Citrine: Providing Intelligent Copy-and-Paste

Jeffrey Stylos, Brad A. Myers, Andrew Faulring
Computer Science Department and Human-Computer Interaction Institute
Carnegie Mellon University
Pittsburgh, PA 15213, USA
{ jsstylos, bam, faulring }@cs.cmu.edu
http://www.cs.cmu.edu/~citrine/

ABSTRACT

We present Citrine, a system that extends the widespread copy-and-paste interaction technique with intelligent transformations, making it useful in more situations. Citrine uses text parsing to find the structure in copied text and allows users to paste the structured information, which might have many pieces, in a single paste operation. For example, using Citrine, a user can copy the text of a meeting request and add it to the Outlook calendar with a single paste. In applications such as Excel, users can teach Citrine by example how to copy and paste data by showing it which fields go into which columns, and can use this to copy or paste many items at a time in a user-defined manner. Citrine can be used with a wide variety of applications and types of data and can be easily extended to work with more. It currently includes parsers that recognize contact information, calendar appointments and bibliographic citations. It works with Internet Explorer, Outlook, Excel, Palm Desktop, EndNote and other applications. Citrine is available to download on the internet.

Author Keywords

Copy-and-paste, web-pasting, intelligent user interfaces.

ACM Classification Keywords

H.5.2. User Interfaces: Interaction Styles.

INTRODUCTION

A common and tedious task encountered while using a computer is that of taking data from one source—an email, webpage, or program—and entering it somewhere else—into a web-based form, contact manager, bibliographic database, or other program. The technique of copying and pasting is useful and easy to use, but is limited, especially
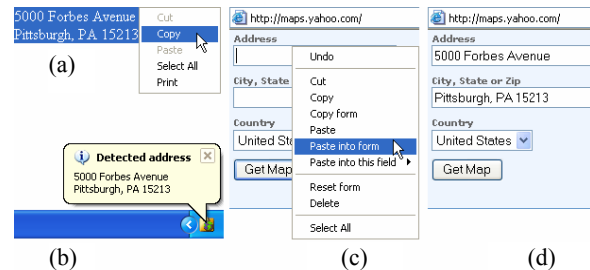
Figure 1: Using Citrine to paste an address

when dealing with data that has multiple parts. For example, entering contact information from an email signature into a contact manager might require copying (or manually retyping) a dozen different fields, and require the user to switch between the source and destination window with each copy and paste. Similarly, filling in a web-based form might require a separate copy and paste for each field.

We extend the copy-and-paste interaction technique to work better for structured text by using a centralized text-parsing and format conversion framework that supports multiple pluggable text parsers and format converters.

The framework is embodied in Citrine, a new tool that intelligently infers the structure of copied text, adds new representations of the data to the clipboard, and modifies existing applications when necessary to support pasting the new structured data representations. We also modify existing applications to be able to copy structured data in cases where the applications do not support this (for example, to copy multiple fields from a web-form).

Figure 1 shows an example of using Citrine to paste an address. In part (a), the text:

> *5000 Forbes Avenue*
> *Pittsburgh, PA 15213*

is copied from a webpage, causing Citrine to recognize that an address has been copied (b). A user can then paste this into a web-based form such as on Yahoo! Maps, which has several fields, with a single paste operation (c), and all the fields are filled in (d).

Citrine works with many different types of data and applications. It recognizes contact information and creates stan-

dard contact clipboard formats that can be pasted into Outlook, Palm Desktop, Netscape address book, CardScan address book and many other applications. It recognizes paper citations and creates a format compatible with the EndNote reference manager. It recognizes calendar appointments, which with the help of application plug-ins, can be pasted into Outlook and Palm Desktop. All of the above data types can be pasted into Internet Explorer or Excel, where plug-ins let the user define by example how the fields should be mapped to the different fields and columns. After being trained once, Citrine remembers these associations and uses them to paste subsequent items.

Citrine is extensible in a number of ways. First, its parsing can be extended so that it can recognize new types of data. Second, its clipboard-format generation can be extended, so that it can easily provide compatibility without modifications to new applications. And third, it can be extended with new application plug-ins that can easily use Citrine's custom clipboard data format.

