Personal Numeric Information Management

Alan Dix
alan@hcibook.com
Cardiff Metropolitan University and Swansea University
Wales, UK
https://alandix.com/academic/papers/PNIM-2024/

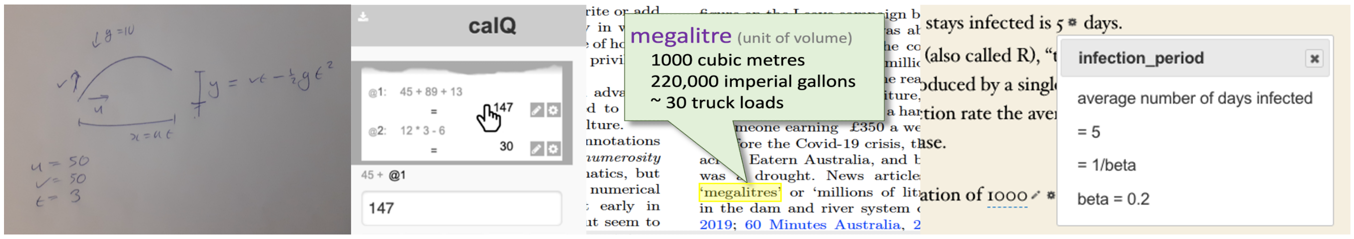


Figure 1: Screenshots of (i) hand calculation, (ii) calQ, (iii) TSoW, and (iv) WS2.

ABSTRACT

N.B. Draft paper. Extended version of AVI 2024 short paper "Just Counting – a tool ecosystem for personal numeric information". Please check https://alandix.com/academic/papers/PNIM-2024/ for updates and citation information.

Numbers are part of day-to-day life from household budgeting to making sense of global warming and planning academic projects. But, for many, dealing with numeric information is daunting with multiple step changes in complexity moving from, say simple calculations to spreadsheet use, as well as difficulties managing different sources of complex information. In this paper we present an ecosystem of interconnected prototype tools that explore this space, including TSoW interpreting unfamiliar orders of magnitude; calQ a four-function calculator that shifts seamlessly to micro-spreadsheet; WS2 embedding spreadsheet-like features in web pages; and myData collating and connecting the diverse data sources. Collectively, these tools offer an envisionment to prompt discussion both of the way end-users can more easily deal with numeric information and of the background technical infrastructure necessary for this to happen.

KEYWORDS

Numeracy, personal information management, qualitative–quantitative reasoning

1 INTRODUCTION

"*The power to understand and predict the quantities of the world should not be restricted to those with a freakish knack for manipulating abstract symbols.*" Brett Victor [59]

Numbers are part of day-to-day life from household budgeting to making sense of global warming and planning academic projects. This was particularly clear during the Covid pandemic when viral spread, growth rates and exponential growth became common parlance. However, there is constant talk of a 'numeracy crisis' [61], which affects the day to day life of many, but also has large scale effects. In terms of economics, a report commissioned by the UK charity National Numeracy suggests that poor numeracy costs the UK economy £25 billion a year [42], an impact that will be replicated across the world. However, perhaps the greatest deficit is in terms of governance as so many vital national and international issues, such as those above, require a numerically informed citizenry.

There are long-standing educational issues, discussed further in Section 2.1, but also a need for immediate solutions. One of the problems is that those using tools and techniques for dealing with numerical information often encounter multiple hurdles – for example, the step from calculator to spreadsheet, or spreadsheet to coding. In addition, the information we need is often scattered and complex to understand.

Of course, this paper does not solve this issue, but it does attempt to explore how appropriate tools could help. Note the focus throughout will be on 'data in the small', the scraps of numerical and structured information we encounter day-to-day in professional and domestic lives. So, while in some ways this can be seen as data-science-in-the-small it is closer to the world of personal-information-management (PIM) [7, 33], that is the goal is *personal numeric information management*.

The roots of this work go back many years, including the Kay's analysis of VisiCalc and systems such as HyperCard (discussed in Section 2.2) and more recent recognition of the need for 'qualitative–quantitative" reasoning [13].

David Mackay's highly influential book *Sustainable Energy-without the hot air* [39] has also been an inspiration. Mackay was an accomplished physicist, but also involved in public policy, so used to communicating complex information. The book uses many back-of-the-envelope style calculations, often using everyday information, such as the calorific value of a packet of butter. These both inform the reader in an accessible way, but also could, in principle, be recalculated in the light of changing information or assumptions. The book's excellent website, makes these calculations easy to find, but they need to be transferred to a calculator or spreadsheet for manipulation. What are the tools that would make this process easier for the end user, or allow those less numerically skilled to author a similar book or website?

In this paper we present an ecosystem of interconnected prototype tools that explore this space: ***calQ*** a four-function calculator that shifts seamlessly to micro-spreadsheet; ***myData*** collating and connecting the diverse data sources we encounter; ***WS2*** a WordPress plugin to embedded spreadsheet-like features in web pages; and ***TSoW*** (the size of Wales) interpreting unfamiliar orders of magnitude. These tools offer an envisionment to prompt discussion of both of the way end-users can more easily deal with numeric information and the background technical infrastructure necessary for this to happen.

The goal is not to promote these tools individually, nor even in composite, but more to use them as a technology probe [31] to help think about these issues and prompt further work.

2 BACKGROUND

2.1 Motivation: The crisis in numeracy

For many people mathematics and numerical information is regarded daunting, a pain one had to endure at school and ideally ignore or avoid ever since. In 2023, a UK survey found that, "*over a third of adults (35%) say that doing maths makes them feel anxious, while one in five are so fearful it even makes them feel physically sick*" [43]. These negative feelings about numbers have been recognised as a serious psychological impediment for more than sixty years [16], initially called number anxiety in the 1950s [17] and now mathematics anxiety [3, 28].

Some of the roots are in education. John Holt's classic '*How Children Fail*' [29] describes the way that in traditional education children learn to succeed at giving the right answers, but fail to really learn. His examples are primarily mathematical not least because of the 'one right answer' nature of much of mathematical education, which ignores the exploratory nature of real mathematics. Primary education in many countries has improved since the early 1960s, albeit with backward steps along the way, often prompted by the need for easy bench-marking. Current best practice, at least at primary level, is often focused around exploratory learning and knowledge embedded in the real world [18], but later in schooling this is often lost as the need for measurable outcomes grows.

There is clearly still need to improve at an educational level, both in terms of pedagogic practice, but moreover at a policy level where metrics-driven results make it hard for teachers to educate in the most productive fashion. However, even if this were solved tomorrow (unlikely as it may seem), this would not help the current adult population. We cannot wait sixty years for a citizenry able to deal with the increasingly numerical nature of national and global issues, let alone the minutiae of managing everyday life.

2.2 Inspiration: rethinking computation

In 1984 Alan Kay wrote a piece entitled "*Computer Software*" [36], where he compared different software genres, starting with assembly language, then progressing through high level languages (HLL) such as Fortran and Algol (where many current languages would still sit), through to very high-level languages (VHLL) including Prolog (AI logic programming) and Smalltalk (object oriented), and finally ultra-high-level languages (UHLL) where he placed VISICALC, the first spreadsheet, still new at that time. (He also placed Lisp at virtually every level.)

Crucial to his view of VISICALC was the way rules were intimately tied to data, and he looked forward to ways in which the simple numeric spreadsheet could develop [26], including his own early concept piece, "*Opening the Hood of a Word Processor*" [35], which envisaged a next generation document processor that would combine text, graphics and computation.

