SHIFT-BOX: INBOX Time Shifting to Reduce Email Clutter

Romain Vuillemot, Jean-Marc Petit, Mohand-Saïd Hacid
Université de Lyon, CNRS
LIRIS, UMR5205
Lyon, France
romain.vuillemot@insa-lyon.fr

ABSTRACT

In this paper, we are interested in exploring the idea of giving users the ability to shift emails reception time, such as to pause or replay flows of emails, during a user-selected time window. To our knowledge, email analysis and interaction techniques mainly use reception time to extract sequential events from emails, such as to reconstruct conversational threads: time stays no more a *physical quantity*. The hypothesis we want to investigate is whether giving users the ability to shift time would help them to better deal with INBOX clutter and reduce email overload. To test this hypothesis, we are currently designing a media player-like widget to let users re-play email receptions and actions (e.g. classification, tagging, archival, etc.) using a time-centric data stream model. We detail our implementation roadmap and next steps from our ongoing project.

1. INTRODUCTION

Time is part of every single action of our daily activities. It is a strong footprint of occurring events, and it is a useful mental index to retrieve accumulated information, even heterogeneous. For example, when remembering about a meeting, one can easily tie together locations, social networks of people, photos and videos, because all having a matching time attribute. Time also brings sequentiality to information and put it in order.

In the context of emails communications, multiple references to time can be made. The most common ones are time referred in emails text body, and reception time in emails header and emails clients. In this work, we focus on the latter one.

In all email clients we reviewed, reception time is visually encoded as a position (e.g. most recent emails are on top of the INBOX emails' list) or as timestamps on a temporal axis, displayed as timelines [6, 11]. Moreover, time is also often visible as a numerical value, and may also be encoded as colors, symbols, or even both to display new or recent emails. In those cases, a differential value of time is used (i.e. to extract sequentiality out of emails), rather than its absolute value.

Those various visual encoding of time aim at reducing users cognitive load, by quickly showing incoming emails to process, or reconstructing conversations threads. But since email is asynchronous, conversations can be interwoven with no visual order. Thus, only relying on the visual display of emails to extract knowledge or organize tasks is limited. Automated systems provide solutions to better navigate in email archives, such as Thread-based mail browser [8] or [7] to detect hierarchies, using emails mutual links (reply, follow up, etc.). Conversations reconstruction are not only a visual aid, and can lead to advanced networks analysis and exploration [2]. Other approaches extract tasks from emails [11] to compute task-centric views of the INBOX [4].

In all the related works we reviewed, time is either visually encoded, or vanishes under inferences or topological constructions. Except from short peripherical animations (such as with notification mechanisms [12]), time does not stay a *physical quantity*.

CEAS 2010 - Seventh annual Collaboration, Electronic messaging, Anti-Abuse and Spam Conference July 13-14, 2010, Redmond, Washington, US

1.1 Preliminary Email Usage Study

In the frame of the DLM 3.0 project[1], we conducted a preliminary email usage study, to investigate email communications usages of our colleagues at LIRIS lab[2]. We interviewed 7 people, 5 men and 2 women, age ranging from 23 to 42. They are all familiar with computers and technology, and email is central in their everyday professional activity. After briefly introducing the aim of the interview, we asked them the same set of questions about their habits, satisfactions and pains with emails. Average interview time was 40 minutes, in a neutral environment (except for one person who had to remain on duty in her office).

While our initial ambition was to identify a set of user profiles to better categorize individuals and provide ad-hoc solutions, we quickly realized that email usages are not universal, and even contradictory from one colleague to another. Similar issues had already been raised earlier [5].

We now detail our early findings, quoting our colleagues own words. One colleague said *"I classify my emails as a preliminary step before even doing any task -even reading emails' body"* while another one said *"email classification is the final step of my process, a classified email won't be touched again, since it does not appear anymore in my INBOX"*. One colleague didn't even classify his INBOX, leaving up to 3000 emails in it and is only using Thunderbird's folder filter to retrieve a specific email.

