Data as Script:

A Critical Lens for Interface Production

Geoff Hinchcliffe, Assistant Professor in Graphic Design, University of Canberra

Introduction

The advent of Web 2.0 has seen the proliferation of "feeds" and "application programming interfaces" (APIs) that allow data to be shared across platforms and integrated into different applications. At the time of writing, the website programmableweb.com lists 3580 web-based APIs. Aside from the practical benefits of data access, APIs in particular make explicit the distinction between application and interface. In traditional desktop software this delineation has not been apparent, with graphic interface, application logic and data all created by the same author/company and presented as a single definitive view. By providing access to application data and logic, APIs enable independent designers to create their own custom representations. In this way the API is akin to the script from theatre or cinema; with API data and logic serving as the common source to be interpreted and realised by different producers. As a means of investigating data as script, this paper uses the concept as a critical lens in the analysis of different interface realisations of a common API "script".

The Script

Twitter is a popular micro-blogging platform in which its users post short status updates of 140 characters or less. In addition to posting status updates, users follow other users and receive an aggregated feed of tweets from their network. All of the examples cited in this paper are representations of Twitter, and access the Twitter service through its Application Programming Interface (API). In simple terms the Twitter API is a web-based application that receives requests and returns relevant data. It provides a non-graphical interface to almost all of Twitter's data and functionality. A typical API transaction sees a client application requesting the 20 latest tweets of a user and their network of friends. The Twitter web servers process the request and return any matching data as a structured list. The client application then parses the data, formats it, and presents it to the user.

```
{
  "coordinates": null,
  "favorited": false,
  "created_at": "Fri Jul 16 16:58:46 +0000 2010",
  "truncated": false,
  "entities": {
    "urls": [
      {
        "expanded_url": null,
        "url": "http://www.flickr.com/photos/cindyli/4799054041/",
        "indices": [
          75,
          123
        ]
      }
    ],
    "hashtags": [

    ],
    "user_mentions": [
      {
        "name": "Stephanie",
        "id": 15473839,
        "indices": [
          27,
          39
        ],
        "screen_name": "craftybeans"
      }
    ]
  },
  "text": "got a lovely surprise from @craftybeans. She sent me the best tshirt ever. http://www.flickr.com/photos/cindyli/4799054041/ ::giggles::",
  "annotations": null,
  "contributors": null,
  "id": 18700887835,
  "geo": null,
  "in_reply_to_user_id": null,
  "place": null,
  "in_reply_to_screen_name": null,
  "user": {
    "name": "cindy li",
    "profile_sidebar_border_color": "AD0066",
    "profile_background_tile": false,
    "profile_sidebar_fill_color": "AD0066",
    "created_at": "Wed Nov 29 06:08:08 +0000 2006",
    "profile_image_url": "http://a1.twimg.com/profile_images/553508996/43082001_N00_normal.jpg",
    "location": "San Francisco, CA",
    "profile_link_color": "FF8500",
    "follow_request_sent": false,
    "url": "http://www.cindyli.com",
    "favourites_count": 465,
    "contributors_enabled": false,
    "utc_offset": -28800,
    "id": 29733,
    "profile_use_background_image": true,
    "profile_text_color": "000000",
    "protected": false,
    "followers_count": 3395,
    "lang": "en",
    "notifications": true,
    "time_zone": "Pacific Time (US & Canada)",
    "verified": false,
    "profile_background_color": "cfe8f6",
    "geo_enabled": true,
    "description": "Just me, Cindy Li.Giving cute substance since 1997.\r\nMarried to @themattharris.\r\nProduct designer for Yahoo! ",
    "friends_count": 542,
    "statuses_count": 4847,
    "profile_background_image_url": "http://a3.twimg.com/profile_background_images/3368753/twitter_flowerbig.gif",
    "following": true,
    "screen_name": "cindyli"
  },
  "source": "web",
  "in_reply_to_status_id": null
},
```

As can be seen in Figure 1, the API data can be read in its raw form but the intention is that it will be interpreted, reformatted and presented to the user. The notion of creating a work to be mediated by independent producers rather than published for a general audience, aligns the API conceptually with the script (screenplay/play).

