J Digit Imaging (2014) 27:474–478
DOI 10.1007/s10278-014-9690-3

Autopage and the Use of Computer Scripts to Automate Microtasks

Jason D. Balkman · Alan H. Siegel

Published online: 16 April 2014

Abstract Paging referring clinicians with imaging results is a frequently repeated "microtask" performed by practicing radiologists. Many institutions use online alpha-paging systems to provide this integral part of safe and efficient patient care. Although sending an alpha-page can often be accomplished within one minute, current tools may disrupt workflow by distracting users with a series of tedious mouse clicks. We describe the development, evaluation, and updates of a portable tool that sends alpha-pages to referring clinicians using two keystrokes. This software integrates study information obtained from the Picture Archiving and Communication System (PACS) with an existing hospital paging system.

Keywords Reading Room · Workflow reengineering · PACS integration

Introduction

Paging referring clinicians with imaging results is a routine, yet critical task performed by radiologists and represents a National Patient Safety goal by the Joint Commission [1]. A microtask such as this takes less than 1 min to complete, but often requires a radiologist's coordinated attention to multiple mouse clicks as well as population of online web forms containing callback numbers and patient information. Such activities generate a repeated distraction that may contribute to fatigue, decreased attention, and overall decreased efficiency. Paging a referring clinician with a critical result may be delayed or forgotten altogether because of these deterrents.

Many institutions use web-based online forms to send text, or "alpha-pages", to referring clinicians about imaging results. Alpha-paging may involve opening a web browser, querying the referring physician's pager identification, followed by populating an online form that contains a callback number, message, referring physician pager identification, and the name of the sending radiologist.

An automated paging script could combine these multiple steps into one or two keystrokes. This would significantly decrease the amount of time and number of distractions incurred by sending pages for both routine and critical communications. A similarly functioning program might require integration into Picture Archiving and Communication System (PACS) software and could be expensive to implement. A free and adaptable computing script was developed instead to increase the efficiency of paging referring clinicians. Similar scripts may be implemented at most institutions that currently use standalone online alpha-paging systems.

Materials and Methods

Autopage was implemented on Windows (XP and 2000) (Microsoft, Redmond, WA) operating systems using the free online scripting programs "AutoHotkey" and "Python 2.7" in conjunction with PACS software, Imagecast (General Electric, Fairfield, CT). AutoHotkey is a free online scripting language used to program routine computer desktop tasks such as repeated mouse clicks into single keystrokes [2]. Python is a free, portable, and extensible scripting language that can automate online searching and handle the Asynchronous JavaScript and XML (AJAX) requests and responses commonly encountered during web navigation [3]. Notepad version 5.1 (Microsoft, Redmond, WA) and Emacs version 23.3.1 were used to compose the AutoHotkey and Autopage

J. D. Balkman (✉) · A. H. Siegel
Department of Radiology, Dartmouth-Hitchcock Medical Center,
One Medical Center Drive, Lebanon, NH 03766, USA
e-mail: jason.d.balkman@hitchcock.org

scripts, respectively. Emacs is a versatile script editor widely used in the computer programming community [4].

Autopage is customized to execute when a diagnostic study is being viewed in PACS, and the "Control-Alt" and "p" keys are pressed simultaneously. An AutoHotkey script registers the keystrokes and copies the study information from the PACS screen panel to the Windows clipboard (Fig. 1). The study information panel contains the type of study as well as the names of the patient and referring clinician. The AutoHotkey script then calls the executable Autopage Python code.

Once executed, Autopage code copies, parses, and stores the name of the patient and referring clinician from the Windows clipboard. A series of AJAX requests and responses are performed in the background using the hospital intranet directory. The online directory responds with a hypertext markup language (HTML) form containing the pager identication as well as a link to the online paging system. The pager identification is stored and another automated request is made for the online hospital paging system using the provided link.