Citrine differs from existing techniques [2, 5, 6] in three important ways. First, Citrine operates *between* the regular copy and paste actions. So, unlike SmartTags [3], which require modified source applications, or "paste special," which requires modified destination applications, Citrine works transparently in between the source and destination. This allows it to more easily support compatibility with a broad range of unmodified applications and also does not require the user to learn a new user interface. Second, Citrine works with multiple fields of data at the same time, even when the fields are unlabeled—that is, it can detect a multi-field structure from a single copy, and paste the various pieces with a single paste operation. Finally, Citrine provides customized pasting by allowing users to teach it how to paste into web-forms and spreadsheets.

DESCRIPTION OF CITRINE
This section presents Citrine: "Clipboard Interaction Techniques that Recognize Information such as Names and Events."

User Interface
Wherever possible, Citrine uses the applications' own unmodified copy-and-paste user interfaces and is therefore transparent to the user. When applications must be extended with plug-ins, we try to make the added functionality match the existing user interfaces for copying and pasting, as in Figure 1c.

Because Citrine is often transparent—using existing applications' copy and paste functions—we optionally provide some non-intrusive feedback to notify the user that something has taken place and that now they can paste into a wider range of applications. First, Citrine displays an icon in the system tray that changes to represent the format of the data detected on the clipboard (see Figure 2). Second, Citrine optionally pops up a notification of the inferred format (Figure 1b).

Figure 2: Icons used by Citrine to represent the format it has detected on the clipboard. From left to right: an empty clipboard, text, an image, a citation, an address, a contact, an appointment, and other data.

Citrine includes a tool for viewing and editing the clipboard text and recognized fields. This is a way for users to enter new information or correct parser mistakes. We plan to use this interface in the future as a way for users to provide feedback that could improve our parsers when machine learning techniques are added (see future work section).

Clipboard Formats
The clipboard in Microsoft Windows (and other operating systems) supports multiple representations (formats) of the same data. A key feature of Citrine is that it adds formats to the clipboard. Citrine watches the clipboard for changes. When new text is copied, Citrine tries to recognize the content, and if so, it adds new formats to the clipboard. It does this without disturbing the formats that were already on the clipboard. Figure 3 shows our debugging application displaying the list of formats on the clipboard before and after Citrine recognizes contact information.
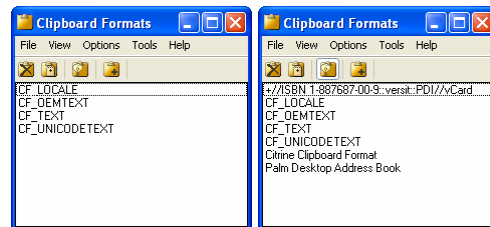

Figure 3: Clipboard formats before and after Citrine recognizes contact information

The new formats on the clipboard cause applications that recognize these formats to enable their paste functions, which otherwise would be disabled when the clipboard only contains plain text. After Citrine adds formats to the clipboard, applications that previously allowed the pasting of the plain text still work as before. Applications that previously allowed pasting of plain text but *also* allow pasting of contacts (such as Outlook's contact manager, which pastes plain text by creating an empty new contact with the clipboard contents in the "notes" field) will now paste the preferred contact format instead, which fills in contact fields such as "name" and "email."

Citrine can also convert between existing structured data formats. Unlike conversions of plain text, these conversions can be done unambiguously. This is useful for providing clipboard compatibility between applications. For example, by automatically generating additional contact formats, Citrine enables copying and pasting between the Netscape and Outlook address books, which otherwise is not supported. The same type of conversions also enables

copying and pasting appointments between the Palm Desktop and Outlook calendars.

Application Plug-Ins
Many applications natively support the copying and pasting of some clipboard format and work with Citrine without modification. However, some applications do not support any copying and pasting of particular data at all. For these applications, we created plug-ins that extend the applications to allow users to copy and paste. For example, the Palm Desktop calendar does not support the copying or pasting of appointments at all without the Citrine plug-in. The plug-in adds new "Copy appointment" and "Paste appointment" buttons to Palm Desktop.

We created a special "Paste appointment" function in Outlook's calendar as well. Outlook natively supports pasting appointments only as plain text into the currently selected date and time. Our added paste function creates a new appointment window with the subject, location, and date and time fields filled in using the information from the copied text. Showing the new appointment window allows the user to verify that the detected fields are correct and to modify any fields before saving an appointment.