Although it is not documented, the timing of this article and a seminar of Alan Kay at Apple in the early 1980s suggests that this paper or the talk was part inspiration for HyperCard and indeed Alan Kay spoke highly of HyperCard [30], which was a mix of hypertext authoring tool and programming environment.

Knuth's tools for literate programming [38], also developed in the early 1980s, allowed code and documentation to be freely mixed. Thimbleby's executable extensions [55] to this made code examples in text executable and outputs automatically updated ensuring the documentation is accurate, further blurring the boundaries between code and documentation. Nearly forty years on, current tools such as Jupyter Notebooks [34] take this same approach, albeit more in a 'press to run' paradigm.

Around the same time that Thimbleby experimented with executable program documentation, he also began to explore novel paradigms for calculators [54] that used computation to do more for you. Applying *principles-driven design*, the calculator was 'equal opportunity' in that an equation such as '3+4=7' could be written '3+4=' with the computer supplying '7', or just typing '+4=7' or '3+=7' with the computer solving the equations to substitute '3' or '4' respectively. Follow-on work some years later allowed handwriting input on smart boards, ideal for classroom use [56].

Bret Victor's work addresses many of the issues raised earlier, indeed it is a quote from his '*Kill Math*' project at the beginning of this paper. This work includes envisionments and prototypes that explore ways in which the creation of code and the end-user visualisation of data can be made more accessible to a broad audience. In particular, in '*Up and Down the Ladder of Abstraction*' [60], Victor suggests creating programming paradigms that are constantly grounded in concrete problems and solutions, gradually increasing abstraction for certain aspects of a solution and then re-grounding in fresh examples, before generalising other aspects. While his strongest statements have been critiqued as a form of instant gratification, not expecting enough patience from children or adults [10], the general rule of connecting abstraction with concrete examples is commonly accepted as both a pedagogic and creative heuristic [46].

In addition, Victor's idea of '*Explorable Explanations*' [58] is also particularly relevant in this paper. This takes static documents with fixed scenarios and creates hand-crafted 'reactive documents' where the data values are editable allowing the reader to dynamically explore data and numerical models. This helps the reader's comprehension of complex models – it is often easier initially to understand the overall connection between cause and effect, and then dig into the details of how this comes about. In addition, the reader can find out about scenarios the writer never considered. Through these means Victor envisages an 'active reader', a sort of data equivalent of Barthes' 'writerly' (*scriptible*) texts [5].

This work also relates to end-user development / end-user programming research. Often this is focused towards ways of presenting standard programming in more visual manner, such as Scratch [47]; however according to Barricelli et al.'s systematic 2019 literature review [4], there are also a minority of spreadsheet-based interfaces building on business familiarity. The same survey showed only a tiny number of papers focused on personal use, and of these none aimed at numeric information.

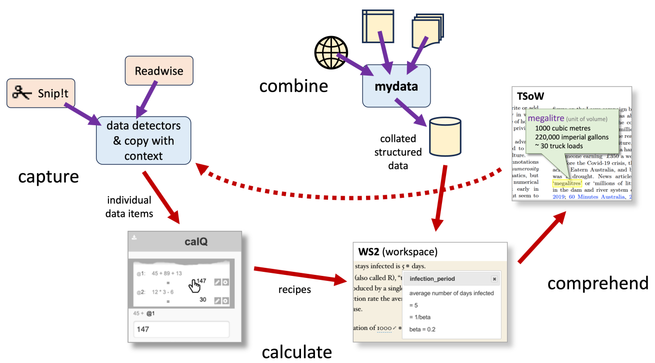


Figure 2: Numerosity tool ecosystem

3 DESIGN FOR A LIGHTWEIGHT NUMERIC ECOSYSTEM

3.1 Process

Figure 2 shows four aspects in the use of numeric and structured information and includes the new tools described in this paper (*in italics below*) as well as some existing tools.

Capture: Information is encountered in news items, reports, or in domestic settings such as receipts. Data items can be captured using the web clipping tool Snip!t [14] and imported from Readwise. The first of these is a research system that already has data detectors for numeric information and an extension mechanism to add actions for these, so easy to connect into the wider ecosystem. The latter, a commercial app, is accessed via its API, so numeric data detectors need to be applied to the extracted content; it is thus a proof-of-concept for other popular clipping and annotation tools such as Evernote. The example data-harvesting systems in this segment are all text based. However, iVolver [41] demonstrates that it is also possible to recover the underlying data from published graphical representations such as pie charts, even if the original data is not made available.

Combine: Often one wants to combine data from multiple sources, for example, regional health spending from a government report with population numbers and demographics from Census results. This is usually achieved by downloading the relevant data, performing some form of data cleaning or transformation and then using tools such as R to process the resulting data. Alternatively the data may be imported into a database and connected using join operations. Typically this relies on the data user to keep track of sources and to re-import the data if it is updated. *myData* allows users to record different data sources and to describe the way these should be transformed and recombined; whilst automatically maintaining the relation between derived data and sources, both in terms of provenance and live data.

Calculate: In some cases the capture and recording of data is sufficient, but often some form of calculation may be required. This may be as simple as adding up expenditure on certain categories, but may also include more complicated 'back of the envelope' models, such as those found in David Mackay's book [39]. Within this aspect *calQ* and *WS2* allow sums, formulae and more complex calculations. As a calculator *calQ* is more about doing instant calculations such as adding up a few numbers, and is focused on helping the end user. *WS2* deals with tables, structured data and models and is focused on presentation and explanation with (at present) more rudimentary authoring support.

Comprehend: Both *WS2* and *TSoW* aim to aid the understanding of data published on the web. *TSoW* is focused on individual numbers themselves, especially how to make sense of the very large or very small quantities such as 'megalitres'. *WS2* helps users to both expose and explore the calculations behind published data.

Crucially it is not simply the individual aspects that are critical, but the way in which they interlink, we will return to this in Section 5. As noted previously, the focus here is on data-in-the-small. For large data collection exercises it is easier (although still problematic) to create workflows that automate or at least document the various stages above, and also easier to justify the effort in doing this. The goal we want to address is how to make this possible in a more ad hoc way for more fragmentary data.

3.2 Principles

A number of design principles have driven or underlie the tools described below. Some are embodied in specific tools, but some relate to the collection as a whole. This principles-focused approach has a long history within HCI and innovation. On the academic side this includes Harold Thimbleby's generative user engineering principles (GUEPs) [52, 53] and Thomas Green's cognitive dimensions [8, 23] and commercial design examples include both the Xerox Star [50] that gave rise to the graphical user interface and the popular personal knowledge management tool Obsidian [44].

The first principle **incremental/smooth transitions** is most pervasive. In all aspects of learning, but especially numbers where people have existing anxiety, we need to create smooth paths between levels of knowledge and avoid barriers that need to be climbed to achieve the next stage. A personal inspiration for this comes from differential geometry, the branch of mathematics that is used to model gravity as curved space–time in Einstein's General Relativity. Given a curved surface, such as a sphere, differential geometry effectively covers it in small flattish patches, where the overlaps between patches are 'smooth' (in a mathematical sense!). You have to move from patch to patch, just as you may need to move between tools or notations, but there should be sufficient overlap to create comfort and confidence.

The second principle **the leaves are golden** [12] relates more to tabular and structured data in *WS2* and *myData*. People use many ways to collect and store their information. The usual information systems approach to this is to create a new all-encompassing system, pull in existing data and then expect users to adapt. The more user-centred approach is to accept as far as possible the existing way people manage their data and work with it. Sometimes there may need to be small adaptations, but the digital system should do the heavy lifting of dealing with multiple (sometimes weakly conflicting) kinds of heterogeneous data. Such a system may still import data in order to more effectively deal with it, but anything held centrally is a cache, the user's own data is golden.