"Script: the working text, manuscript, or the like, of a play, film, television program, etc." (ed. Yallop, 2004)

While both script and API are created for producers, neither provides elaborate specifications for how a production should be realised. Typically, the script for theatre or cinema will include only rudimentary suggestions regarding its production. Instead of detailed descriptive prose the script writer uses subtext, action and symbolism to indicate a character's motives and emotions. Michael Holt describes the theatre designer's reading of a script as a forensic process in which they must identify significant settings, objects and costumes whether specified or implied: "Some details are hidden in the text. They may be revealed only in the dialogue, when a character refers to them. Sometimes they are there solely by inference, not actually mentioned, but nonetheless needed for the dialogue to work in action. Great care is needed to identify these items and decide how important they are." (pg. 37)
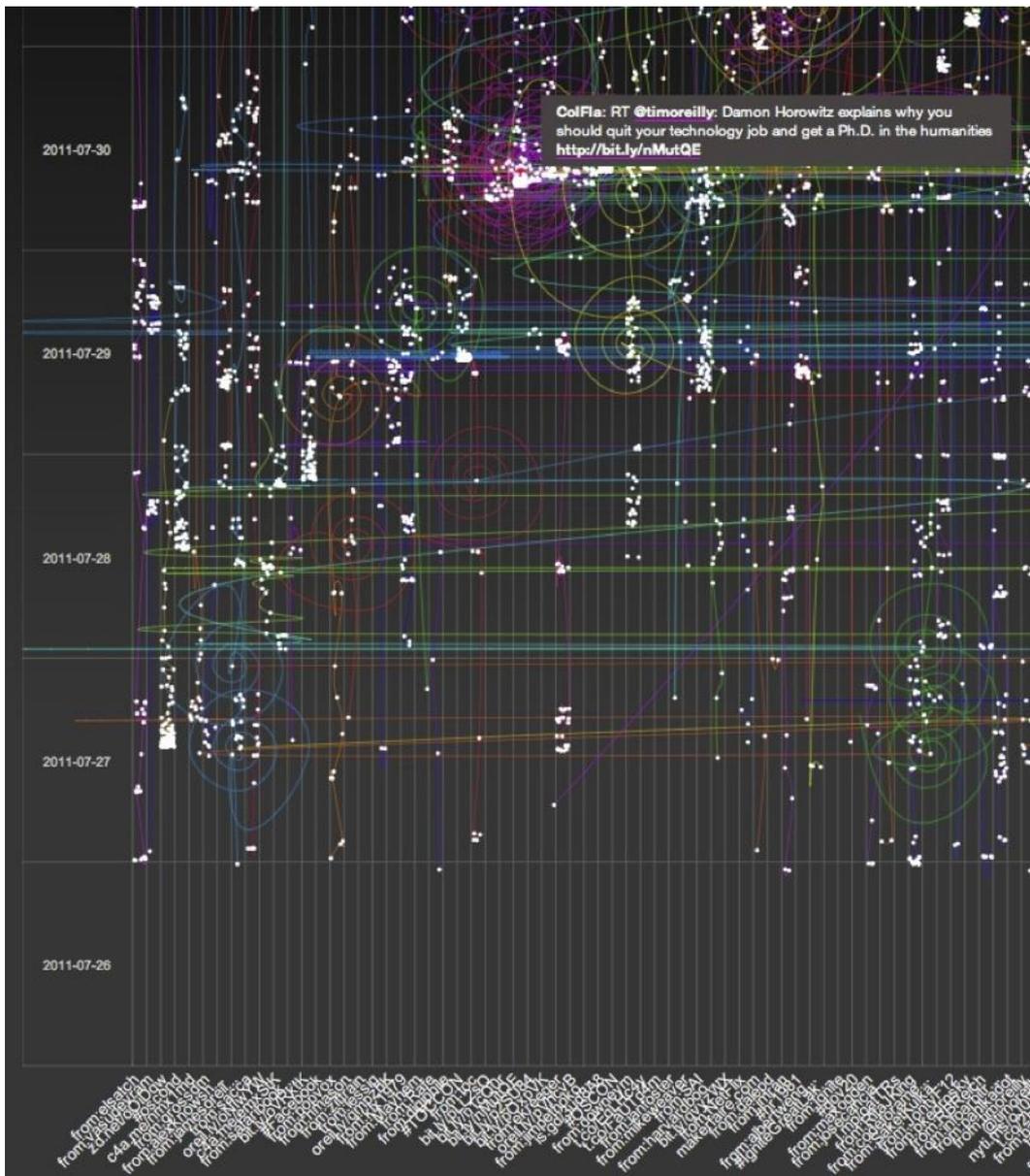
In applying the script analogy to Twitter's API, it is tempting to correlate Twitter users with characters, and status updates with dialogue. Certainly this correlation could yield some interesting results, but it would not hold when considering non-social and non-verbal API data. In any case, it must be noted that the script generated by Twitter's API is dynamic. Each successive set of tweets contains different status updates, as well as different users with different interrelations. Producers for theatre or cinema will drill down to the semantics of individual words in the process of analysing a script, and clearly this is not possible when the script is being generated dynamically. Instead, interface designers must focus on the constant elements within the script: its structure and data types. Within the Twitter API (Figure 1) the structure and data types are clearly identifiable nodes with corresponding data: the "text" node contains the status update; the "created_at" node contains the creation date; the "source" node lists the application used to post the status; the "user" node contains a subset of nodes with information about the author of the update; etc. So while the exact data is unknown to the producer, the structure is defined and can be used to construct a model to serve as script. As we will see in the examples that follow, there is great scope for interpretation in the adaption of script to interface.