We particularly focused on classifiers, whose main pain was their INBOX' size. *"I don't like having many emails in my INBOX since it makes me overlooking many times at the same set of visible emails"* [..] *"it gives recent emails more importance"*. When asking about some potential overload,

[1] http://www.dlm30.com/
[2] http://liris.cnrs.fr/

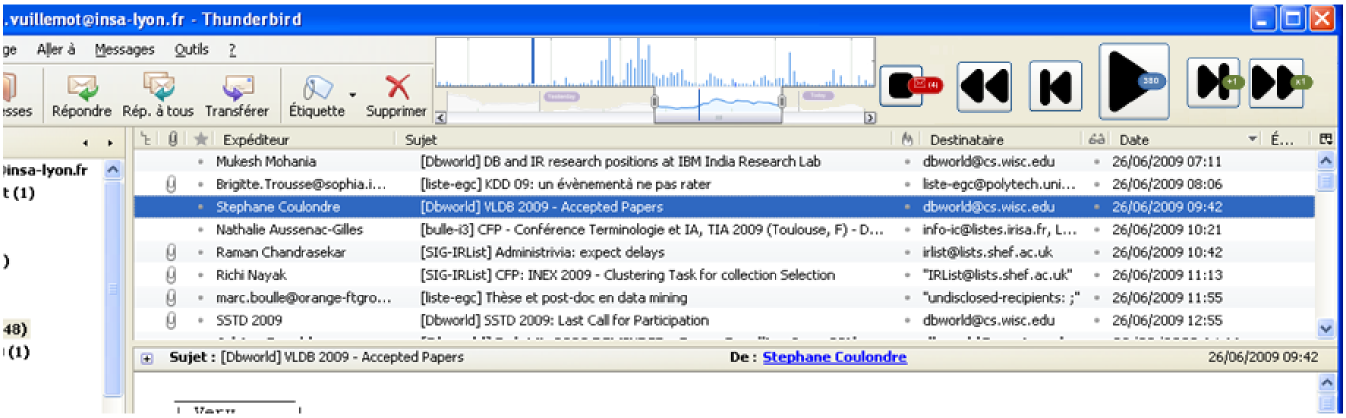

Figure 1: Overview of a classic email interface including our Shift-BOX widget (top right). The widget is similar to a video player controller coupled with a timeline. The widget aims at time shifting email actions (e.g. classification, tagging, etc.), such as to replay emails reception in the current INBOX as it happened in the selected time window.

it was recurrent (5 people of 7) to hear saying *"it's ok as long as there are no email out of sight"* or *"If I can't see all my emails, I feel overloaded and don't know which email to process first"*. One colleague also claims *"it happens very frequently that I don't remember what I did with my INBOX emails. I always have the feeling that I forgot or missed something."*.

All participants consider their INBOX as the central communication hub, and the start of most of their daily activities. A participant even coined the term *"active memory"* to refer to his INBOX, to emphasize its importance.

None of our colleagues were satisfied yet with current email applications. None of them also had the feeling to be in control of their email flow.

1.2 Analysis and Research Hypothesis

From our results, users seem to construct a strong mental model over their INBOX. As the temporal stream of emails occurs, emails sequences and their context (i.e. other consecutive emails) seem to be linked together during the time window within which they are visually adjacent. This effect has been identified has the INBOX clutter issue, that states INBOX complexity makes important tasks overlooked and become a factor of email overload [10].

The hypothesis we want to investigate is *"Would users be more productive and satisfied if time stays a physical quantity that they can shift, to perform such actions as to pause or to replay sequences of email reception in their* INBOX*?"*. We expect users to be more confident in archiving emails and quickly take decisions, since they would be able to go back and forth to various INBOX states.

