set (the source) on the left, the possible matches (targets) on the right, and a confidence value between. A 'match' field is also included, initially set to '?'. The musicologist then was able to go through these changing the '?' to 'Y' (yes), 'N' (no) or 'P' (not sure).

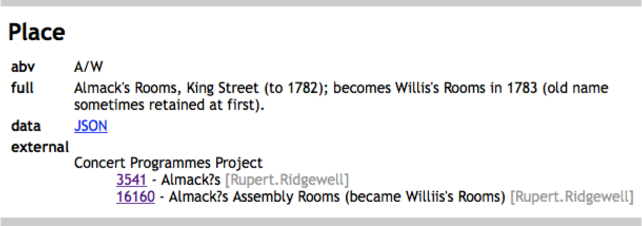
Figure 3 shows the beginning of the completed places spreadsheet for the LC18 matches against the CPP authority file. Note the two verified candidates for LC18 'A/W' against CPP ids '79' and'56'. This represents a case where the CPP authority file has unresolved entities from its own different sources. The 'APL' entry has no matches. In general, verified matches were almost always for the entry with highest automatic confidence score; however, there was no sensible 'critical value' for this confidence score, highlighting the need for human expert evaluation.



	A	B	D	E	F	G	I	J	K
1	lc18_abv	lc18_full	match	score	cpp_id	cpp_name			
2	PM	(New) Auction Room, Pall Mall.							
3	A/W	Almack's Rooms, King Street (to 17	Y	76	16160	Almack?s Assembly Rooms (became Williis's Rooms)			
4	A/W		Y	56	3541	Almack?s			
5	APL	Apollo Gardens.	N	70	13495	Apollo Theatre, Berlin			
6	APL		N	70	8001	Apollo theatre, Hammersmith			
7	APL		N	67	16179	Apollo Theatre, Shaftesbury Avenue			
8	APL		N	58	16178	Apollo Theatre, Oxford (was New Theatre until late 1970s; beca			

Figure 3. Place matching spreadsheet

The completed spreadsheet was processed to create a link dataset listing the connections between the datasets (similar to RDF 'sameAs'). By keeping this separate, it is possible to easily maintain the provenance of the link information, fully automatic or human, and if human by whom (see fig. 4). Different experts may resolve the names in different ways, or decide whether they trust the source of the linkage information (automatic or human) for a particular scholarly purpose. This cross-linking will also be used to enable RDF Linked-Data views of the datasets [6].

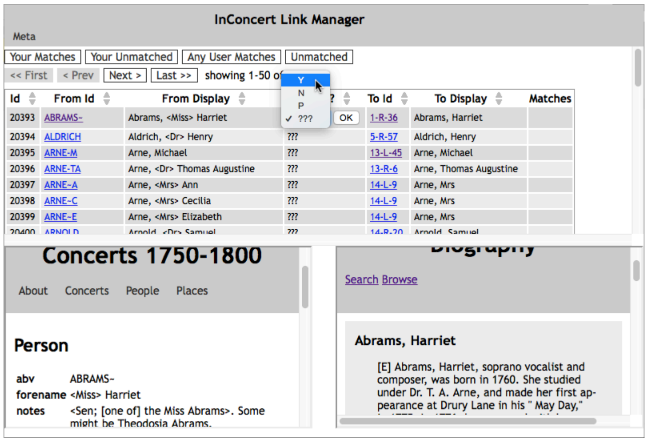


Place

abv A/W
full Almack's Rooms, King Street (to 1782); becomes Willis's Rooms in 1783 (old name sometimes retained at first).
data JSON
external
Concert Programmes Project
3541 - Almack?s [Rupert.Ridgewell]
16160 - Almack?s Assembly Rooms (became Williis's Rooms) [Rupert.Ridgewell]

Figure 4. Links displayed with provenance

While this process worked, the musicologist who performed the matching felt there was insufficient information readily available. For many of the matches it was a simple matter to look at the entries in the table and see whether they did, or did not, constitute a match. However, some entries were more complex and, while it was possible to look up the full information from the ids, this was not as easy as simply scanning the list. A web interface was therefore constructed following the tabular list metaphor as closely as possible, but adding panes showing web pages with full information for the entries currently being matched (fig. 5). Note that the offline spreadsheet effectively acted as a *fully functional prototype*, *establishing requirements* for the final web interface.