The third principle is to interrelate, where possible, dual **intensional/extensional representations**; that is, whereever there is some sort of computation, to visualise both the formula/algorithm (intensional representation) and outputs (extensional representation) either simultaneously, or easily accessible. This reduces *hiddenness* – one of Green's earliest cognitive dimensions [23], but also increases comprehension as intensional and extensional representations allow different forms of critique. For example, in the formula "max(100,min(0,percent))", it is easy to see that we have the right maximum and minimum bounds for a percentage, but we might need to look at some examples of how the transformation works to realise we have min and max precisely the wrong way round (a common mistake the author makes).

Two other principles are closely related to this. Rapid incremental feedback and continuous representation have always been a key feature of direct manipulation [21, 49]. Where there are dual intensional/extensional representations, this implies the need for **immediate effect** as seen in spreadsheet recalculations or, albeit slightly indirectly, in notebook-style coding interfaces such as Jupyter notebooks. At a more theoretical level, this is also related to **code–data duality** the notion that in some senses computation and its output differ principally in *when* you look at them – variables are past computation remembered. This is very direct in lazy functional languages where outputs are often computation graphs that are evaluated when needed. Seymour Papert exploited this principle when teaching programming to young children [45]; the power of Turtle Graphics was that the execution of the Logo code was evident in the *trace* left behind on paper. This is equally true for all ages; indeed, it is often when students debug their code, running through it line by line, seeing the variables change, that they really understand the meaning of their code.

Finally, under-girding several of of the prototypes is the principle that **use is development**, or perhaps stated more directly: the best path to good code is though real use cases. For example, Knuth's literate programming arose from his need to manage the code of TeX, and TeX itself arose from his need to typeset his classic books of algorithms. More recently, start-ups often talk about "*eating our own dogfood*" [27], that is using their own software. In general, starting with concrete examples offers potential for both better communication and countering the '∀∃' problem of software that works for everything ... except anything useful. This principle certainly chimes with Victor's vision for greater access to mathematics and coding [60]. Looking forward, the presence of example-driven design offers many opportunities for AI interventions, for example suggesting automated generalisations or edge examples, or in a case-based fashion finding similar examples used elsewhere and hence potential solutions.

4 THE TOOLS

We will now look at the four prototoype tools in more detail. Their key attributes are summarised in Table 1 alongside other parts of the ecosystem that build on existing tools.

4.1 calQ – re-imagining a desk calculator

At first glance, calQ is just a web-based four function calculator, like many others but with the notable addition of a *virtual till roll*. It used to be common for electronic calculators used in shops and offices to print a record of calculations, making it easy to check past sums and also reuse numbers that had previously been calculated. This was lost in hand-held calculators with tiny fixed displays, but it seems strange that computer-based calculators, with large amounts of screen real estate, and phone-based ones where it is easy to scroll, do not all have this. In addition to being visually able to scroll back over past computations, the till roll enables past results to be copied into the current calculation using a single click or touch (Figure 1.ii). This largely obviates the need for explicit memory operations, which are always somewhat confusing. When the till roll gets too full a new roll can be started and the old one is saved, not unlike putting the cut-off paper roll in a draw or file.

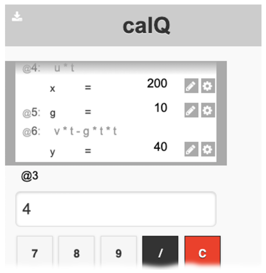


Figure 3: calQ showing named steps in a calculation

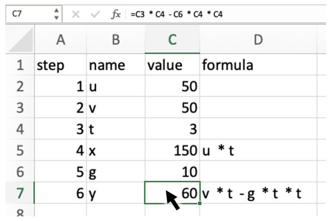
So far, this offers the same functionality as an old desktop calculator, but using the digital interface to reduce retyping, and indeed it is possible simply to use this functionality. However, when a value is copied from the till roll, as well as the numeric value, the formula includes a reference to the step it came from (e.g '@5' for step 5). Again this can be used on its own as an aide-mémoire to help trace the path of calculation (*intensional/extensional*), but also hints at further functionality. Previous steps in the calculation can be given a short name (and longer description) and the name is then automatically shown in any (past or future) formulae that reference it (see Fig. 3), making the till roll look more like a human description (e.g. 'amount = cost + cost × tax rate').

Past steps can also be edited to correct mistakes in the calculation. Crucially, the subsequent steps are recalculated, so, if the step's value had been copied into a subsequent step's calculation, the dependent value will be updated (*immediate effect*). That is the four-function calculator becomes a one-dimensional spreadsheet. A past till roll can also be copied to allow, for example, a weekly calculation to be redone simply by changing a few values.

Table 1: Tools in the current numerosity ecosystem

tool	use stage	development stage	description
calQ	calculation	deployed alpha	web calculator, allowing smooth transition from four function use to micro-spreadsheet
WS2	calculation and comprehension	configurable demo	allows spreadsheet features to be embedded in web pages, extensible block-based notation including nestable tables
myData	combination	configurable demo	collating, connecting, transforming and querying data from multiple sources
TSoW	comprehension	envisionment	creating human meaningful descriptions of numbers, such as "the size of Wales" often used as a unit if area in TV news
Snip!t	capture	existing tool	web page snipping and organisation, data detectors include extracting numeric data
harvesting	capture	early prototypes	accessing API feeds from Readwise and other note taking and and snipping services to scan for numeric data

Finally, the till roll can be turned into a fully reusable form in a number of ways. It can be transformed into an internal function by nominating one of the values entered as an input and one dependent step as an output. It can also be exported as a JavaScript function by nominating one or more inputs and outputs, or as a spreadsheet, which includes both the calculations in spreadsheet-executable form and the calQ version in a text column as documentation of the formulae (see Fig. 4).


C7 =C3 * C4 - C6 * C4 * C4

	A	B	C	D
1	step	name	value	formula
2	1	u	50	
3	2	v	50	
4	3	t	3	
5	4	x	150	u * t
6	5	g	10	
7	6	y	60	v * t - g * t * t
8				

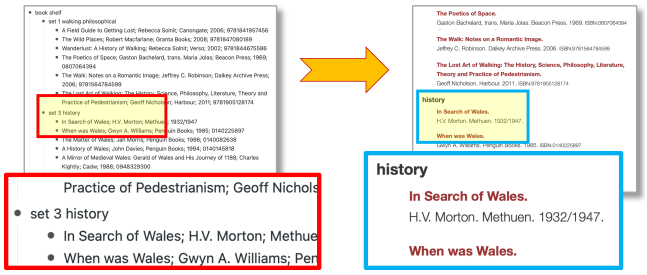
Figure 4: Spreadsheet exported from calQ. Note the Excel formula "C3 * C4 - C6 * C4 * C4" in the top edit window.

Note how calQ offers a *smooth transition* between four-function use and abstracted reusable code, albeit fairly simple. At each stage the 'code' or formulae are not abstract, they are doing a specific calculation on specific values (*intensional/extensional*), but they can be *abstracted* by reuse and so allow generalisation into formulae (*use as development*).

