Left-over Windows Cause Window Clutter... But What Causes Left-over Windows?

Julie Wagner, Wendy E. Mackay, Stéphane Huot

Left-over Windows Cause Window Clutter...
But What Causes Left-over Windows?

Julie Wagner[1,2,3]
wagner@lri.fr

[1] INRIA
F-91405 Orsay, France

Wendy E. Mackay[1,2,3]
mackay@lri.fr

[2] Univ Paris-Sud (LRI)
F-91405 Orsay, France

Stéphane Huot[2,1,3]
huot@lri.fr

[3]CNRS (LRI)
F-91405 Orsay, France

ABSTRACT

Sleep mode lets users go for days or weeks without rebooting, supporting work on multiple tasks that they can return to later. However, users also struggle with window clutter, facing an increasing number of *'left-over windows'* that get in the way. Our goal is to understand how users create and cope with left-over windows. We conducted a two-week field study with ten notebook users. We found that they work in very short sessions, switching often between computer-based and external tasks. 34% of left-over windows remain untouched for a day or more, increasing in quantity until they all disappear after a reboot. Some users reboot as a deliberate 'clean-up' strategy, whereas others lose left-over windows after an unexpected system crash. Users intentionally keep left-over windows as to-do lists, as reminders of upcoming tasks, and for facilitating future access; the rest are simply forgotten. Tools for visualizing and managing left-over windows should help users reduce window clutter, while maintaing the benefits of interruptible work sessions.

Author Keywords

Left-over windows; Window clutter; Window logger.

ACM Classification Keywords

H.5.2 Information Interfaces and Presentation: User Interfaces—*Windowing systems*.

General Terms

Experimentation; Human Factors.

INTRODUCTION

Window managers have changed little over the past decade, despite major changes in their context of use. Laptops are now powerful enough for both work and home settings, enabling users to shift between office and personal activities in the course of a day. Travelling now involves taking along one's computer, increasing the scope and variety of tasks performed. Unlike early desktop computers, which forced users to close all open windows at the end of the day, modern laptops come equipped with a *sleep* mode that preserves
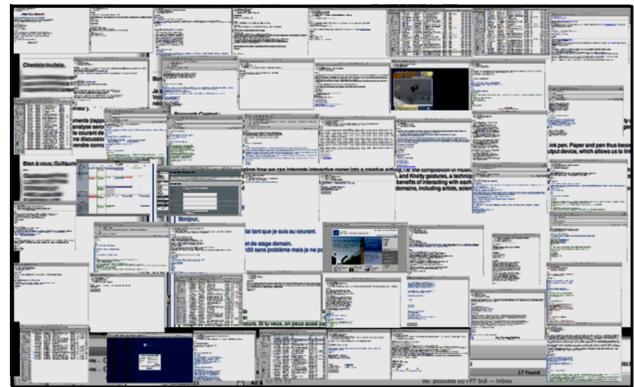
Figure 1. Example of a progressive clutter process over weeks: screenshot in Exposé mode of over 50 windows on a user's virtual desktop.

the state and layout of their windows when the computer shuts down. Users can shift among multiple tasks, preserving context in the face of interruptions or when restarting the computer. Many users thus avoid rebooting for weeks or even months at a time.

There is, however, a drawback: leaving windows open can result in overwhelming 'window clutter'. Figure 1 shows an *Exposé* view of one user's screen, with over 50 open windows that have accumulated since the last reboot. These include left-over windows, i.e. duplicate or no-longer-needed windows that get in the way of finding windows needed for a particular task, reducing productivity and contributing to the user's sense of information overload.

This paper investigates the lifecycle of windows between reboots, with particular emphasis on 'left-over windows'. We conducted a field study that logged ten users' window activity for two weeks. In addition to semi-structured *in situ* interviews, we also introduced 'critical incident' questionnaires that asked about the status of a randomly chosen selection of left-over windows. We discuss the results and conclude with implications for design.

RELATED WORK

The first system to address window overload explicitly was Xerox PARC's *Rooms* [4], which let users organize groups of tasks into multiple virtual project rooms. Later window managers, e.g., Apple's *Spaces*, offer similar task-context organizational strategies, but do not explicitly address the problem of extraneous windows.

Although we know of no studies that investigate the specific causes of left-over windows, Barreau and Nardi's [1] early study of desktop management offers some clues. They identified three types of files, *ephemeral*, *working* and *archived*, noting that *'quick-and-dirty reports and memos* [end up in a] *no-man's land between ephemeral and working information'*. Users must track but do not actively work on these files until after completing a follow-up activity.

Hutchings et al. [5] emphasized the fact that information workers are constantly interrupted, with an average of 50 tasks shifts over the course of a week. Resuming tasks requires effort, depending upon task complexity, duration, and length of absence, and they noted that some users leave windows open to remind them of previous tasks when they return after an interruption.