Id	From Id	From Display	?	To Id	To Display	Matches
20393	ABRAMS~	Abrams, <Miss> Harriet		1-R-36	Abrams, Harriet	
20394	ALDRICH	Aldrich, <Dr> Henry	???	5-R-57	Aldrich, Henry	
20395	ARNE-M	Arne, Michael	???	13-L-45	Arne, Michael	
20396	ARNE-TA	Arne, <Dr> Thomas Augustine	???	13-R-6	Arne, Thomas Augustine	
20397	ARNE~A	Arne, <Mrs> Ann	???	14-L-9	Arne, Mrs	
20398	ARNE~C	Arne, <Mrs> Cecilia	???	14-L-9	Arne, Mrs	
20399	ARNE~E	Arne, <Mrs> Elizabeth	???	14-L-9	Arne, Mrs	
20400	ARNOLD	Arnold, <Dr> Samuel	???	14-R-20	Arnold, Samuel	

Person

abv ABRAMS~
forename <Miss> Harriet
notes <Sen; [one of] the Miss Abrams>. Some might be Theodosia Abrams.

Abrams, Harriet

[E] Abrams, Harriet, soprano vocalist and composer, was born in 1760. She studied under Dr. T. A. Arne, and made her first appearance at Drury Lane in his " May Day,"

Figure 5. Prototype web interface for link checking

4.3 Grouping and matching within a dataset

The numbers of concerts and the number of press notices for each concert were far higher in the 19th century than the late 18th century, so that, even with year selection, the volume of work required for the LC19 dataset is large. The capture of the data had used a number of research assistants over several years – this compiled substantial information about notices or adverts about concerts. Multiple notices often referred to the same concert. The remaining task, which the expert musicologists needed to do, was to go through these concert notices, work out which ones referred to the same event and create an authoritative entry for each concert. This process the musicologists refer to as 'skewering', but database technologists would think of as entity/object identification or record linkage [15, 1].

This process had acted as a block to progress, as it was so substantial and required expert attention. A major breakthrough was realising that this consisted of (at least) two separable sub-tasks: (i) *match* – 'skewer' multiple notices referring to the same concert; (ii) *merge* – combine the data from the notices to create an authoritative record for the concert. It became clear that, while the effort in doing the match task was substantially less than the merge task, still the dataset would become substantially more valuable once the first sub-task was complete.

4.3.1 Automatic entity identification

There is a substantial literature on entity/object identification dating back from the early days of databases [1] to semantic web applications [32]. Sometimes this involves simple similarity measures such as Jacquard distance between feature sets, or Levenshtein edit distance for string matching. Other researchers have used complex machine learning techniques, including using structural relationships in relational or graph databases [36, 5, 18]. There is also tool support. OpenRefine (formerly Google Refine) supports the management of data including linking names to entities (possibly more like the name matching in the previous section), although it does not do matching itself, passing this task on to external data services through its Reconciliation Service API [33]. RELAIS (REcord Linkage At IStat) is dedicated to the process of record linkage itself [37]; it supports a number of different matching algorithms that can be applied to any combination of fields.

However, as with the name matching, because this was part of human–computer process, simpler automatic matching was sufficient combined with methods to make the human task easier.

Initial matching used the date and venue of the concerts; those that had the same date and similar venue names were matched into groups. This led to some false negatives (e.g. if the date or venue of a concert changed between notices) and false positives (several concerts at the same venue on the same day). This data was then exported to an Excel spreadsheet for processing by the musicologists. The use of Excel rather than CSV was to enable highlighting and the generation of live url links (see below).