4.2 myData – collating your scattered data

In the office work you might have a CSV download of accepted papers with authors and titles on a conference management system that need to go onto a website, or you may repeatedly do the same transformations to the arcane spreadsheet you download from the university finance system. At home you may be organising a wedding with lists of guests or wanting to take the data from a spending tracking application to use to help make longer term financial decisions. In your research, you may find spreadsheets downloadable from government statistical services, tables in reports from a charity and results of your own surveys. Typically all this data is in different formats and spread over different machines and cloud services.

myData attempts to address these issues by creating a place to record, document, transform and combine the various multiple data sources, many quite small, that we live with day-to-day. The normal organisational approach when faced with multiple data sources is to import them into a single mega-system and then process things through that. Instead, myData follows the '*leaves are golden*' information design principle [12], where the primary sources, whether a spreadsheet that you keep on your laptop, or the data stored on government website are seen as primary. To be transformed or combined, data still needs to be read from these multiple sources, but this is viewed as a cache or working copy. Following the *leaves are golden* princple, the primary home, the place where that data is updated, is elsewhere. In essence, myData treats diverse sources as though they were an informal federated database [48].


- book shelf
  - set 1 walking philosophical
    - A Field Guide to Getting Lost; Rebecca Solnit; Canongate; 2006; 9781841957456
    - The Wild Places; Robert Macfarlane; Granta Books; 2008; 9781847080189
    - Wanderlust: A History of Walking; Rebecca Solnit; Verso; 2002; 9781844675586
    - The Poetics of Space; Gaston Bachelard, trans. Maria Jolas; Beacon Press; 1969; 0807064394
    - The Walk: Notes on a Romantic Image; Jeffrey C. Robinson; Dalkey Archive Press; 2006; 9781564784599
    - The Lost Art of Walking: The History, Science, Philosophy, Literature, Theory and Practice of Pedestrianism; Geoff Nicholson; Harbour; 2011; 9781905128174
  - set 3 history
    - In Search of Wales; H.V. Morton; Methuen; 1932/1947
    - When was Wales; Gwyn A. Williams; Penguin Books; 1985; 0140225897
    - The Matter of Wales; Jan Morris; Penguin Books; 1986; 0140082638
    - A History of Wales; John Davies; Penguin Books; 1994; 0140145818
    - A Mirror of Medieval Wales: Gerald of Wales and His Journey of 1188; Charles Kightly; Cadw; 1988; 0948329300

The Poetics of Space.
Gaston Bachelard, trans. Maria Jolas. Beacon Press. 1969. ISBN:0807064394

The Walk: Notes on a Romantic Image.
Jeffrey C. Robinson. Dalkey Archive Press. 2006. ISBN:9781564784599

The Lost Art of Walking: The History, Science, Philosophy, Literature, Theory and Practice of Pedestrianism.
Geoff Nicholson. Harbour. 2011. ISBN:9781905128174

history

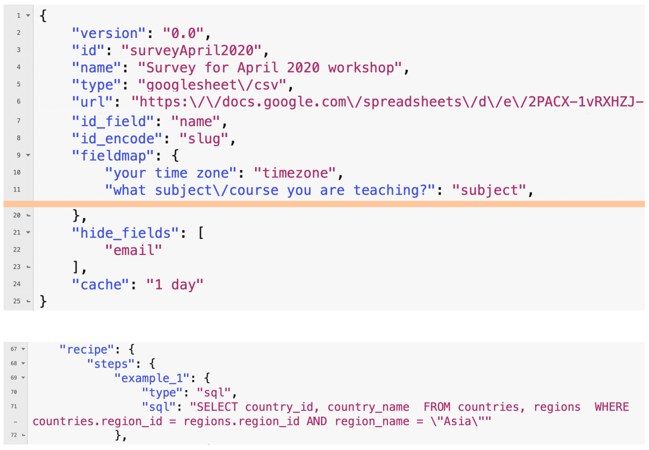
In Search of Wales.
H.V. Morton. Methuen. 1932/1947.

When was Wales.
Gwyn A. Williams. Penguin Books. 1985. ISBN:0140225897

Figure 5: myData: from WorkFlowy outline (left) to web page (right) – Note, the bibliographic entries in the WorkFlowy outline are delimited by semicolons, parsed by myData.

myData currently has no explicit front end. This is partly because the raw data is edited at its source and the final destination is usually a web page through WordPress plugins, or bespoke applications using myData services behind the scenes. An example of the former is where data on books is held in Workflowy (a popular PIM tool) which are then re-presented on a public-facing website (see Figure 5). An example of the latter is PhysProto, a system for creating video prototypes of physical devices [9, 62], where the description of the device behaviour and the locations of clips of video within longer videos are all held in a Google Doc spreadsheet.

myData transforms the various parts of the online spreadsheet into a structured form that can be easily used by the application.



```
{
    "version": "0.0",
    "id": "surveyApril2020",
    "name": "Survey for April 2020 workshop",
    "type": "googlesheet\/csv",
    "url": "https:\/\/docs.google.com\/spreadsheets\/d\/e\/2PACX-1vRXHZJ-
    "id_field": "name",
    "id_encode": "slug",
    "fieldmap": {
        "your time zone": "timezone",
        "what subject\/course you are teaching?": "subject",
    },
    "hide_fields": [
        "email"
    ],
    "cache": "1 day"
}
```

```
    "recipe": {
        "steps": {
            "example_1": {
                "type": "sql",
                "sql": "SELECT country_id, country_name  FROM countries, regions  WHERE
countries.region_id = regions.region_id AND region_name = \"Asia\""
            },
```

Figure 6: Transformation blocks in myData: (top) simple field renaming and hiding; (bottom) SQL query

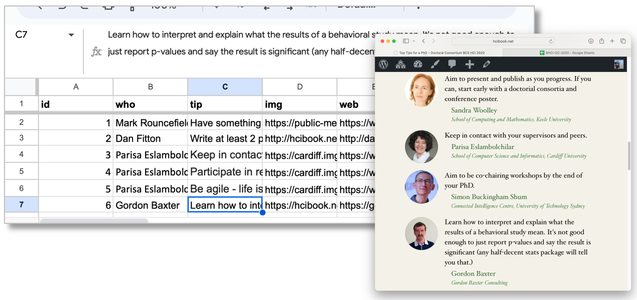
Under the hood, myData is built using nestable transformation blocks (Fig. 6). The set of blocks is extensible with each block configured using JSON as an abstract notation with no fixed concrete syntax. Some blocks provide access to different forms of data including Excel spreadsheets, CSV (including from Google Docs), and bespoke APIs such as WorkFlowy. These are where data sources are recorded and their structure described. Other simple blocks provide filtering and simple transformations, for example renaming fields or performing simple field decomposition (such as the semicolon delimited field in Figure 5). Further blocks allow different data sources to be combined including (should you so wish) with SQL queries. The last is included partly as proof of principle, but does mean that, for example, a local Excel spreadsheet (web-accessible using cloud service such as Dropbox), a Google Doc and a CSV document from an official website can be queried as if they were tables in the same database.

The output of myData is an API allowing access to the primary and derived data through a unified computational interface. It s thus an enabler of other applications on the data, including WS2, below. However, there is also a dedicated JavaScript library and WordPress plugin that allows data from the API to be formatted using templates, thus creating web pages that automatically update as the underlying data changes (Figure 7).

4.3 WS2 (workspace)

WS2 has its roots in two use cases.