Salvucci [6] argues that reconstructing the task context after an interruption is critical for successful resumption and depends upon sometimes subtle cues in the external environment. For laptop users, moving from one physical location to another not only affects the tasks performed, but may also disrupt the reconstruction process, potentially increasing the number of left-over windows.

This paper investigates the characteristic patterns and underlying causes of left-over windows. We hope this will lead to better window management tools that help users cope effectively with window clutter.

FIELD STUDY
We conducted a field study to help us understand the causes of window clutter, or why some users leave large numbers of windows open between work sessions. We asked the following research questions:

Q1: How do interruptions affect work-session duration?

Q2: What is the relationship between left-over windows and the number of work sessions between reboots?

Q3: Why don't users close left-over windows?

We define abandoned or *left-over windows* as those that remain open from one work session to another, without being modified by the user in any way. Some windows are all or partially visible, others may be minimized or hidden behind other windows. We define a *work session* as a continuous interaction between the user and the computer, bracketed by periods of inactivity. Rebooting or sleeping the computer always ends a work session, as does lack of input from the user for one hour or more.

Participants
We recruited ten laptop users (three women) aged 31 to 58: 1 CEO, 1 managing director, 3 professors, 2 associate professors, 1 researcher and 2 post-doctoral fellows. They engage in teaching (60%), engineering (60%), administration (70%) and research (80%).

WM-LISA Logger
We developed *WM-LISA* – **W**indow **M**anagement **L**ogger **I**nvestigating **S**ession behavior and **A**bandoned windows – to log the opening and closing of windows, applications and system events. Similar to the *wmtrace* [2] logger for XWindows, WM-LISA runs as an unobtrusive background process on the Mac OS X operating system (10.5 and higher) that intercepts low-level window manager events and Apple Accessibility API information.

WM-LISA records IDs and times of all standard window events and uses Applescript to log information about tabs within *Safari* web browser windows, e.g. the number of tabs and their corresponding IDs. It uses low-level accessibility and device events to generate high-level window and tab manipulation events, e.g., *window content changed* and *window viewport changed (scrolling)*. WM-LISA also logs when the computer goes to sleep, awakens, is rebooted or shut down. Temporary windows, e.g., dialog boxes, tooltips, menus, system menu bars and icons, are not logged.

Data collected
We recorded responses to semi-structured interviews and critical-incident style questionnaires. We also collected logs of every window, tab, sleep, wake-up and reboot event monitored by WM-LISA.

Procedure
We asked participants to avoid deleting windows prior to our initial interview, except any they considered confidential. We conducted each interview in the participants' offices and asked about their window management strategies. We installed WM-LISA and asked them to use the Safari web browser for the 14 days of the study. WM-LISA logged events silently for the first 10 days. For the final four days, whenever the participant started a new session, a questionnaire popped up based on five randomly chosen screenshots from their current set of left-over windows. Participants were encouraged to avoid speculation and offer specific, in-context reasons as to why they abandoned certain windows.

RESULTS
Seven users answered at least one critical incident questionnaire. We used ANOVA and Nominal Logistic analysis[1] to analyze 164,897 logged events.

Session duration
We identified 512 discrete sessions across all participants over the two-week study period, 50% of which were shorter than 65 minutes. Participants averaged 1-9 sessions per day (*median* = 4), divided into three periods: *night* (0am-6am), *day* (7am-5pm), and *evening* (6pm-11pm).

Most sessions occur during the day ($F_{2,12} = 30.73, p < 0.0001$, mean = 27.1 day sessions and mean = 88.9 length). These sessions tend to be shorter than evening or night sessions, which do not differ significantly from each other. Several

---

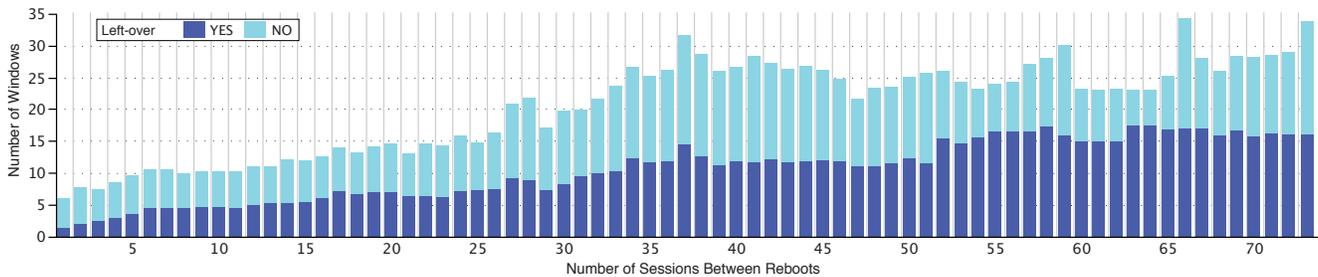[1]We used the *JMP 9* statistical package.