Entries that had been grouped together were listed on subsequent lines with multiple line gaps between each group to make visual scanning as easy as possible. To help deal with the false negatives, the data was sorted by date, and to help deal with false positives, some groups were highlighted as 'warnings'. In addition, live url links were included to the web version of each record where all the fields could be studied, as only a small selection of the data was included in the spreadsheet itself. The warning labels were added when either the times or venue names did not match exactly and they were displayed partly by a 'warnings' column and partly by highlighting the records affected. Of course, venue names might be slight variants of the same location, and notices saying "evening" or "7pm" might refer to the same time. The choice to be slightly liberal in similarities for grouping (favouring false positives) and also liberal in marking warnings was because of the subsequent human verification stage.

As with the name matching, columns were added for the musicologists to fill in to agree with or modify the groupings: a 'match' column, an expert confidence column (high/medium/low) and a 'comment' column, so that any odd cases could be annotated. If the grouping was correct, a 'Y' was recorded in the 'match' column, and a simple coding scheme was agreed for cases where the grouping had to be ammended. The completed spreadsheets were then simply re-imported.

4.3.2 Spreadsheets anxiety and appropriation

Overall the process worked well. However, one musicologist initially expressed concern about using the spreadsheets. It turned out that this was because of having used university financial spreadsheets in the past that were very 'fragile'. Such spreadsheet anxiety has been described elsewhere [38] and is, of course, 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. However, this justified fear of 'breaking' the spreadsheet was one reason for using an additional column to mark groups rather than moving rows around and inserting blank lines (the original design idea); it was felt that editing fields was far less fragile than moving rows.

The musicologist who completed this task liked the spreadsheet view, noting that it was reminiscent of a Paradox database used many years before. 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. 6), and covering the paper copies in copious notes (fig. 7). While some notes were transcribed into the notes field in the spreadsheet, others were left only on the paper copy as a record of the process that led to the decisions.

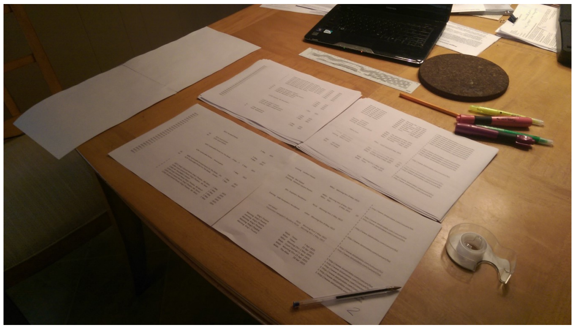
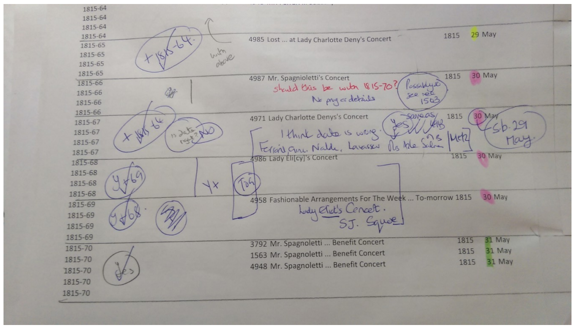


Figure 6. Printed spreadsheet for working



1815-64
1815-64
1815-64
1815-64 +1815-64. with above 4985 Lost ... at Lady Charlotte Deny's Concert 1815 29 May
1815-65
1815-65
1815-65
1815-65
1815-66 4987 Mr. Spagnioletti's Concert 1815 30 May
should this be with 1815-70? Possibly see 1563
No prog or details
1815-66
1815-66
1815-66
1815-67 +1815-64 is date right? NO 4971 Lady Charlotte Denys's Concert Yes same as 4918 1815 30 May S.b. 29 May.
I think date is wrong.
Ferrari, Guu. Naldi, Lavasseur, Ms Able Salm Metz
1815-67
1815-67
1815-67
1815-68 4986 Lady Eli[cy]'s Concert 1815 30 May
1815-68 Y+69 YX Tog
1815-68
1815-68
1815-69 Y+68 4958 Fashionable Arrangements For The Week ... To-morrow 1815 30 May
Lady Eli[cy]'s Concert.
SJ. Square
1815-69
1815-69
1815-69
1815-70 Yes 3792 Mr. Spagnoletti ... Benefit Concert 1815 31 May
1815-70 1563 Mr. Spagnoletti ... Benefit Concert 1815 31 May
1815-70 4948 Mr. Spagnoletti ... Benefit Concert 1815 31 May
1815-70
1815-70

Figure 7. Copious notes on printed spreadsheet

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 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 also may be a slight and reasonable distrust of electronic storage.