To tackle this hypothesis, our approach is twofold. In a first part we introduce a data stream model (Section 2) to formally define users' email flow and actions (e.g. classification, tagging, etc.). This model is a preliminary ground for more advanced formal definitions and processes, such as pattern mining and rules detection. In a second part we detail SHIFT-BOX, our media player-like interactive widget to time shift emails flow (e.g. pause, replay, fast-forward, etc.) (Section 3). We also give our implementation roadmap (Section 4) and next steps (Section 5) of our work.

2. EMAIL FLOW AS A DATA STREAM

Following [1], a data stream S is a set of pairs (tuple, timestamp) as $S = \{(s_1, t_1), (s_2, t_2), ..., (s_n, t_n)\}$. An email is a pair (s, t). The tuple attribute s is a list of elements composing an email (title, reception timestamp, recipient, etc.). The timestamp attribute t is a sequence attribute, that can be ordered. We use the physical (i.e. current time) timestamp as this sequence attribute (other email timestamps such as delivery time are encoded in the tuple s).

The data stream being virtually infinite, we focus on a finite sub-set that we call an email *active window*.

2.1 Email Active Window

An email *active window* is defined as a time interval $w_a = [t_{start}, ..., t_{end}]$, $t_{start} \leq t_{end}$, with t_{start} and t_{end} the start and the end of the window. N is the width of the window (i.e. count of time units). The window operates on the data stream S, resulting in $S_{w_a} = \{\forall (s,t) \in S | t \in w_a\}$, $S_{w_a} \subset S$. $|S_{w_a}|$ is the number of emails stored within w_a.

Let t_{now} be the current physical time. $t_{end} = t_{now}$ means S_{w_a} automatically stores incoming messages (e.g. automatic email check). $t_{end} < t_{now}$ means S_{w_a} is not up to date: there might be queued messages in w_q, the *queue window*. We define $w_q = [t_{end}, .., t_{now}]$. w_q also operates on S, S_{w_q} is the subset of queued emails and $|S_{w_q}|$ its count.

Let t_{shift} be the current time *shift*, $t_{start} \leq t_{shift} \leq t_{end}$. t_{shift} can be seen as a virtual physical time shift. Let δ be a positive value that is a multiple of time unit. δ is called the time shift unit increment and represents the pace at which shifted time is played.

2.2 Actions on the Data Stream

We now define two classes of actions on the data stream. Those classes remain generic on purpose, since the functionalities space is wide and can be extended dynamically.

The first class of actions A operates on emails pairs, and it timestamps them to current physical time $A : (s,t) \rightarrow (s', t_{now}), t \leq t_{now}$. For instance, adding flags (e.g. read / unread) or set class or label to tuples, results in new pairs in the stream. Actions in this class can be replayed during a time shift, since they directly deal with email management tasks.

The second class W of actions operates on the active window, and does not generate new pairs. For instance, email checking means extending t_{end} to t_{now}. Active window's boundaries extension / reduction (using handlers) are also actions in this class. Flow control actions (e.g. play, pause, etc.) belong to this class too. The flagship action is play and can be defined as automatically browsing S at a current t_{shift} that is increment by a δ time unit, from t_{start} up to t_{end}. Actions in this class are not replayed during a time shift, since they don't directly deal with email management tasks.

3. SHIFT-BOX

We now detail the widget's visual and interactions design. A preview of the widget is illustrated as integrated in a classic email client real estate (Figure 1). The widget aims at time shifting email actions, such as emails reception in the current INBOX but in a time-delayed manner. To reduce the learning curve, we used a media player-like widget, coupled with timelines which are well-known interactive items.

The widget behavior is very intuitive. By clicking on the "Play" button, emails appear in the currently selected INBOX (below the widget, Figure 1) as they were just received at the time the vertical blue segment is (on the timeline), progressively sliding to the right, over time. Past actions are replayed (e.g. classification, deletion, etc.) such as defined in the A actions class.

