Big Brother is Watching You—Now in a Doubleplusgood Way

Corey Sterling
University of Waikato
Hamilton, New Zealand
Email: cts16@students.waikato.ac.nz

Carlin St. Pierre
Atlassian
Sydney, Australia
Email: cstpierre@atlassian.com

David Bainbridge
University of Waikato
Hamilton, New Zealand
Email: davidb@waikato.ac.nz

***Abstract*—This paper reports on the progress made with our X Marks The Spot (XMTS) system. In XMTS we make a fundamental change to the compositing operation in a desktop manager, storing all text-display events and their related raster-draw events. This means we can provide the user with a search interface that allows them to go back in time and view how their desktop used to look. Because XMTS is aware of what text was drawn into which window and at what position, the provided snapshot is semi-interactive in that the user can bring individual windows forwards and backwards, and copy text from them. In previous work we demonstrated the approach is technically feasible. In this paper we detail infrastructural advances we have made and showcase its new search interface.**

I. Introduction

Having just closed a window or tab, how many times have you later come to regret the action—often only moments later—because it showed you something you needed to look at again. Sometimes it is possible to recover the step, through undo or view history operations, but it is seldom a simple or convenient action, and—more often than not—not an option at all. It's gone! The only way to get it back is to reconstruct the steps you went through in the first place, something that increases in difficulty the more time that has elapsed.

Now imagine a desktop environment that allows you to scrub back through time, like a video player, and view how the desktop had previously been. Add to this the ability for the desktop environment to be indexing the text that is drawn to the screen over time—no matter which application was responsible for producing it—and this imagined system can then also allow you to specify query terms for searching back through previous desktop screens to pinpoint the specific moments where information was displayed that you wish to refer to once more.

We reported on our initial findings from developing software to do this in an earlier article [1]. Utilizing the open source X-Windows system for Unix, we showed the approach is technically feasible by binding in digital library back-end components into the Glyph Compositor of the desktop/windowing environment. The addition of processing the live stream of text events along with screen capture of the window it was drawn to increased the computational cost of the windowing server by roughly a factor of two, something that is comfortably accommodated on a modern desktop PC or laptop.

We named the system XMTS for X Marks The Spot, which we retrospectively label here, Mark I. With the viability of the approach established, we undertook a redevelopment of the software, Mark II, which we present in this current paper. In Section II we describe the infrastructural changes that have been made, followed by presenting an example (Section III) of the updated search interface.

II. Technical Advancements

Our evaluation of XMTS Mark I highlighted two key areas that would benefit from technological improvement:

- the unexpectedly high computational cost of taking image snapshots; and
- glyph code to Unicode mapping errors that were effecting some Roman alphabet characters, and the majority of non-Roman alphabet languages.

Image Snapshots. We were expecting the raster operation of XMTS Mark I to be expensive in terms of storage requirements, but in fact it turned out to be significantly more computationally expensive as well, in comparison to the full-text indexing task. In XMTS Mark II we have changed image capture to be derived from the desktop environment's damage events, rather than snapshotting an entire window every time a text event occurs.

Based on how the user interacts with the desktop, damage events are the sub-regions of the screen that need to be redrawn. It is the mechanism used by the desktop window manager to optimize the redrawing of the screen. XMTS Mark II takes advantage of the same savings by taking snapshots of these damage areas rather than entire windows. The new approach is not a zero-cost change, however, as now several images typically need to be composed together to form a complete window to show to the user. XMTS Mark II uses MongoDB to store window ID and region information as the damage events occur. When the search interface needs to generate a snapshot for the user, this database is queried to determine which captured images need to be composed together. To further optimize the composite calculation, we make use of R-trees [2] to determine when a damage event is not visible due to a newer damage event occluding it. MongoDB also supports full-text indexing, so we opted to utilize this as well, rather than retain the use of Solr that had been embedded in Mark I.

Glyph code mappings. Unlike the Unicode standard, which provides round-trip compatibility with existing character encodings, the rendering of glyphs (characters) in a desktop environment has no requirement to do so. Glyph code-point tables, for example, regularly skip non-printable characters; they may also use a non-standard value when the rendering of a particular character has been deprecated in favor of a newer one. This can lead to mismatches between the integer code-points used for glyphs and their logical Unicode code-points, which are needed for indexing.

In XMTS Mark I we made use of an ad hoc calculation that worked well in practice for ASCII based characters, but became less reliable for characters with code-points above this numeric range. In XMTS Mark II we have developed a more refined calculation: when the XMTS indexer server is started, we use the FreeType library to compute a reverse-mapping lookup table that—for a given font—allows us to take a glyph code-point value and look up its Unicode value.