LC18, the oldest electronic dataset being used here, was on its fourth iteration of technology. The LC19 dataset had been locked in an out-of-date version of a large proprietary database for some years, in theory 'live', but in practice virtually inaccessible. It had to be migrated through an up-to-date (and expensive) version of the database and then via an SQL dump into MySQL; the latter stage entailed substantial manual and automated transformations, as SQL has many proprietary variations.

One musicologist had concert data in Paradox and an old version of Access, inaccessible for many years. We managed to transform that data into bare CSV with only minor losses. The data was on an old Zip 'backup' disk and floppy disks, presenting physical medium as well as file format issues. In contrast, paper notes for InConcert from the same period were in old archive boxes; not at the musicologists' fingertips, but still accessible.

The musicologists are not alone in facing this problem, sometimes referred to as the 'digital dark ages' [27]. Major archival institutions have projects to restore and future-proof past digital materials [25] and the major BBC 'Domesday' dataset collected in 1986 was only narrowly saved when the Laserdisk format became unreadable [28] and this was despite the best efforts of the original technology team [42]. In 2002 Jeff Rothenberg of Rand Corporation, was quoted as saying: "*There is currently no demonstrably viable technical solution to this problem; yet if it is not solved, our increasingly digital heritage is in grave risk of being lost*." [28], and in 2007 Adam Farquhar of the British Library was reported as saying that, "*the nightmare of millions of stored unreadable files had caused him sleepless nights*" [25]. 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.

4.4 Analysing and Visualising

The previous vignettes have concerned the creation of the cross-linked InConcert digital datasets; this last vignette relates to the actual use of the datasets.

One of the musicologists was presenting at an important venue and wanted to demonstrate new ways of viewing the datasets. This included histograms and timelines of the popularity of individual composers and countries of origin and also map-based visualisation of the changing popularity of venues covering both LC18 and LC19 datasets.

Some of the data for this was sourced directly from the canonical LC18 CSV files, and some from CSV files exported from the LC19 through web-based queries. The data was processed in Excel partly by the technologist and partly by the musicologists themselves. Crucially, early analysis was driven entirely by the musicologist responsible for the LC18 data using Excel functions and the entire analysis driven by real musical questions, not "what we can do with the technology".

The whole process was documented in order to generate requirements for aspects better managed within an online system. We are again using the spreadsheet for real work, but also as a form of highly functional prototype, or possibly provotype [7], to understand future system requirements.

Some of the requirements were 'standard' data analysis operations, for example, the ability to perform rich aggregation calculations (e.g. cross-tabulation counts of concerts by particular composers and years) and graph them. However, not all of these operations were easily managed using spreadsheet formulae alone, and were certainly not available with most standard database interfaces.

There were various more interesting outcomes and requirements. The most significant venues were *geo-referenced*. Around 2/3 of this data was obtained directly from the links to CPP. This was a strong validation of the *value of linking datasets*. While some information was obtained directly from the datasets, *new columns* were added to the spreadsheets, notably the countries of composers. These required interpretation in the context of the particular analysis (country of birth vs residence at time of composition or period of study). For this exercise, the data was processed for each dataset largely independently and was brought together for visualisation. For other analysis it would be ideal to create *working sets of data crossing (subsets of) the datasets*, whilst retaining provenance, and potentially normalising to account for different data collection methods.

5. LESSONS FOR DESIGN

The vignettes and other examples demonstrate a variety of uses of spreadsheets as the user interface. This section summarises some of the main modes of use and broader issues raised.