Conversations and Replies

Context is key. Tweetbot makes it effortless to catch up on a conversation or see responses to a tweet. Just swipe to the right to view a conversation thread or to the left to view replies to a tweet.

The "TweetBot" iPhone application (Tapbot, 2011) (Figure 2) adopts what has become the standard reading of the Twitter script. Its chronological presentation of a user's timeline includes the text of the tweet, the author's details, and the time of publication. Less conventionally, the app uses the content of the "in_reply_to" nodes to reorganise tweets into conversations, revealing dialogues that would otherwise be disrupted and obscured by unrelated intervening tweets. While by no means radical, the conversations example shows how an alternative reading of the source can produce novel perspectives of it.

In "Social Collider" (Figure 3), Schmidt and Pohflepp take a network view of the script. Describing the work they state: "One can search for usernames or topics, which are tracked through time and visualized much like the way a particle collider draws pictures of subatomic matter. Posts that didn't resonate with anyone just connect to the next item in the stream. The ones that did, however, spin off and horizontally link to users or topics who relate to them, either directly or in terms of their content. The Social Collider acts as a metaphorical instrument which can be used to make visible how memes get created and how they propagate. Ideally, it might catch the Zeitgeist at work." (2009) To establish links between tweets, the designers use "hashtags", "user_mentions", and "in_reply_to" in combination with the tweet "text". They then use the "created_at" node to map the connections over time. Social Collider's mapping of tweet diffusion provides a unique illustration of network narratives. Where conventional readings place the user at the centre being fed tweets, the Social Collider positions the user within a complex constellation, revealing the energy of their influence within that network.

Like the designers of Social Collider, the producers of "FlipBoard" (Figure 4) prioritise nodes other than the tweet "text" in order to create a very different view of a user's timeline. Using the "urls" node as a source, website excerpts, videos, and photos are reformatted into a magazine layout. FlipBoard organises its content by time but also by size in its bid to form convincing page spreads. In addition to the magazine layout the novelty of the app is that the content is curated by a user's Twitter network. In their rendition, the FlipBoard producers prioritise the media embedded within tweets and create a rich magazine experience, but in the process depersonalise the script by concealing the accompanying comments. FlipBoard treats the script as an aggregation of links and perpetuates this view by supporting easy retweeting of the content from within the app.

In each of the examples the designers combine and prioritise particular attributes of the script to provide an audience with a unique perspective of it. Producers can also take a more critical approach to the script. "Twistori" (Hoy & Fuchs, 2008) (Figure 5) and "Murmur Study" (Baker, 2009) (Figure 6) read the

script as a cacophony of voices in which none can be heard. Twistori's interface consists almost entirely of statements commencing with one of six verbs: I love; I hate; I think; I believe; I feel; I wish. Murmur Study filters tweets using verbs such as "argh, meh, grrrr, oooo, ewww, and hmph" and renders them with twenty thermal printers mounted high upon a gallery wall. While sometimes amusing and occasionally surprising, the unrelenting stream of statements merge into noise. In the case of Murmur Study, this is physically represented as an ever increasing pile of tangled ribbons. Because neither of the works offer an interface for response or discovery, the audience is distanced from the dialogue, impotent in the face of an irrepressible outpouring.

Like Murmur Study, "Tweetleak" (Wunderling, 2008) (Figure 7) involves the physical production of tweets. The work takes the form of a large sealed box with a single letterbox-like slot. The box is placed in a public place and aggregates tweets from nearby, materialising them in the form of small printed adhesive strips which appear from the slot in the top surface. The author of the tweet is notified via Twitter when their tweet has been printed and when it has been taken away. Wunderling uses physical artefacts to explicate the "sharing" of information that occurs invisibly in the network, and draws our attention to the lack of control we have over our digital data.