III. Updated User Interface

With the introduction of capturing damage events, an important new ability in XMTS Mark II is that it can produce a continuous representation of desktop images—synthesizing a snapshot for any given point in time—as opposed to only the discrete points of time that the Mark I version could do, where a text event occurred.

This difference manifests itself in many ways in the new interface. Figure 1 shows a snapshot of it in use, showing the desktop interactions of one of the authors during a period of activity where they confirmed the dates of the JCDL conference on-line, and then entered this information into their calendaring system. Next they made use of Google to hunt for images that would fit in with an idea that was forming as to how a "minute madness" slide might convey the essence of the project. Having downloaded a variety of images, they then started to make use of an image/photo application to refine their selections.

As with XMTS Mark I, in the new interface a time-line view is given along the bottom portion of the interface, with the upper portion used to display a particular snapshot. Again, like the previous version, the time-line can be zoomed in and out and panned, however in other regards things are considerably changed. In particular the time-line is no longer a single line with snapshot events attached to it. Instead, with the ability to recreate the desktop for any point in time, greater use of the y-axis is made: each application running is depicted with a continuous horizontal bar to show when it was launched and quit. The start of the bar is decorated with an icon for the application along with some descriptive text. These pieces of information are stored as part of a window's data structure (metadata if you will) and is straightforward to extract.

The user is free to drag and drop the main (mid-blue) viewer line anywhere on the time-line, and after a short delay, the relevant screenshot of their desktop is displayed. Alternatively they can use the search box, positioned in the bottom-left corner. In Figure 1 the user has entered the query

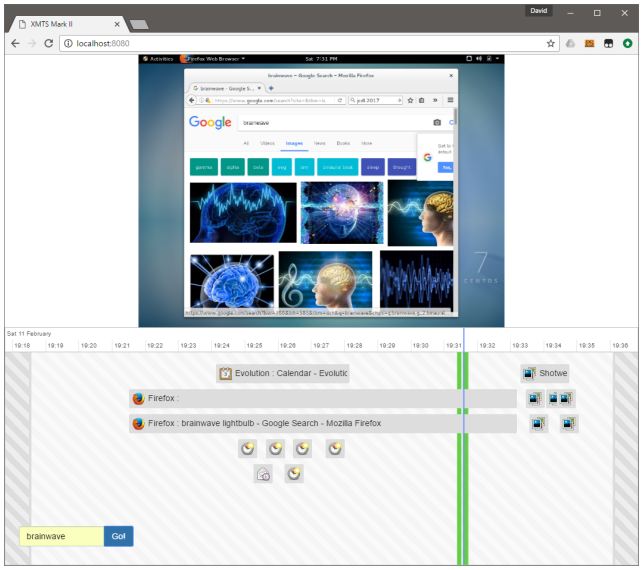

Fig. 1. The new searching and browsing interface in XMTS Mark II.

term *brainwave*, recalling part of the Google Image search activity they undertook. Green lines are added to the time-line visual to show locations where matches to this term occurred. Our user has moved the viewer line to a position near the end of the marked section. For the snapshot that is displayed, the new form of time-line gives us an added bonus: clicking on one of the horizontal application bars causes that window to be brought to the top of the snapshot for easier viewing. From here a rectangular region can be dragged out, and text that falls inside it selected.

In Mark II we have also dispensed with the additional search boxes for controlling date/time ranges and the applications in which to search. Given the new spatial characteristics of the time-line layout, a comparable end-result can now be accomplished through direct manipulation of pan and zoom to achieve the desired date/time range displayed horizontally, and a particular application of interest brought into focus vertically.

IV. Conclusions

In conclusion, this paper details a set of developments to XMTS that improves computational efficiency, and provides better handling of the glyphs (especially for non-English languages). The most significant change is the switch to using damage events, rather than snapshotting entire windows. Not only does this reduce the computation cost of capturing windows, it also allows for the generation of the desktop for any given point in time, something that was not possible before.

References

[1] C. S. Pierre, D. Bainbridge, and B. Rogers, "Big brother is watching you—but in a good way," in *IEEE/ACM Joint Conference on Digital Libraries*, Sept 2014, pp. 277–280.

[2] A. Guttman, "R-trees: A dynamic index structure for spatial searching," in *Proceedings of the 1984 ACM SIGMOD International Conference on Management of Data*, ser. SIGMOD '84. New York, NY, USA: ACM, 1984, pp. 47–57.