Autopage conducts an authentication handshake with the online paging system. The user name is obtained from the Windows session and password is obtained from an encrypted file provided at the beginning of each session. After authentication, Autopage populates the online paging form using previously stored information including the referring physician name, pager identification, message (patient name), current user (radiologist sending the page), and callback number (provided by the user).

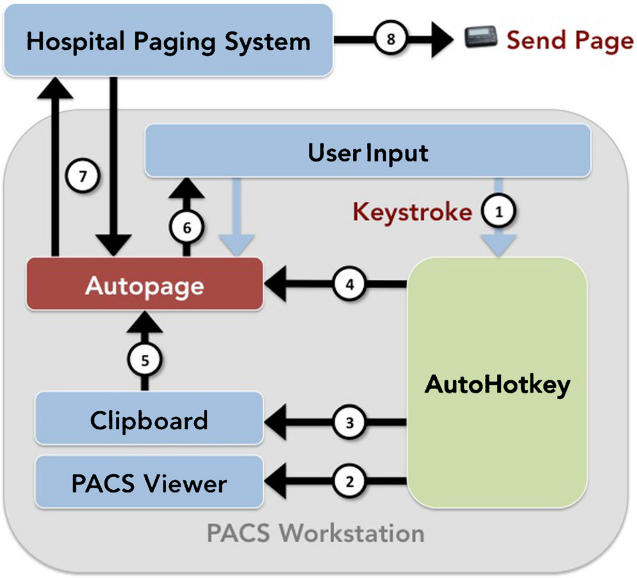


Fig. 1 Paging in two keystrokes (*blue arrows*). AutoHotkey script launches with the first keystroke [1] causing study information to be copied from the radiology PACS [2] to the Windows clipboard [3]. Autopage Python code is executed from the AutoHotkey script [4] and retrieves the study information from the clipboard to construct an alpha-paging form [5]. The form is presented to the user for verification and can be confirmed with a second keystroke [6]. This initiates the sending of the alpha-page using a series of seamless handshakes between Autopage code and the hospital paging system [7]