5.1 Import/Export

The simplest use of CSV is for import or export of data. The examples of university exam systems and the REF system, adopted *download/upload* of spreadsheets through web interfaces. Alternatively, CSV export may become *input into a desktop system* such as JASP [23], or, as often seen in the InConcert vignettes, produced by, or be input to, *batch processing* steps before being incorporated into online systems.

Web systems may access spreadsheet data 'live'. In various projects, we have used web applications to directly access *links to shared Dropbox files*. The end-user can edit these files directly, or do a more controlled export to the file. In InConcert, this was used for both CSV files, and post-processed JSON files. For prototyping purposes or small datasets, this process can sometimes be sufficient, but for larger datasets this would normally be used as input to a further processing stage to initialise a database. Similar techniques can be used to feed information directly using *APIs to online spreadsheets*. For example, timeline.js uses this method to input raw data form Google Sheets for timeline visualisation [26] (e.g. timeline tool).

5.2 Forms of update

In several InConcert vignettes and in the REF and exam mark system examples, we saw the use of spreadsheets as the means to update live data. Here the spreadsheet really is being used as an integral part of the user interface, leveraging the development of the spreadsheet as part of the overall system workflow. However, we have seen several different kinds of update:

editing fields – This is the simplest use, updating one or more fields as we saw in exam mark systems and REF. This may be for additional data entry or some form of checking and confirmation.

adding fields – In the last InConcert vignette we saw the potential to add user defined fields. Of course this means that the online data store, or other post-processing, has the means to manage additional fields. In the case of noSQL databases this is quite straightforward. Such fields may be private, but may be sharable, allowing datasets to grow through a form of crowdsourcing.

matching and linking – The InConcert person and place name linking has shown ways in which spreadsheets can be used to manage link creation. In this vignette, the user was validating automatic candidate links; different mechanisms may be needed for entirely user driven linking.

grouping – The third vignette showed how the spreadsheet could be used to confirm and to modify record grouping (this need not necessarily be for entity identification). Again, slightly different means might be needed for entirely user driven grouping.

reordering – We did not seen any examples of using spreadsheets to alter orders of elements, but, so long as the records have identifiers/keys, this would be a straightforward application.

5.3 Issues and lessons

from prototype to full system – We have seen examples that range from initial prototypes to those where the spreadsheet is the fully delivered user interface. Furthermore, the functional nature of the spreadsheet as user interface means that this can be a 'soft' decision, initially delivering a spreadsheet user interface that can be replaced by a bespoke one if it becomes necessary.

avoiding fragility – Care is need to ensure robust solutions. Unique id fields are central to this (as we saw in the first vignette), but also careful choices as to what kinds of spreadsheet updates to allow (whether by protecting fields, or human processes).

complex fields – Data designed for human use may not always be 'normalised' in the way databases expect (e.g. the programme field in the LC18 dataset). However, this can be a strength as well; codified fields can be used to manage more complex forms of data (e.g. the programme field would probably require at least two extra subsidiary tables in a relational design).

golden copy – We emphasised several times the importance of preserving the users' own dataset curated and managed in the way most meaningful to them. This is a general information systems design principle, the opposite of the more common approach of centralising data and treating peripheral datasets as 'views' 'derived' from it. This principle applies to a golden copy in any format, but because of the ubiquity of spreadsheets and CSV, it applies particularly to tabular data.

importance of exceptions – In multiple places we talked about the importance of exception sets. This goes hand in hand with the last point: if the users' original data is seen as the golden copy, then it is important to be able to re-import it when it changes. Classic data cleaning tends to work on the imported data and has embedded the assumption of centralised canonical data. Exception sets allow one to break this import-once mindset.

provenance and crowdsourcing – In InConcert, we needed to keep track not just of changes, but who made those changes; this is required both for expert updates and even more for less expert or crowdsourced data. Again this is not purely a spreadsheet issue, but the ease of spreadsheet editing makes third-party additions easier. It is critical to track and make visible the sources of updates and, where necessary, filter based on sources.

flexibility and appropriation – Many appropriation design principles [13] apply to spreadsheets: for example, "allow interpretation" (formatting, layout, etc.), "support not control" (a tool that allowed financial calculations, but was not built solely to do them). In some ways the use as a user interface component is an extreme form of appropriation and by providing spreadsheets to people, they can perform their own appropriation. For example, one of the authors involved in REF reviewing added a worksheet to calculate statistics of graded papers.