Figure 2. Numbers of active (pale) and left-over (dark) windows both increase with greater inter-reboot intervals (all participants).

participants commented that they are interrupted more frequently during office hours than in the evening or at night:

> *"Very often when I leave work, I say I have not worked at all today, because for me working is doing research ... and that only happens when I close my [office] door." (P2)*

Reboot strategies
We asked participants to describe their rebooting strategies. Six avoid rebooting and only do so for system updates, performance issues or system crashes:

> *"I never choose to reboot. ... I have to reboot because of updates. I do not choose to." (P10)*

We classified these six participants as *seldom rebooters (SR)*. They often become overwhelmed by their left-over windows, dealing with them by default when they are forced to reboot.

The remaining four are *normal rebooters (NR)*. They control the number of left-over windows by setting aside a regular time to 'clean-up' their windows, at the end of the day or in conjunction with a scheduled reboot. Not surprisingly, SR participants rebooted less frequently than NR participants ($p < 0.006$, $r^2 = 0.57$): One participant (P1), a seldom rebooter, never rebooted during the two weeks of the study.

Figure 2 shows that, as the number of sessions between reboots increases, so does the overall number of open windows ($p < 0.0001$, $r^2 = 0.03$) and tabs ($p < 0.0001$, $r^2 = 0.34$). Seldom rebooters keep significantly more windows open simultaneously than normal rebooters ($p < 0.0001$, $r^2 = 0.27$): *median = 17 vs. median = 5 windows*.

> *"It's a cycle: things get progressively more cluttered and one day I say, 'OK', I cannot do this anymore and so I clean up." (P1)*

As the interval between reboots increases, the number of left-over windows tracks the overall increase in the number of windows ($p < 0.01$, $r^2 = 0.5$). However, the number of left-over tabs does *not* show a corresponding increase ($p < 0.3$, $r^2 = 0.22$), implying that tabs are more likely to be reused than left over.

Left-over window patterns

How long do windows stay open?
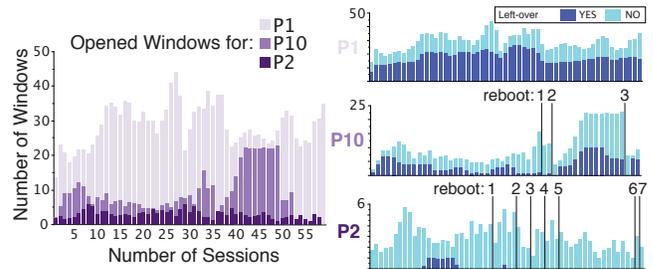We define the window session lifetime as *the number of sessions from when the window or tab was first opened and*

Figure 3. Open vs. left-over windows:
P1 (SR) never rebooted, has the most open and left-over windows;
P10 (SR) rebooted 3 times, half as many open and left-over windows;
P2 (NR) rebooted 7 times, few open and almost no left-over windows.

when it was closed. This varies across participants ($F_{9,9} = 5.1$, $p < 0.0001$) who cluster into three groups (based on a Tukey post-hoc test): (*A*) P1 keeps windows the longest (*mean = 22.2 sessions*); (*B*) P8 (*mean = 14.1 sessions*) and P9 (*mean = 13.1 sessions*) keep windows for an intermediate number of sessions; and (*C*) the remaining participants keep them for relatively few sessions. Seldom rebooters keep windows longer than normal rebooters ($F_{1,1} = 97.3$, $p < 0.0001$, *mean = 8.5 sessions vs. mean = 2.7 sessions*).

P1, who never rebooted, had the longest window session lifetime, 58 sessions for both windows and tabs. Even group *C*, with the shortest window session lifetime, left some windows open for up to 29 sessions and left some tabs open for up to 21 sessions.

How many windows are left over within a session?
The most striking finding was that, within a given session, all participants avoided interacting with approximately one third of their windows (*mean = 36.6%*), regardless of the overall number of open windows or reboot practices. However, if we examine participants' total numbers of left-over windows, we see three distinct groups: (*A*) P1, P3, P8, P4 keep a mean of 57.1% left-over windows, (*B*) P10, P9 keep a mean of 30.8% left-over windows and (*C*) P2 keep a mean of 4.7% left-over windows ($F_{9,9} = 144.6$, $p < 0.0001$). The remaining three participants fit between groups (*B*) and (*C*). Figure 2 shows the aggregate number of active (pale blue) and left-over (dark blue) windows per session: the numbers of each both increase as the number of sessions between reboots (inter-reboot interval) increases. Figure 3 compares active vs. left-over windows for three representative participants from each group: P1 (*A*), P10 (*B*), and P2 (*C*).