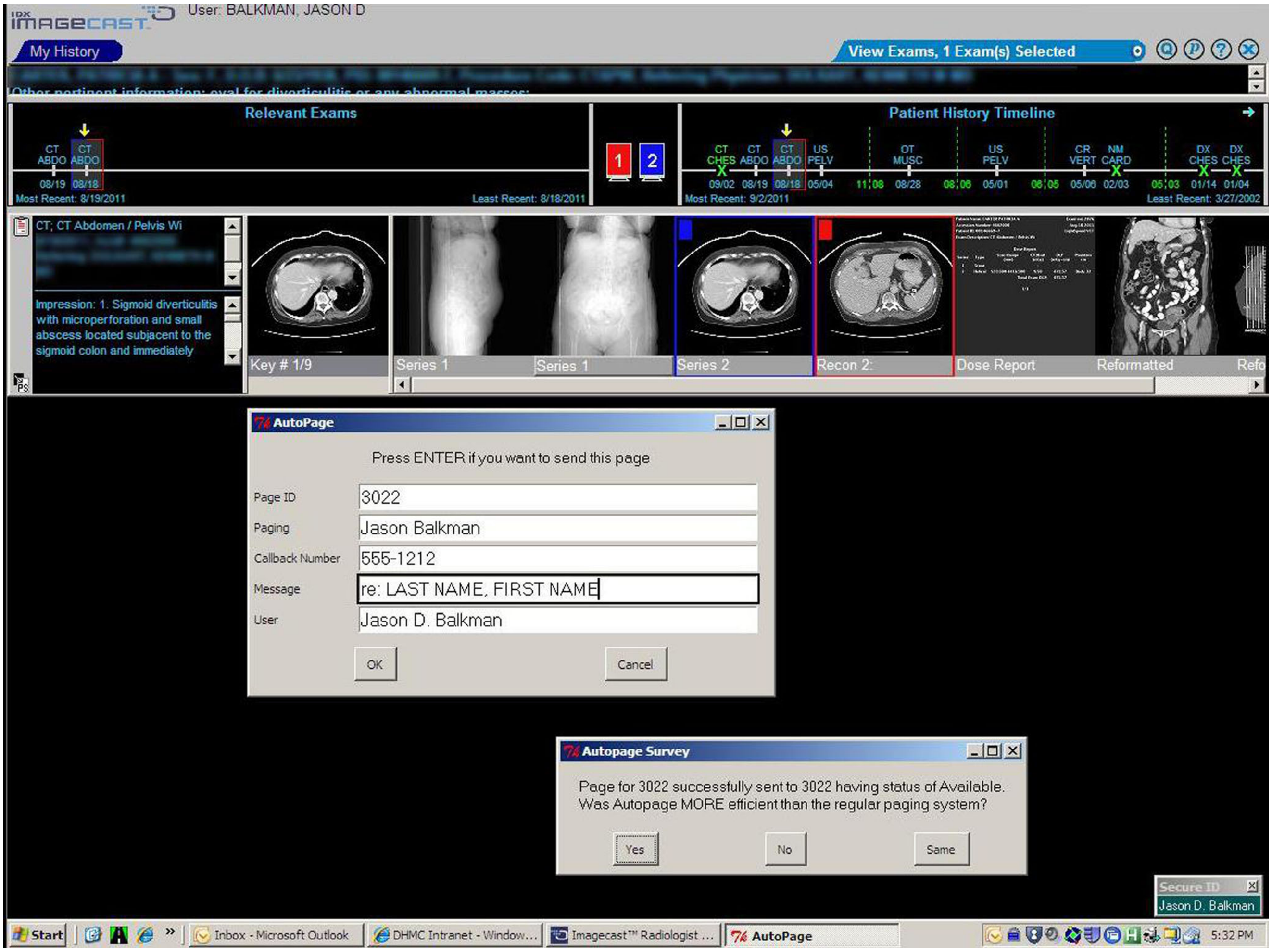
Before submitting the online paging form, Autopage prompts the user to confirm the callback number, referring physician and pager identification, message, and user name (Fig. 2). If no changes are made, the user may submit the page using a second keystroke ("Enter"). Afterwards, a confirmation message is obtained from the hospital paging system and displayed to the user.

Each use of Autopage is logged on a local HIPPA compliant file system as a simple text file for tracking purposes. Log information within this text file includes the sender and recipient of the page, subject of the alpha-page, and timestamps recording the date and time when the Autopage script was launched and when it finished. During the testing phase of this application, a single question survey was launched after each page asking whether the user believed Autopage was more, less, or similarly efficient to the standard paging method. Users of the software preferred a single question survey over multiple questions in order to minimize disruption to workflow. Survey results were recorded in the log file.

Results

AutoHotkey and Autopage were installed on 17 radiology workstations running Windows XP or 2000. Radiology staff and residents received an informational email regarding the program and its usage. A total of 16 residents and attending physicians used the program in a monitored setting a total of 107 times prior to this writing. A survey after each page asked whether the user believed Autopage was more efficient than the previous online paging system. Users percieved Autopage to be more efficient 98 % (105/107) of the time. To determine the actual time saved using Autopage, a total of five radiology staff and residents were observed sending pages to referring clinicians using the old paging system. Users spent on average 45 s of distracted time sending alpha-pages using the traditional online hospital paging system. The same users were able to send pages on average in 6 s with minimal distraction using Autopage. The same five users directly surveyed reported that Autopage was significantly easier to use, more satisying, and required less attention in comparison to the old paging system.

Autopage is maintained by a radiology resident, requiring approximately one half-day of work effort every 3–4 months to maintain compatibility with department workstation updates. A number of changes in the workstation environment have resulted in necessary updates to the Autopage program. These include changes to the Imagecast user interface requiring modification of the Autopage parser module which extracts paging information from the Windows clipboard.