The first is about computation. Often one makes a spreadsheet for some purpose, say a financial model, with several modifiable parameters, for example expected inflation rate and mortgage interest rate. This allows experimentation with the different parameters. However, in order to compare different scenarios, it is necessary to modify the parameters, and then copy critical results into cells of a different worksheet, creating the potential for copy-paste errors, and also meaning that the whole task has to be re-done if the model



C7 Learn how to interpret and explain what the results of a behavioral st
just report p-values and say the result is significant (any half-decen

	A	B	C	D	E
1	id	who	tip	img	web
2	1	Mark Rouncefiel	Have something	https://public-me	https://w
3	2	Dan Fitton	Write at least 2 p	http://hcibook.ne	http://da
4	3	Parisa Eslambolc	Keep in contac	https://cardiff.img	https://w
5	4	Parisa Eslambolc	Participate in re	https://cardiff.img	https://w
6	5	Parisa Eslambolc	Be agile - life is	https://cardiff.img	https://w
7	6	Gordon Baxter	Learn how to inte	https://hcibook.n	https://g

hcibook.net

Top Tips for a PhD – Doctoral Consortium BCS HCI 2020

BHCI-DC-2020 - Google Sheets

Aim to present and publish as you progress. If you can, start early with a doctorial consortia and conference poster.
Sandra Woolley
School of Computing and Mathematics, Keele University

Keep in contact with your supervisors and peers.
Parisa Eslambolchilar
School of Computer Science and Informatics, Cardiff University

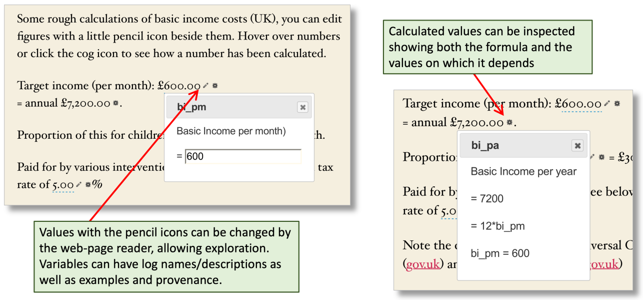
Aim to be co-chairing workshops by the end of your PhD.
Simon Buckingham Shum
Connected Intelligence Centre, University of Technology Sydney

Learn how to interpret and explain what the results of a behavioral study mean. It's not good enough to just report p-values and say the result is significant (any half-decent stats package will tell you that.)
Gordon Baxter
Gordon Baxter Consulting

Figure 7: myData: from Google Doc to web page.

changes. Although it is possible to use the spreadsheet's scripting language, this is a major step in terms of complexity. It seems as though it should be easy to transform the parameterised worksheet into something like a function.

The second use case is about presentation. Sometimes one wants to make a web-based calculator, say for wind resistance whilst cycling. Victor's '*reactive documents* are a richer example of this [58]. It is possible to do this by writing JavaScript code, potentially making use of a framework such as React. However, this is again a major step up in skills for the typical web content author.



Some rough calculations of basic income costs (UK), you can edit figures with a little pencil icon beside them. Hover over numbers or click the cog icon to see how a number has been calculated.

Target income (per month): £600.00
= annual £7,200.00 .

Proportion of this for childre ... h.

Paid for by various interventi ... tax rate of 5.00 %

bi_pm
Basic Income per month)
= 600

Values with the pencil icons can be changed by the web-page reader, allowing exploration. Variables can have log names/descriptions as well as examples and provenance.

Calculated values can be inspected showing both the formula and the values on which it depends

Target income (per month): £600.00
= annual £7,200.00 .

bi_pa
Basic Income per year
= 7200
= 12*bi_pm
bi_pm = 600

Figure 8: WS2 web page embedding – simple calculations

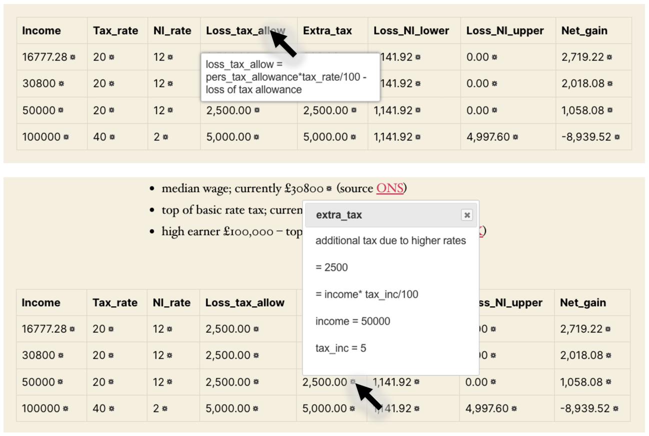
WS2 itself consists of two parts. The first, addressing the first use case, is an extensible block-based notation. Like myData's transformation blocks, WS2 uses a JSON-based abstract syntax, but with a plugin scheme for block visualisations and editors. WS2 has a *declarative* live-updating computational model, like that in a spreadsheet (*immediate effect*). The standard blocks include a table block that is rather like a spreadsheet table, except slightly more structured in that columns have default calculation formulae (see Figure 9) avoiding many of the unexpected problems when formulae are not copied correctly as tables are updated. Tables can be based on existing data (from myData API) or built as sequences with stopping criteria allowing fully visible loop-like behaviour, whilst still in an overall declarative framework. Note the *code–data duality* here – in this case a loop and a table. In addition, a group of variable blocks and tables can be bundled as a 'recipe' by declaring some variables as inputs or outputs. This can then be used function-like as a reusable computational block, including within cells of another table. Note that, like calQ, this is adopting a *use is development* paradigm, building abstraction bottom-up with examples of actual data at each stage.

The second part of WS2, addressing the second use case, is that WS2 can be embedded into web pages using a JavaScript library or WordPress plugin. Figure 8 shows a portion of a simple calculation page, which allows the reader to explore different levels of Universal Basic Income and its implications on tax rates and the national budget. This example is using the WordPress plugin which provides several shortcodes (surrounded by '[]', the normal way to embed additional features in WordPress pages) to enable the authoring of active pages. The authored text for the paragraph looks as follows:

```
Target income (per month): £[ws-var name="bi_pm"
description="Basic Income per month)" val="600" edit="true"
format="currency"/]
 = annual £[ws-var name="bi_pa" description="Basic Income per
year" expr="12*bi_pm" format="currency"/].
```

Note how each field has a short name and longer description, as in calQ. The first field (`bi_pm`) is set to be editable, allowing the user to update its value, whereas the second field is calculated (and recalculated) based on the first (*immediate effect*). In this case the calculated field follows the editable one, but, as in a spreadsheet, the underlying model is *declarative* and so a calculated field can be present on the web page *before* the values on which it depends. This is useful, for example, when some form of executive summary is required before more detailed workings.

In this example the formulae are given entirely in the WordPress shortcodes, that is the act of authoring the web page is implicitly creating the dynamic computation. In more complex cases recipes can be imported from other tools, including from calQ. Note too that whilst WordPress shortcodes are not the most usable way to author materials, but are familiar to users of WordPress (which powers more than 40% of all websites worldwide [20]).



Income	Tax_rate	NI_rate	Loss_tax_allow	Extra_tax	Loss_NI_lower	Loss_NI_upper	Net_gain
16777.28	20	12			141.92	0.00	2,719.22
30800	20	12			141.92	0.00	2,018.08
50000	20	12	2,500.00	2,500.00	1,141.92	0.00	1,058.08
100000	40	2	5,000.00	5,000.00	1,141.92	4,997.60	-8,939.52

loss_tax_allow =
pers_tax_allowance*tax_rate/100 -
loss of tax allowance

- median wage; currently £30800 (source ONS)
- top of basic rate tax; curren
- high earner £100,000 – top)

extra_tax

additional tax due to higher rates

= 2500

= income* tax_inc/100

income = 50000

tax_inc = 5

Income	Tax_rate	NI_rate	Loss_tax_allow			ss_NI_upper	Net_gain
16777.28	20	12	2,500.00			0	2,719.22
30800	20	12	2,500.00			0	2,018.08
50000	20	12	2,500.00	2,500.00	1,141.92	0.00	1,058.08
100000	40	2	5,000.00	5,000.00	1,141.92	4,997.60	-8,939.52

Figure 9: WS2 tables (top) formulae are attached to columns; (bottom) an individual cell can be interrogated showing the formula applied and the values from which it is derived.