3.1 Time Shift Controls

SHIFT-BOX features common video players controls, acting on emails located in the currently selected INBOX. We added colored bubble boxes on controls to show quantitative information, such as buffered emails count. We now detail each button behavior:

Play/Pause: plays/pauses the email stream, currently set at t_{shift}. A blue bubble box 380 shows number of emails $|S_{w_a}|$ left to play in the active window. The pause button appears when the play button has been clicked, and vice-versa.

Stop: resets t_{shift} back to present day and time t_{now}. The red bubble box (4) appears and shows queued emails count $|S_{w_q}|$ when new emails arrived in the queue window. Once a reset is done, the button is not be available until the active window is set again.

Speed: increases/decreases by one more/less unit the current playing speed δ/s. When clicked, a green bubble box x1 quickly appears to show the rate $\times\delta$, and then smoothly vanishes.

Next by Next: jumps to the next email event, by a group of δ items. A green bubble +1 quickly appears to remind the group size δ.

Many other tiny design details -such as emails preview by button hovering- are to be implemented to reduce interactions overload.

3.2 Time Shift Window Interactions

SHIFT-BOX also features a quantitative preview (using a timeline) of emails stored in the active window (and around). The active window is binded to a range slider with two window handlers, and a multi-resolution time strip:

Window Handlers: shows current w_a boundaries t_{start} to t_{end} and current re-play time t_{shift} as a vertical blue segment. A caption yesterday on time intervals shows N, window's width in plain text, and clicking on it sets the active window to it.

Multi-Resolution Time Strips: shows the email distribution over time, both from the window (higher strip on Figure 1), and from the context around the window (lower strip with less details but on a wider time range). Regarding the Y-axis, the higher the peak is, the more emails have been received at that time.

Once the window is set, emails located in the INBOX at that time appear in it. In case they have been processed after the current t_{shift}, little visual cues show coming interactions to be performed on emails. For instance, based on existing Thunderbird graphics to show replies, forwards, both and new emails, we use the same but emptied to show upcoming actions, respectively , , , right by emails. Those graphics go back to color when the action is performed.

4. IMPLEMENTATION ROADMAP

We plan to implement our data stream model in a declarative way, with both live streams (incoming mails) and archived streams (logs on email actions) as inputs [3]. Logs are provided by `IMAP` servers and client activity logging. Regarding the widget, our prototype is bundled as a Thunderbird extension (Figure 1). Privacy and reliably are major concerns, which are anyway the case with any live email experiment.

5. NEXT STEPS

Our primary next step is to keep implementing our widget. Once a satisfying development state is reached, we perform a longitudinal usability test with real users in their everyday working environment. The widget's logs will help us to automatically track users activity at a wide scale, and compare it with previous behaviors without the widget.

Our current data stream model still needs some extensions. Time in emails tuple s may be extended beyond reception time, up to process and submission time, and even referred time in emails. Referred time is extracted using text analysis tools, and would be useful to delete outdated emails (such as expired meeting requests). A model extension would also consist in pattern mining and rules detection.

Playing emails by "batch" of thematic conversational threads over a time period seems promising. Even if SHIFT-BOX is initially conceived to replay INBOX mails, it can also replay any automatically rule-classified email which didn't even meet the INBOX (and landed directly into categories, labels or virtual folders). Other sequential attributes -such as threaded conversations- can also be timely-replayed. Replaying emails editing steps (i.e. drafts) might be interesting to better understand previously taken decisions. Last but not least, one can also set a "cognitive" limit of received emails in INBOX. For instance, let's consider an email client that can only display 10 visible emails at the same time in the INBOX. Once this limit is reached, SHIFT-BOX automatically pauses current email flow, and buffers upcoming emails. Advanced policies can be set to "*only add a new email when there is room*" (e.g. a user just been moved an email from the INBOX to another folder) or even "*do not restrain urgent emails*". This last policy will rely on semantic features provided by our partners in the project. We think SHIFT-BOX can also be applied to newsfeeds (Facebook) and social updates (Twitter).