User: BALKMAN, JASON D

[redacted]

Other pertinent information: oval for diverticulitis or any abnormal masses:

CT; CT Abdomen / Pelvis Wi

[redacted]

Impression: 1. Sigmoid diverticulitis with microperforation and small abscess located subjacent to the sigmoid colon and immediately

AutoPage

Press ENTER if you want to send this page

Page ID	3022
Paging	Jason Balkman
Callback Number	555-1212
Message	re: LAST NAME, FIRST NAME
User	Jason D. Balkman

OK Cancel

Autopage Survey

Page for 3022 successfully sent to 3022 having status of Available.
Was Autopage MORE efficient than the regular paging system?

Yes No Same

Fig. 2 Screen shot of PACS display and Autopage confirmation. After the initial keystroke, study information is collected from the *top panel* (blurred text) and a confirmation page form is created. Text within the form was modified for this figure. A second keystroke ("Enter") sends the page using the hospital paging system and a confirmation message and one question survey are displayed (see "Autopage Survey" message box)

Additionally, while the original Autopage version used absolute screen coordinates to reference and copy study information from Imagecast, recent migration to a browser-based PACS led to variations in screen displays across users, requiring a more flexible approach. For example, users were found to have a variable number of Internet Explorer browser panels (e.g., "Favorites" or "Tools") visible when Imagecast is running. This moved the Imagecast study panel lower on the screen, making any hard coded screen references not portable across work stations. To solve this problem, a common logo within the Imagecast panel was located using AutoHotkey's "ImageSearch" function, and the study panel was referenced with respect to the logo coordinates.

Another critical update to the software was required because of changes to the hospital intranet directory. During the initial implementation of Autopage, the intranet phonebook used a simple PHP web form that could be automatically accessed by the software. When the directory changed to a web page using AJAX, the program needed to be modified to communicate with the directory using AJAX requests as opposed to PHP web forms. This required approximately 1 week's time of debugging and code modification. It is anticipated that future updates will be needed to comply with upgrades to the workstation operating system or PACS software.

At our institution, the referring physician on the imaging request is most often the physician to be notified of critical results. During the testing phase, it was acknowledged that the program should give the option of paging an alternative physician if needed. Consequently, Autopage was updated so that typing part of a name into the "Paging" field would return a list of possible matches obtained from the online paging directory. In addition, replacing the prefilled pager number in the "Paging ID" field would result in direct paging of that number. If no current PACS study was open when calling Autopage, a blank template would appear and the user could populate the fields as desired. Outside of these features, it was not feasible for the program to predict the current covering physician if it differed from the referring physician on the imaging order form. In the event that a referring physician has no pager number, alternative contact information obtained from the online hospital phonebook would be passed to the user. The current software design could be easily extended to dial cell phone numbers or landlines using an internet-based calling system.

Discussion

Increasing volumes and complex work environments in radiology are driving the need for improved workflow efficiency and communications [5]. Through the implementation of computer scripts, repetitive microtasks can be automated to significantly improve quality and overall efficiency, though this area has not fully been explored [6]. At many institutions, the simple task of alpha-paging a referring clinicians may require manual navigation of several different program interfaces including a PACS viewer to find the referring physician, a hospital intranet portal, an online directory, and a paging system accessed via a browser form. While this task may take less than 1 min to complete, it is broken into several smaller steps that each require coordinated attention to numerous mouse clicks and online form submission. When repeated numerous times, paging may become a source of fatigue, distraction, and inefficiency.

Automation of such microtasks is becoming popularized by the widespread availability of free online scripting software. Macro scripting can unify several programs without upgrading expensive commercial programs. Several examples have been implemented in radiology departments to improve workflow and reduce errors caused by manual repetition [6], but there is currently no free and adaptable script to send alpha-pages to referring clinicians from within PACS. A major academic institution implemented a sophisticated alert tracking and critical communication system into viewing workstations, but it requires navigation of an additional interface which may be distracting to workflow [7]. The complexity of such a system may also make integration into an existing paging system difficult. Furthermore, an agreed-upon standard of critical results communication is needed and implementation of such software may be overly complex for routine communication [8].