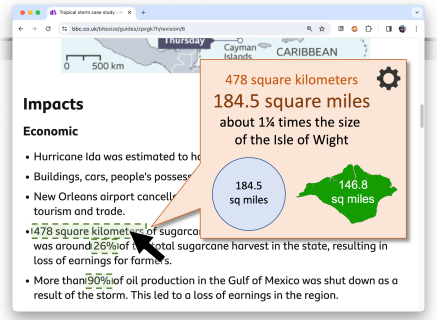
Tables can also be embedded as shown in Figure 9. As noted, columns may have a uniform formula, and both these column formulae and the calculations that give rise to individual cell values can be interrogated by the user (*intensional/extensional*), thus enhancing *explainability*. Of course, as the user experiments with different values (in this case different levels of basic income and changes to tax rates), the values in the table and also the explanations update correspondingly (*immediate effect*).

4.4 TSoW (The Size of Wales)

In the UK it is common for news presenters, when talking about large land areas, to use Wales or the Isle of Wight as units of measurement, for example, "*an iceberg the size of Wales*". Similar expressions are found elsewhere, for example an Australian documentary described the annual water loss in the Murray Darling River Basin as "*two Sydney-harbours full*" [1].

However, many numbers we read have no such day-to-day explanations, for example a BBC News web item described ice loss from Greenland as "*over 35,000 cubic metres of ice*" [40], which is about enough to fill a largish street with ice, hardly world changing. In fact the true amount was a million times more, the reporter had misread "*km*^{3}" in a UNESCO report [57] as meaning one thousand cubic metres rather than a cubic kilometre.

Imagine if the reporter or editor were able to see the everyday description alongside the number, both to communicate with the reader (when the figure is correct) and for their own understanding (when the figure is wrong!)


Tropical storm case study - H
bbc.co.uk/bitesize/guides/zpxgk7h/revision/6
Thursday
Cayman Islands
CARIBBEAN
0 500 km
184.5 sq miles
146.8 sq miles

Impacts

Economic

- Hurricane Ida was estimated to h
- Buildings, cars, people's possess
- New Orleans airport cancell tourism and trade.
- 478 square kilometers of sugarca was around 26% of t total sugarcane harvest in the state, resulting in loss of earnings for farmers.
- More than 90% of oil production in the Gulf of Mexico was shut down as a result of the storm. This led to a loss of earnings in the region.

478 square kilometers
184.5 square miles
about 1¼ times the size of the Isle of Wight

Figure 10: TSoW in action – the area of 478 square kilometers is presented in units familiar to the user (envisionment using extract from BBC Bitesize web page about Hurricane Ida [6]

TSoW does this. Figure 1.iii shows this principle being applied to the term "*megalitres*" in a report about the Australian water loss, that got the figure wrong by a factor of a thousand, and Figure 10 shows it being applied to the text of web material about the damage caused by Hurricane Ida, converting "*478 square kilometers*" into "$1\frac{1}{4}$ *times the size of the Isle of Wight*" (which has an area of 146.8 sq miles [32]) accompanied by (an envisionment of) a graphical view of the comparison. The informal scale comparisons are configurable; so, for example, the Italian option would read "*about twice the size of Elba*" (at 223 sq km [51]) as well as give sizes in square kilometres rather than square miles.

The current prototype is a small JavaScript plugin that can be included in a web page. It requires the web page author to add explicit markup:

```
<li><span class="tsow" tsow-dimension="area" tsow-number="478"
tsow-unit="square kilometers">478 square kilometers</span> of
sugarcane crops in Louisiana were destroyed. This was around 26%
of the total sugarcane harvest in the state, resulting in loss of
earnings for farmers.</li>
```

For authoring, this would ideally be replaced by a standardised markup. There is no suitable current schema.org [24, 25] or microformat [37] ontology. but RDFa [2] markup would be possible using the '*The Ontology of units of Measure*' [19], which is quite complete although, it appears, not currently maintained. For the reader this would ideally be integrated into a browser extension and/or data-detectors. Equally, the authored markup is itself harvestable, so could feed back into other tools notably calQ.

The informal scale configuration has a list of comparison objects for each type of data, for example for area 'football pitch', 'Isle of Wight', and 'Wales' or for fluid volume 'tanker load', 'Olympic swimming pool', or 'Sydney Harbour'. Each has a size which may be exact (e.g. height of Everest), or approximate with a range (e.g. truckload). The formal value is compared with the appropriate scale list, depending on configuration, and chooses the 'closest' but with a preference for larger multiples to very small fractions – for example, 300 billion kilometres would be described as "*about a hundred times the distance to Uranus, the furthest planet in the Solar System*", rather than "*one thirtieth the distance to Proxima Centauri, the closest star*". Multiples are rounded to approximate values with small fractions used where appropriate (e.g. $1\frac{1}{4}$ or 'one and a quarter' rather than 1.25), depending on configuration.

5 TOWARDS A RICHER ECOSYSTEM

We have seen the range of exploratory tools and how they connect to the design principles in Section 3.2. However, as has been emphasised, the most important thing is that these are an experimental ecosystem, they serve both to illustrate the space of potential tool support for lightweight numerical information and the way in which they can operate together. We have seen examples of this during the tool descriptions and Figure 2 illustrates some of these interconnection paths. The connections are of two kinds:

extensional feeds – The individual items from capture applications feed into *calQ*; data collated by *myData* can be used in *WS2*; *WS2* pages can link to *TSoW* and *TSoW* can be a source for other people's capture.

intensional feeds – The recipes (formulae) created in *calQ* and used in *WS2*; and some computational blocks created using *WS2* editors can be adapted (by hand at present) for use in *myData*.

Furthermore some are *live connections*, notably the external data – *myData* – *WS2* path, but others, including all of the intensional flows, are through copy and paste. Ideally more links would be live, at least in the sense that full provenance is retained so that one knows *where* the data originates, but also allowing, where desired, live update as sources change. For intensional feeds, one may want some form of check/approval if the source changes – of course one of the advantages of example-driven development is that there are effectively test cases that can be automatically used for differential verification when code changes. Existing code-sharing mechanisms, notably GitHub could be used for this, although ideally built into individual tools using the API, so that flows can be more seamless for end users.

If the underlying methods and infrastructure for these lightweight numeric applications are well established, then it is easy for research systems and commercial innovation to fill the gaps in the various phases above. However, there are barriers, notably that commercial interests often favour closed ecosystems, hence leveraging existing open infrastructure, such as GitHub, seems essential.