As the examples have shown, the data script is open to multiple readings and representations, of practical or critical orientation. The first three examples involve a form of characterisation of the Twitter API script. The suggestion here is not that the API nodes are characters, but that the interface producers create novelty by prioritising particular elements, or emphasising relationships in much the same way that a theatre director will work with characters and relationships in a play. Alternatively, the final three works

use interface like stage setting, the aesthetics of which are not explicit but based on the broader themes that the designer identifies within the script.

A more difficult aspect when applying the analogy of script to interface design is the dynamic nature of the user's contribution. The examples show that while the user content of the tweet is unknown, the designers are able to use the data structure and data types to forecast the content within a workable range. For example, the "text" node will consist of text and be between 1-140 characters in length, "created_at" node will consist of a date string in a specific format, etc. In some respects the dynamic user content is akin to improvisation in theatre or cinema. For example, where the origin and outcome of a scene are defined but the dialogue and action are invented by the actors. Like the Twitter API, the improv script is in part dynamic but provides producers with enough structure and detail to work with.

Critical Evaluation

In the previous section I argued that a script is a foundation structure rather than a complete blueprint, requiring interpretation and adaption in order to be realised in a mediated form. In this section I will argue that interpretation and adaptation of the script is also central in critical evaluation of script-based productions.

In his production of Shakespeare's "Richard III" for London's Old Vic theatre, Sam Mendes' realisation "steers a clear course through the internecine drama. It is in modern dress but unspecific about its exact setting, which allows Spacey to channel touches of dictators through the ages" (Lawless, 2011). The setting "reminds us how today's dictators seek spurious constitutional legitimacy and become skilful media manipulators" (Billington, 2011). Alternatively, the Telegraph reports that while the production is "fluent and lucid, it lacks the striking invention and disconcerting dreamlike atmosphere of Edward Hall's production now running at Hampstead. Like Tom Piper's architectural design of receding walls and doors, Mendes's staging feels a touch obvious and over-deliberate, leaving little room for the audience to let their own imaginations soar." (Spencer, 2011).

From these commentaries we are able to appreciate the practical and poetic significance of the producer's interpretation of the script. With its unspecific, yet modern setting, Mendes' production amplifies themes within the script and allows his audience to connect a centuries old text with contemporary characters and events. And while it may seem incongruous (if not dangerous) to relate a Twitter interface to a Shakespeare production, reviews of the TweetBot app have commonalities with theatre reviews listed above, particularly in their framing of the interface as a novel realisation of a familiar source, as in Lofte's review entitled "Tweetbot once again redefines Twitter for iPhone". Consistent amongst the many TweetBot reviews is a strong appreciation of its interface aesthetics:

"The interface is what stands out the most about the application. This is something we have come to expect from Tapbots, who have a reputation for crafting gorgeous looking applications." (Naik, 2011)

"From Tapbots – the creators of Convertbot, Pastebot and Weightbot – Tweetbot promises to be a full-featured Twitter client with a lot of personality; boasting a meticulously crafted user interface, smart gestures, and customizable navigation." (Bell, 2011)

"There is this addictive cleverness and playful uniqueness to the way Mark and Paul (the app designers) build their apps. The sounds, the animations, and graphics don't feel or act like a standard app, they feel more like a toy. A toy you get to use for work." (Blanc, 2011)

The positive appraisal of the interface's playful antics is typical, as is the identification of the designers of the work. Another key aspect of the reviews for both Richard III and TweetBot is the highly contextual nature of the evaluation. In the reviews of Mendes' Richard III the production is understood within a London Shakespearean theatrical context, while the reviewers of TweetBot write for an audience already familiar with a host of existing mobile Twitter applications. In both cases, the works are assessed as independent productions and as critical contributors in their respective fields.