What causes long-term left-over windows?
Gonzalez et al. [3] observed that information workers organize their daily activities into multiple high-level units of work, referred to as *working spheres*. They found that people switched to a new working sphere about one third of the time, without returning to it that day. This suggests that interruptions and task switching increase the number of work sessions in a day, while decreasing their duration.

We decided to examine the differences between short-term and long-term left-over windows. We categorize a window as 'short-term left-over' if it is opened in a session and, after a number of short, interrupt-driven sessions, dealt with later the same day. In opposition to such windows, which users return to relatively quickly, windows that are abandoned for days at a time are 'longer-term left-over'.

We found that the participants who seldom reboot (SR) leave a higher percentage of windows open and untouched over multiple days (34% vs. 22% for normal rebooters; $F_{1,1} = 5.3$, $p = 0.049$). Although everyone had left-over windows, only seven had left-over tabs. Of these, only 16% were left over for more than a day, with no significant differences between SR and NR participants.

Why do users create left-over windows?
Although a number of left-over windows are simply forgotten, others persist for good reasons. Table 1 summarizes why participants create left-over windows, drawn from both the interviews and the critical-incident questionnaires.

DISCUSSION AND CONCLUSION
This paper describes a field study of how ten laptop users deal with the window clutter caused by left-over windows. We found that interruptions during the day not only result in shorter work sessions but also increase the total number of left-over windows, with a corresponding increase in window clutter (Q1). Users who seldom reboot are more affected than those who reboot often.

We also found that all participants accumulated left-over windows over time, with a corresponding increase in window clutter until the next reboot (Q2). Regardless of the overall number of open windows or reboot rate, approximately one third of each participant's windows remained untouched during the current work session.

Participants offered a number of positive reasons for keeping left-over windows (Q3), including reminders, to do lists and facilitating future access to specific windows. However, they also admitted that many were forgotten and caused undesireable window clutter.

Although our study was limited to a small group of Mac OS X users, it suggests that beneficial features such as using sleep mode to delay closing windows may have unintended side effects, in this case progressively increasing the user's window clutter. We argue that we need to consider this when designing window management tools, allowing users to quickly identify left-over windows and distinguish

Reason	Sample participant comment
Intermittent Access	*This is documentation I program with ... I keep the window open because I still have [to work on it].* (P8)
Permanent Access	*This is my email client which stays open when I'm in front of my computer. Sometimes I minimize it.* (P6)
Task Reminders	*[This] email window has probably been around since February* (in a July interview.) *I need to collect data and it isn't high enough on my priority list."* (P1)
Longterm Reminders	*I keep this window open as a bookmark but also to remind me to finish it. This is important as a mid- or long-term [task], but not for now."* (P8)
To do Lists	*I minimize [windows] ordered by priority. It is also a todo list. I look at the filename to see what I have to do and I order them accordingly.* (P6)
Awaiting Action	*This message arrived today that I want to remember to tell [name] about. I'll close it when I've talked to her about it (probably tomorrow).* (P1)
Completed Tasks	*This is a Finder window with the folder of an old presentation. I usually keep several Finder windows open, some are just forgotten.* (P4)
Forgotten Clean up	*I searched this PDF document for a presentation that I am preparing. I closed the window of the active presentation but forgot this window.* (P8)

Table 1. Reasons for keeping left-over windows.

between those that serve a useful purpose and those that were simply forgotten and are now getting in the way.

ACKNOWLEDGMENTS

REFERENCES
1. Barreau, D., and Nardi, B. A. Finding and reminding: file organization from the desktop. *SIGCHI Bull. 27* (July 1995), 39–43.

2. Chapuis, O. Gestion des fenêtres: enregistrement et visualisation de l'interaction. In *Proc. IHM '05*, ACM (2005), 255–258.

3. González, V. M., and Mark, G. "Constant, constant, multi-tasking craziness": managing multiple working spheres. In *Proc. CHI '04*, ACM (2004), 113–120.

4. Henderson, Jr., D. A., and Card, S. K. Rooms: the use of multiple virtual workspaces to reduce space contention in a window-based graphical user interface. *ACM Trans. Graph. 5*, 3 (July 1986), 211–243.

5. Hutchings, D. R., and Stasko, J. Revisiting display space management: understanding current practice to inform next-generation design. In *Proc. GI '04* (2004), 127–134.

6. Salvucci, D. D. On reconstruction of task context after interruption. In *Proc. CHI '10*, ACM (2010), 89–92.