6 CONCLUSION AND FURTHER WORK

Each tool has novel features, and opportunities for further work. calQ is closest to being complete in itself and is now in an incremental improvement stage. WS2 is already deployed on some web pages, but requires far better authoring support to be generally useful; in addition it would benefit from more blocks focused on user-meaningful high-level data types, for example maps/floorplans as first-class objects. TSoW is a proof of concept, but a path to usable system is clear; the main focus will be on further customisation and ways to make real use easier (e.g. browser and WordPress plugins). Like WS2, myData is actively being used in deployed systems, but very much as a back-end technology and requires a far more usable configuration front end; this could include novel query mechanisms such as Query-by-Browsing, or Query-through-Drilldown featured at previous AVI [11, 15]

The main goal of this paper has to consider how the tools *together* can form more fluid flows, both of data (intensional) and formulae/code (extensional). In general, the former is more mature than the latter, but both require further work. For this calQ needs to go beyond copy–paste with better ways to copy individual numbers *with context* from all the other tools. Retaining context is essential for provenance, but will make generalisation easier, for example, if a calculation includes several numbers from a row in a table, then it is an obvious automated step to suggest applying it to to create a new computed column for all rows in the table. TSoW could likewise integrate more fully into each of the other tools to improve interpretation of data produced by them. The data connection from myData to WS2 is already in place as a feed into the data values for a table. However, the opposite direction, from the table recipe back to a more persistent re-usable data would enable a rich process of data use, reuse and sharing, as was possible in the now retired Google Fusion Tables [22].

This paper is intended to open up discussion. If it has prompted the reader to consider better tools, or ways that existing ones could be connected more deeply and more easily, it has done its job. We critically need to open up numeric information to the whole citizenry.

ACKNOWLEDGMENTS

Anonymised for review

REFERENCES

[1] 60 Minutes Australia. 2019. Scandal wasting millions of litres of water during Australia's worst drought. https://www.youtube.com/watch?v=SgF4W_oDrLA.
[2] Ben Adida, Mark Birbeck, Shane McCarron, and Ivan Herman. 2015. *RDFa Core 1.1*. W3C Recommendation. W3C. https://www.w3.org/TR/2015/REC-rdfa-core-20150317.

[3] Mark H. Ashcraft. 2002. Math Anxiety: Personal, Educational, and Cognitive Consequences. *Current Directions in Psychological Science* 11, 5 (2002), 181–185. https://doi.org/10.1111/1467-8721.00196

[4] Barbara Rita Barricelli, Fabio Cassano, Daniela Fogli, and Antonio Piccinno. 2019. End-user development, end-user programming and end-user software engineering: A systematic mapping study. *Journal of Systems and Software* 149 (2019), 101–137.

[5] Roland Barthes. 1975. *S/Z: An Essay.* Hill and Wang. English edition, R. Miller (Translator). Original French edition, 1970..

[6] BBC Bitesize. 2024. Tropical storm case study - Hurricane Ida. Accessed 5/1/2024. https://www.bbc.co.uk/bitesize/guides/zpxgk7h/revision/6.

[7] Ofer Bergman and Steve Whittaker. 2016. *The science of managing our digital stuff.* Mit Press.

[8] Alan F Blackwell, Carol Britton, Anna Cox, Thomas RG Green, Corin Gurr, Gada Kadoda, Maria S Kutar, Martin Loomes, Chrystopher L Nehaniv, Marian Petre, et al. 2001. Cognitive dimensions of notations: Design tools for cognitive technology. In *Cognitive Technology: Instruments of Mind: 4th International Conference, CT 2001 Coventry, UK, August 6–9, 2001 Proceedings.* Springer, 325–341.

[9] Anna R. L. Carter, Miriam Sturdee, and Alan Dix. 2022. Prototyping InContext: Exploring New Paradigms in User Experience Tools. In *Proceedings of the 2022 International Conference on Advanced Visual Interfaces (AVI 2022)* (Frascati, Rome, Italy). Association for Computing Machinery, New York, NY, USA, Article 22. https://doi.org/10.1145/3531073.3531175

[10] Channel TWo. 2017. Array []: Coding Slowly. In *Teaching Computational Creativity*, Veronika Tzankova Michael Filimowicz (Ed.). Cambridge University Press, Chapter 3, 75–44.

[11] Alan Dix. 2012. Asynchronous active values for client-side interactive service coordination. Association for Computing Machinery, New York, NY, USA, 26–33. https://doi.org/10.1145/2254556.2254565

[12] Alan Dix. 2016. The Leaves are Golden–putting the periphery at the centre of information design. Keynote at *HCI2016* https://www.alandix.com/academic/talks/HCI2016-the-leaves-are-golden/.

[13] Alan Dix. 2021. Qualitative–Quantitative Reasoning: thinking informally about formal things. In *Theoretical Aspects of Computing–ICTAC 2021: 18th International Colloquium* (Nur-Sultan, Kazakhstan (Virtual Event,), September 8–10, 2021). Springer, LNCS 13490, 18–35.

[14] Alan Dix, Tiziana Catarci, Benjamin Habegger, Yannis loannidis, Azrina Kamaruddin, Akrivi Katifori, Giorgos Lepouras, Antonella Poggi, and Devina Ramduny-Ellis. 2006. Intelligent context-sensitive interactions on desktop and the web. In *Proceedings of the international workshop in conjunction with AVI 2006 on Context in advanced interfaces.* 23–27.

[15] Alan Dix and Damon Oram. 2008. Query-through-drilldown: data-oriented extensional queries. In *Proceedings of the Working Conference on Advanced Visual Interfaces (AVI '08)* (Napoli, Italy). Association for Computing Machinery, New York, NY, USA, 251–259. https://doi.org/10.1145/1385569.1385610

[16] Ann Dowker, Amar Sarkar, and Chung Yen Looi. 2016. Mathematics anxiety: What have we learned in 60 years? *Frontiers in psychology* 7 (2016), 508.

[17] Ralph Mason Dreger and Lewis R Aiken Jr. 1957. The identification of number anxiety in a college population. *Journal of Educational psychology* 48, 6 (1957), 344.

[18] Education Scotland. 2017. Numeracy and Mathematics. In *Curriculum for Excellence:experiences and outcomes.* https://education.gov.scot/curriculum-for-excellence/curriculum-for-excellence-documents/experiences-and-outcomes/.

[19] Commit/ eFoodLab. 2024. *The Ontology of units of Measure (OM) 2.0.* Retrieved 6/1/2024 from http://www.ontology-of-units-of-measure.org/page/om-2

[20] Jack Flynn. 2023. *Wordpress Market Share + Statistics [2023]: How Many Websites Use Wordpress? Zippia*, dated Feb. 6, 2023. https://www.zippia.com/advice/wordpress-market-share-statistics/.

[21] David M Frohlich. 1993. The history and future of direct manipulation. *Behaviour & Information Technology* 12, 6 (1993), 315–329.

[22] Hector Gonzalez, Alon Y. Halevy, Christian S. Jensen, Anno Langen, Jayant Madhavan, Rebecca Shapley, Warren Shen, and Jonathan Goldberg-Kidon. 2010. Google fusion tables: web-centered data management and collaboration. In *Proceedings of the 2010 ACM SIGMOD International Conference on Management of Data (SIGMOD '10)* (Indianapolis, Indiana, USA). Association for Computing Machinery, New York, NY, USA, 1061–1066. https://doi.org/10.1145/1807167.1807286

[23] Thomas RG Green. 1989. Cognitive dimensions of notations. In *People and computers V (Proceedings of British HCI 1989)*, Alistair Sutcliffe and Linda Macaulay (Eds.). Cambrdge University Press, 443–460.