Finally, extending email tasks to users personal environment [9] aims at making email clients an unified application. As we mentioned while introducing our work, time is a universal key to integrate external data, such as sensors values or coffee machine state. Including those environmental data in our model, coupled with email replay, may enable an interesting reflexive activity for users, to think again on a previously taken decisions. New design issues appear, such as smoothly including and interacting with multiple visualization like bar charts (e.g. temperature histograms), maps (e.g. emails route) or graph (e.g. social network of users contacts at the time the email is sent).

6. ACKNOWLEDGMENTS

This work takes part in the ongoing research project DLM 3.0[3] devoted to semantic email, and has been partially funded by the French Ministry of Economics and Labor. We would like to thank our colleagues at LIRIS, for their time and thoughtful comments on their email usage: Simon Meyffret, Pierre-Édouard Portier, Vasile-Marian Scuturici, Mabrouka Gheraissa, Adrien Joly and Amélie Cordier.

7. REFERENCES

[1] B. Babcock, S. Babu, M. Datar, R. Motwani, and J. Widom. Models and issues in data stream systems. In *PODS '02: Proceedings of the twenty-first ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems*, pages 1–16, New York, NY, USA, 2002. ACM.

[2] J. Diesner, T. L. Frantz, and K. M. Carley. Communication networks from the enron email corpus "it's always about the people. enron is no different". *Comput. Math. Organ. Theory*, 11(3):201–228, 2005.

[3] N. Dindar, B. G
"uç, P. Lau, A. Ozal, M. Soner, and N. Tatbul. DejaVu: declarative pattern matching over live and archived streams of events. In *Proceedings of the 35th SIGMOD international conference on Management of data*, pages 1023–1026. ACM, 2009.

[4] A. Faulring, B. Myers, K. Mohnkern, B. Schmerl, A. Steinfeld, J. Zimmerman, A. Smailagic, J. Hansen, and D. Siewiorek. Agent-assisted task management that reduces email overload. In *IUI '10: Proceeding of the 14th international conference on Intelligent user interfaces*, pages 61–70, New York, NY, USA, 2010. ACM.

[5] W. E. Mackay. More than just a communication system: diversity in the use of electronic mail. In *CSCW '88: Proceedings of the 1988 ACM conference on Computer-supported cooperative work*, pages 344–353, New York, NY, USA, 1988. ACM.

[6] S. Rohall et al. Information Visualization, Electronic Mail, Threads, Trees. In *IEEE Symposium on Information Visualization (InfoVis)*, pages 22–23, 2001.

[7] R. Rowe, G. Creamer, S. Hershkop, and S. J. Stolfo. Automated social hierarchy detection through email network analysis. In *WebKDD/SNA-KDD '07: Proceedings of the 9th WebKDD and 1st SNA-KDD 2007 workshop on Web mining and social network analysis*, pages 109–117, New York, NY, USA, 2007. ACM.

[8] G. Venolia, L. Dabbish, J. Cadiz, and A. Gupta. Supporting email workflow. *Microsoft Research*, pages 2001–88, 2001.

[9] S. Whittaker, V. Bellotti, and J. Gwizdka. Email in personal information management. *Communications of the ACM*, 49(1):73, 2006.

[10] S. Whittaker and C. Sidner. Email overload: exploring personal information management of email. In *CHI '96: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 276–283, New York, NY, USA, 1996. ACM.

[11] K. Yiu, R. Baecker, N. Silver, and B. Long. A time-based interface for electronic mail and task management. *ADVANCES IN HUMAN FACTORS ERGONOMICS*, 21:19–22, 1997.

[12] L. Zhang, N. Tu, and D. Vronay. Info-lotus: a peripheral visualization for email notification. In *CHI '05: CHI '05 extended abstracts on Human factors in computing systems*, pages 1901–1904, New York, NY, USA, 2005. ACM.

[3] http://www.dlm30.com/