physicality – The paper printouts are a particular example of appropriation, but here escaping the purely digital domain. It would be possible to print out, say, individual web pages, but this would be voluminous, and not as easy to select desired fields.

human–computer symbiosis and appropriate intelligence – The 'intelligent' algorithms used in InConcert followed the principles of appropriate intelligence [12], providing just sufficient cleverness to aid the complete human–computer system. Often 'human computation' [2] treats the human as a (sometimes unwitting) cog in the machine. In contrast, the vignettes showed ways in which the user's expertise is maximised.

6. SUMMARY

The vignettes and examples have shown that the spreadsheet is far from inconsequential, functioning as a rich tabular interaction 'widget' in complex data manipulation workflows. While it would been possible to design dedicated matching, linking and data update interfaces, the export and import of well-designed spreadsheets offer 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 some of the range of potential applications. 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 here 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.

7. ACKNOWLEDGEMENTS

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

8. 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. doi:10.1109/TKDE.2007.9

[2] von Ahn, L., Maurer, B., McMillen, C., Abraham, D. and 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 http://swui.semanticweb.org/swui06/papers/Berners-Lee/Berners-Lee.pdf

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

[6] Bizer, C., Heath, T. and Berners-Lee, T. Linked Data – The Story So Far. *International. Journal on Semantic Web and Information Systems*, 2009.

[7] Boer, L. and Donovan, J. (2012). Provotypes for participatory innovation. *Proc DIS '12*. ACM, 388–397. doi:10.1145/2317956.2318014

[8] Brown, P. and Gould, J. (1987). An experimental study of people creating spreadsheets. *ACM Trans. Inf. Syst*. 5, 3 (July 1987), 258–272. DOI: 10.1145/27641.28058

[9] Brown, J. and Stratton, S. (1897). *British Musical Biography: a dictionary of musical artists, authors and composers, born in Britain and its colonies*. S.S. Stratton, Birmingham. http://www.datatodata.com/in-concert/BMB/

[10] Chang, K. and Myers, B. (2015). A Spreadsheet Model for Handling Streaming Data. In *Proc. CHI '15*. ACM, 3399–3402. doi:10.1145/2702123.2702587

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

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

[13] Dix, A. (2007). Designing for appropriation. In *Proc. BCS-HCI '07* Vol. 2. BCS, UK, pp.27–30. http://ewic.bcs.org/content/ConWebDoc/13347

[14] Dix, A., Cowgill, R., Bashford, C., McVeigh, S. and Ridgewell, R. (2014). Authority and Judgement in the Digital Archive. In *The 1st International Digital Libraries for Musicology workshop (DLfM 2014),* ACM/IEEE Digital Libraries conference 2014, London 12th Sept. 2014. http://www.hcibook.com/alan/papers/DLfM-2014/

[15] Dunn, H. (1946). Record Linkage. *American Journal of Public Health* 36 (12): pp. 1412–1416. doi:10.2105/AJPH.36.12.1412

[16] *European Spreadsheet Risks Interest Group (EuSpRIG)*. (accessed 4/1/2016) http://www.eusprig.org

[17] 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. doi: 10.1080/07421222.1996.11518133

[18] Di Gioia, M., Scannapieco, M. and Beneventano, D. (2010). Object Identification across Multiple Sources. *Proc. of the Eighteenth Italian Symposium on Advanced Database Systems, SEBD 2010*, Rimini, Italy, June 20–23, 2010.

[19] Gooen, O. (2016). *Introducing Guesstimate, a Spreadsheet for Things That Aren't Certain*. (accessed 1/1/2016). https://medium.com/guesstimate-blog/introducing-guesstimate-a-spreadsheet-for-things-that-aren-t-certain-2fa54aa9340

[20] Green, T. (1989) Cognitive dimensions of notations. *People and Computers V*. Cambridge University Press, 443–460.