[24] R. V. Guha, Dan Brickley, and Steve MacBeth. 2015. Schema.Org: Evolution of Structured Data on the Web: Big Data Makes Common Schemas Even More Necessary. *Queue* 13, 9 (nov 2015), 10–37. https://doi.org/10.1145/2857274.2857276

[25] R. V. Guha, Dan Brickley, and Steve Macbeth. 2016. Schema.Org: Evolution of Structured Data on the Web. *Commun. ACM* 59, 2 (jan 2016), 44–51. https://doi.org/10.1145/2844544

[26] Mark Guzdial. 2010. *Alan Kay on Beyond Spreadsheets.* Retrieved 5/1/2024 from https://computinged.wordpress.com/2010/07/01/alan-kay-on-beyond-spreadsheets/

[27] Warren Harrison. 2006. Eating your own dog food. *IEEE Software* 23, 3 (2006), 5–7.

[28] Ray Hembree. 1990. The nature, effects, and relief of mathematics anxiety. *Journal for research in mathematics education* 21, 1 (1990), 33–46.

[29] John Holt. 1964. *How Children Fail.* Pitman Publishing Company.

[30] Don Hopkins. 2023. *Alan Kay on "Should web browsers have stuck to being document viewers?" and a discussion of Smalltalk, NeWS and HyperCard.* Retrieved 5/1/2024 from https://donhopkins.medium.com/alan-kay-on-should-web-browsers-have-stuck-to-being-document-viewers-and-a-discussion-of-news-5cb92c7b3445

[31] Hilary Hutchinson, Wendy Mackay, Bo Westerlund, Benjamin B. Bederson, Allison Druin, Catherine Plaisant, Michel Beaudouin-Lafon, Stéphane Conversy, Helen Evans, Heiko Hansen, Nicolas Roussel, and Björn Eiderbäck. 2003. Technology Probes: Inspiring Design for and with Families. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (Ft. Lauderdale, Florida, USA) *(CHI '03).* Association for Computing Machinery, New York, NY, USA, 17–24. https://doi.org/10.1145/642611.642616

[32] Isle of Wight Council. 2012. *The Island's Environment and Highways – Isle of Wight Facts and Figures 2011/12.* Report. https://www.iow.gov.uk/azservices/documents/2552-The-Islands-Environment-and-Highways-2011-12-v.2-April-2012-Done.pdf.

[33] William P Jones and Jaime Teevan. 2007. *Personal information management.* University of Washington Press.

[34] jupyter. 2024. *Project Jupyter.* Retrieved 5/1/2024 from https://jupyter.org/

[35] Alan Kay. 1984. Opening The Hood Of A Word Processor. Draft. Retrieved 5/1/2024 from http://worrydream.com/refs/Kay%20-%20Opening%20the%20Hood%20of%20a%20Word%20Processor.pdf

[36] Alan C. Kay. 1984. Computer Software. *Scientific American* 251, 3 (sept 1984), 52–59. Also published as Viewpoints Research Institute Technical Report TR-1984-001. https://web.archive.org/web/20171222015552/http://www.vpri.org:80/html/writings.php.

[37] Rohit Khare. 2006. Microformats: The next (small) thing on the semantic web? *IEEE internet computing* 10, 1 (2006), 68–75.

[38] Donald Ervin Knuth. 1984. Literate programming. *The computer journal* 27, 2 (1984), 97–111.

[39] David JC MacKay. 2008. *Sustainable Energy-without the hot air.* UIT. https://www.withouthotair.com/.

[40] Adrienne Murray. 2022. Climate change: Greenland's culture shifts as Arctic heats up. *BBC News*, 12 October 2022. https://www.bbc.co.uk/news/world-europe-63135211.

[41] Miguel A Nacenta and Gonzalo Gabriel Méndez. 2017. ivolver: A visual language for constructing visualizations from in-the-wild data. In *Proceedings of the 2017 ACM International Conference on Interactive Surfaces and Spaces.* 438–441.

[42] National Numeracy. 2021. Counting on the recovery: The role for numeracy skills in 'levelling up' the UK. https://www.nationalnumeracy.org.uk/news/counting-on-the-recovery.

[43] National Numeracy. 2023. A third of adults are nervous about numbers. https://www.nationalnumeracy.org.uk/news/third-adults-are-nervous-about-numbers.

[44] Obsidian. 2024. *About Obsidian: Manifesto.* Retrieved 19/1/2024 from https://obsidian.md/about

[45] Seymour Papert. 1980. *Mindstorms; children, computers and powerful ideas.* New York: Basic Book. Republished 1993, 2020.

[46] George Polya. 2014. *How to Solve It: A New Aspect of Mathematical Method.* Princeton University Press. First published 1945.

[47] Mitchel Resnick, John Maloney, Andrés Monroy-Hernández, Natalie Rusk, Evelyn Eastmond, Karen Brennan, Amon Millner, Eric Rosenbaum, Jay Silver, Brian Silverman, et al. 2009. Scratch: programming for all. *Commun. ACM* 52, 11 (2009), 60–67.

[48] Amit P Sheth and James A Larson. 1990. Federated database systems for managing distributed, heterogeneous, and autonomous databases. *ACM Computing Surveys (CSUR)* 22, 3 (1990), 183–236.

[49] Ben Shneiderman. 1983. Direct manipulation: A step beyond programming languages. *Computer* 16, 08 (1983), 57–69.

[50] David Canfield Smith, Charles Irby, Ralph Kimball, and Eric Harslem. 1982. The Star user interface: An overview. In *Proceedings of the June 7-10, 1982, national computer conference.* 515–528.

[51] The Editors of Encyclopaedia. 2023. *Elba. Britannica*, last updated by Amy Tikkanen, 9th October 2023. https://www.britannica.com/place/Elba-island-Italy.

[52] Harold Thimbleby. 1984. Generative user engineering principles for user interface design. In *Proceedings of INTERACT '84.* North-Holland, 661–666.

[53] Harold Thimbleby. 1985. User interface design: Generative user engineering principles. In *Fundamentals of human–computer interaction.* Elsevier, 165–180.

[54] Harold Thimbleby. 1995. A new calculator and why it is necessary. *Comput. J.* 38, 6 (1995), 418–433.

[55] Harold Thimbleby. 2003. Explaining code for publication. *Software: Practice and Experience* 33, 10 (2003), 975–1001.

[56] Harold Thimbleby and Will Thimbleby. 2007. Mathematical mathematical user interfaces. In *IFIP International Conference on Engineering for Human-Computer Interaction*. Springer, 520–536.

[57] UNESCO. 2004/2019. *Ilulissat Icefjord*. Retrieved 8/1/2024 from https://whc.unesco.org/en/list/1149/

[58] Bret Victor. 2011. *Explorable Explanations*. *Worrydream*, dated March 10, 2011. http://worrydream.com/ExplorableExplanations/.

[59] Bret Victor. 2011. *Kill Math*. *Worrydream*, dated April 11, 2011. http://worrydream.com/KillMath/.

[60] Bret Victor. 2011. *Up and Down the Ladder of Abstraction*. *Worrydream*, dated October 2011. http://worrydream.com/LadderOfAbstraction/.

[61] Sally Weale. 2015. Numeracy crisis threatens to hold back UK in global data race. *The Guardian* 25 Jun 2015 (2015). https://www.theguardian.com/education/2015/jun/25/numeracy-crisis-threatens-uk-global-data-revolution

[62] A. Wlide and A. Dix. 2020. Navigating Challenges on Wide-scale Adoption of Video for HCI Education: The HCIvideoW Experience. *COVID-19 Case Studies at ACM Learning at Scale (L@S2020), 13th August 2020.* http://alandix.com/academic/papers/LatS-hcivideow-exp-2020/.