Importantly for interface design, is the reviewers' appreciation of the poetics of the adaptation. While such a proposition is unremarkable for theatre or cinema, it is novel for computer interface design. To understand its significance we must consider the HCI pragmatics that have defined production aims and audience expectations of interface media. For HCI, the interface should be transparent and ideally invisible, its only purpose to make accessible the information and functionality of the application (Norman, 1990). In contrast, theatre, cinema and the interface examples cited here celebrate the interface as an entirely necessary, conspicuous and constructive component of a production, with aesthetic and experiential objectives. Rather than HCI's singular focus on usability the interface examples demonstrate a diversity of motives for their particular adaptions. In Twistori and Murmur Study we see that the interface may intentionally work against ease and usability, and that producers need not be sympathetic to the source, presenting instead a critical interpretation of the script. This notion is entirely radical to conventional HCI and its view of usability as a moral imperative of design (Greenberg & Buxton, 2008). What these examples offer is not ease and utility but unique insight and experience of a familiar source. In many respects the critical and aesthetic focus of the works fit with Dunne's notion of the "post-optimal object":

"In a world where practicality and functionality can be taken for granted, the aesthetics of the post-optimal object could provide new experiences of everyday life, new poetic dimensions". (2005, pg.20)

Conclusion

The cultural context of computer media has changed radically since its origins in the workplace. With social media and Web 2.0, software has become more social in form and function (Manovich, 2007). Platforms such as the web and mobile App Stores have democratised interface design and created a market that rewards creativity and that celebrates authorship - as evidenced in the TapBot reviews. In a bid to better address the aesthetic priorities of contemporary interface and the creative opportunities of technologies such as APIs, we need critical alternatives to interface as a utilitarian conduit (Greenberg & Buxton, 2008). The concept of script is proposed here to address these needs. Using the script analogy

we are able to understand interface design as a creative act of cultural production and appreciate that the interface medium, like its theatrical and cinematic cousins, can support a variety of realisations from the prosaic to the poetic. More than a convenient way to describe the relationship between API and interface, the script analogy brings with it the potential to connect interface design with highly developed fields of creative production and critical evaluation.

Bibliography

Baker, Christopher 2009, *Murmur Study*, http://christopherbaker.net/projects/murmur-study/

Bell, Killian 2011, ''Tweetbot for iPhone Deserves its Release Day Hype", *Cult of Mac*, http://www.cultofmac.com/tweetbot-for-iphone-deserves-its-release-day-hype-review/90259

Blanc, Shaun 2011, 'Tweetbot's Got Personality', http://shawnblanc.net/2011/04/tweetbot-review/

Billington Michael 2011, 'Richard III - review', *The Guardian*, http://www.guardian.co.uk/stage/2011/jun/29/richard-iii-review-kevin-spacey

Dunne, Anthony 2005, *Hertzian Tales*, MIT Press, MA.

Flipboard, http://flipboard.com/

Greenberg, Saul and Buxton, Bill 2008, 'Usability Evaluation Considered Harmful (Some of the Time)', *Proceeding of the twenty-sixth annual SIGCHI conference on Human factors in computing systems*, pp.111-120.

Holt, Michael 1988, *Stage Design and Stage Properties*, Phaidon Press Limited, London.

Hoy, Amy and Fuchs, Thomas 2008, *Twistori*, http://twistori.com/

Lawless, Jill 2011, 'Kevin Spacey Shines as Shakespeare's Richard III', *Associated Press*, http://abcnews.go.com/Entertainment/wireStory?id=13968815

Lofte, Leanna 2011, 'Tweetbot once again redefines Twitter for iPhone', *TiPb*, http://www.tipb.com/2011/04/13/tweetbot-iphone-app-review/

Manovich, Lev 2006, 'Interaction as an Aesthetic Event', *Vodafone Receiver Magazine*, www.receiver.vodafone.com

Norman, Donald 1990, 'Why Interfaces Don't Work', in Laurel, Brenda (ed.), *The Art of Human-Computer Interface Design*, Addison-Wesley Publishing Company, Reading, MA.

Prasad, Naik 2011, 'Tweetbot Review', *Techtree*, http://www.techtree.com/India/Reviews/Tweetbot_Review/551-115093-609.html

Schmidt, Karsten and Pohflepp, Sacha 2009, *Social Collider*, http://socialcollider.net/

Spencer, Charles 2011, 'Richard III, Old Vic, review', *The Telegraph*, http://www.telegraph.co.uk/culture/theatre/theatre-reviews/8606884/Richard-III-Old-Vic-review.html

Tapbots 2011, *Tweetbot*, U.S. http://tapbots.com/software/tweetbot/

Wunderling, Jens 2008, *Tweetleak*, Germany. http://www.defaulttopublic.net/tweetleak/index.html

Yallop, Colin (ed.) 2004, *Macquarie Concise Dictionary*, The Macquarie Library, Macquarie University, Australia.