A small, efficient, and low-cost paging mechanism is needed for both routine and critical communications in radiology. Our institution required a new paging system which would combine these features in a minimally distracting tool that could be easily implemented and maintained. Previously, our method of paging referring clinicians involved a pager identification query using an online directory, accessing the hospital paging system, populating multiple fields in an alpha-paging form, followed by the submission of an online form. Coordinated attention to this seemingly trivial task was becoming increasingly difficult and distracting in a complex and rushed work environment, and was a common source of disruption to workflow.

We recently implemented a free, adaptable, and easy-to-use computer script, Autopage, that reduces paging to two key strokes. The tool can be used for communicating both routine and critical results. Autopage integrates PACS software with the hospital paging system using two free scripting programs without the need for additional proprietary software. The script is adaptable across multiple computer platforms and can be modified for other uses. No previous experience with AutoHotkey, Python, or HTML coding was needed to implement Autopage, as free documentation and numerous examples were found online to guide its creation. In addition, radiology attendings and residents overall reported greater satisfaction, efficiency, and ease of use when comparing Autopage to the old paging system.

Despite the successful implementation of the Autopage program, there are several limitations to such technology. High level scripting languages such as AutoHotkey have many dependencies. Slight differences in the PACS user interface across workstations may affect the ability of the Autopage program to find and copy information from the screen. Over the past year, several upgrades to our workstation software and the hospital paging system required code modifications. Without support staff familiar with such a program, small changes in the workstation environment may render the program ineffective. A more stable but technically difficult and expensive solution might include gathering data directly from DICOM files rather than relying on screen display information. However, considering information technology (IT) resource limitations, a high level scripting program continued to be the more desirable solution despite the need for occasional code modifications.

Conclusion

Autopage is an adaptable computing script that simplifies hospital paging to two keystrokes for radiologists, thus minimizing workflow disruption. Similar scripts may be implemented in academic, community, and private practice hospital settings without the cost of additional software or hardware. Autopage demonstrates how computer scripts may be used to automate repeated microtasks, especially those that require coordination between multiple programs. Limitations of this technology include the need for periodic maintenance and updates to keep the software operable with its PACS and intranet dependencies.

References

1. Anthony SG, Prevedello LM, Damiano MM, Gandhi TK, Doubilet PM, Seltzer SE, Khorasani R: Impact of a 4-year quality improvement initiative to improve communication of critical imaging test results. Radiology 259:802–807, 2011
2. Free Software Foundation. AutoHotkey. Available at http://www.autohotkey.com/. Accessed November 2, 2013
3. Python Software Foundation. Python. Available at http://www.python.org/. Accessed November 2, 2013
4. Free Software Foundation. GNU Emacs. Available at http://www.gnu.org/software/emacs/. Accessed November 2, 2013
5. Halsted MJ, Froehle CM: Design, implementation, and assessment of a Radiology workflow management system. AJR 191:321–327, 2008
6. Iv M, Patel MR, Santos A, Kang YS: Use of a macro scripting editor to facilitate transfer of dual-energy x-ray absorptiometry reports into an existing departmental voice recognition dictation system. Radio Graph 31:1181–1189, 2011
7. Brigham and Women's Hospital. Alert and Notification of Critical Radiology Results (ANCR). Available at http://www.brighamandwomens.org/Research/labs/cebi/CCTR/ANCR.aspx. Accessed November 2, 2013
8. Proval C: Fail-safe: Automating critical-results notification. Available at http://www.imagingbiz.com/articles/view/fail-safe-automating-critical-results-notification. Accessed November 2, 2013