[21] Hendry, D. and Green, T. (1994). Creating, comprehending and explaining spreadsheets. *Int. J. Hum.-Comput. Stud*. 40, 6 (June 1994), 1033–1065. doi:10.1006/ijhc.1994.1047

[22] *In Concert* (2014–2016). accessed 3/1/2016 http://inconcert.datatodata.com

[23] *JASP*. (accessed 4/1/2016) https://jasp-stats.org

[24] Kay, A. (1984) Computer Software. *Scientific American* 251, 52–59 . doi:10.1038/scientificamerican0984-52

[25] Kennedy, M. (2007). National Archive project to avert digital dark age. *The Guardian*, 4 July 2007. http://www.theguardian.com/technology/2007/jul/04/news.uknews

[26] KnightLab(2016) *Timeline.js*. (accessed 4/1/2016). http://timeline.knightlab.com/#show-faq-20

[27] Kuny, T. (1997). A Digital Dark Ages? Challenges in the of Electronic Prevention Information. *63rd IFLA (International Federation of Library Associations and Institutions) Council and General Conference*.

[28] McKie, R. and Thorpe, V. (2002). Digital Domesday Book lasts 15 years not 1000. *The Guardian*. Sunday 3 March 2002. http://www.theguardian.com/uk/2002/mar/03/research.elearning

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

[30] Monk, A. (1990). Action-effect rules: a technique for evaluating an informal specification against principles. *Behaviour & Information Technology*. 9(2):147–155. doi: 10.1080/01449299008924231

[31] Nardi, B. and Miller, J. (1990). An ethnographic study of distributed problem solving in spreadsheet development. *Proc CSCW '90*. ACM, 197–208. doi: 10.1145/99332.99355

[32] Nikolov, A., d'Aquin, M., and Motta, E. (2012). Unsupervised learning of link discovery configuration. In *Proc. ESWC'12*, Springer-Verlag, Berlin, Heidelberg, 119–133. doi: 10.1007/978-3-642-30284-8_15

[33] OpenRefine: *Reconciliation Service API*. (accessed 2/1/2016).https://github.com/OpenRefine/OpenRefine/wiki/Reconciliation-Service-API

[34] Protege (2000–2016). *The Protégé Project*. (accessed 4/1/2016) http://protege.stanford.edu

[35] Peyton Jones, S., Blackwell, A. and Burnett, M. (2003). A user-centred approach to functions in Excel. In *Proc. ICFP '03*, ACM, 165–176. doi:10.1145/944705.944721

[36] Rendle, S. and Schmidt-Thieme. L. (2006). Object identification with constraints. *Data Mining, 2006.*, 1026–1031, http://www.ismll.uni-hildesheim.de/pub/pdfs/Rendle_SchmidtThieme2006-Object_Identification_with_Constraints.pdf

[37] 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, Italian National Institute of Statistics (Istat). July 2015, doi:10.13140/RG.2.1.1332.5922

[38] 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 (May 2013), 739–746. doi:10.1016/j.chb.2012.11.009

[39] Tennison, J. (2014). *2014: The Year of CSV*. Open Data Institute. 11 Jan. 2014. http://theodi.org/blog/2014-the-year-of-csv

[40] Thorne, S. and Ball, D. (2005). Exploring Human Factors in Spreadsheet Development. *Proc. European Spreadsheet Risks Int. Grp. (EuSpRIG) 2005* 161–172. arXiv:0803.1862 [cs.HC]

[41] Thorne, S. (2009). A review of spreadsheet error reduction techniques. *Communications of the Association for Information Systems* 25 (1), 34. http://aisel.aisnet.org/cais/vol25/iss1/34/

[42] Tibbetts, M. (2008). *Re: BBC Domesday Project* (Leeson, RISKS-21.93). (message dated Tue, 4 Nov. 2008). The Risks Digest, 25(44), ACM, 8 Nov. 2008(accessed 3/1/2015) http://catless.ncl.ac.uk/Risks/25.44.html#subj7

[43] *Transforming Musicology* (accessed 3/1/2016). http://www.transforming-musicology.org