Web-Pasting
We also extended Internet Explorer to support pasting into multiple fields of a web-form at once. In this case, there is an existing paste function available, but it only pastes into one field at a time, and pastes only the first line of text when pasting into regular edit fields.

We created an additional function, "Paste into this form," which fills in multiple form fields at once. For example, Figure 1 showed how a copied address can be pasted into Yahoo! Maps with one action. The plug-in fills in the fields by matching the names of the form fields with the fields of the parsed data in the clipboard. For example, it matches the field from Yahoo! Maps' form that has the internal name of "addr" with the street-address field of a parsed address. Some fields need to be filled in with multiple items. Citrine maps the "City, State or Zip" field in the Yahoo! Maps page to the city, state and zip-code fields of a recognized address and combines the fields appropriately with spaces and a comma. Furthermore, Citrine can fill in fields that use pop-up menus, such as the state field on many forms. The plug-in is not specific to pasting addresses, and can paste any structures that Citrine's parsers recognize.

Citrine has a default set of mappings of form fields that are designed to work with many common websites. Users can teach Citrine about new websites by demonstrating which piece of the source data goes in which form field. This uses an added function, "Paste into this field," which lets users explicitly paste a single piece of the source data into one field. After doing this once, the "Paste into this form" function will continue to use the demonstrated mappings to fill in the fields when pasting new data into the form.

Excel
Citrine includes a plug-in for Excel that lets users copy and paste objects with different fields from and into spreadsheet cells. In addition to the features of the Internet Explorer plug-in, the Excel plug-in also enables the pasting of *multiple* clipboard items at the same time, into consecutive rows. For example, a user can select many different contacts in the Palm Desktop address book (by holding down CTRL while selecting new items, for example) and then paste all of them into an Excel spreadsheet with one "Paste all items into rows" menu action.

IMPLEMENTATION
Citrine consists of a main application that runs in the background and a collection of application plug-ins.

Main Application
The main Citrine application watches the clipboard to detect copy-events, parses the available clipboard formats, and uses this information to generate new clipboard formats. There are two main types of components in this architecture: parsers and format generators. The Citrine-specific clipboard format is the sole method of communication between these two types of components. The components are separate from each other and new ones of each can be easily added.

There are two types of parsers, structured-format based parsers, and heuristic plain-text parsers. Each of these parsers performs two tasks: deciding if the clipboard contains relevant information, and if so, extracting and labeling it and adding it to the Citrine-specific clipboard format.

The structured-format based parsers unambiguously translate structured clipboard formats into the Citrine-specific format. For example, copying a contact from Palm Desktop causes Citrine to extract each of the fields and saves the data in the Citrine-specific clipboard format. This is useful for pasting with Citrine plug-ins and also enables the conversion from one structured format to another. The structured-format based parsers were easy to create for documented clipboard formats but more difficult to create for private, undocumented formats (as was often the case). For undocumented formats, we performed much experimentation to reverse engineer the formats used. The structured-format based parsers decide if the clipboard contains relevant information by checking to see if their targeted format is on the clipboard.

Much of Citrine's power comes from the heuristic text based parsers. When text is copied, each heuristic parser decides if the copied text contains relevant information. If so, it extracts and labels all of the fields of the structure it can and adds them to the Citrine-specific clipboard format. When text contains multiple types of data—such as an event that includes contact information—both parsers decide that the text contains relevant information, both add their structures to Citrine's clipboard format, and both will

have format generators create additional clipboard formats. Because the clipboard can hold many different formats at the same time, this is not a conflict.

Citrine currently includes plain text parsers that use ad-hoc, but well tuned, parsing techniques, rather than the more advanced techniques from the fields of information extraction or text classification [1]. The current parsers look for individual fields of structured information by checking for patterns, such as the number pattern of a phone number, and keywords, such as "Fax," "Researcher," or "University." The parsers also look for some fields based on the location of text. For example, the contact parser looks in the beginning of copied text for a name. To decide whether text contains relevant information, the parsers first attempt to parse it and then check to see whether it contains a sufficient set of fields.

The Citrine architecture allows new parsers to be added with little additional work. The programming interface also can provide feedback to the parsers so that parsers that support learning can be told about user corrections. After saving changes, the dialog sends the new structure and original text back to the parsers. Though our current hand-coded parsers cannot use this additional information, we plan to implement more sophisticated parsers that can [1, 4].

The structured-format parsers run before the heuristic parsers, so that when both structured and plain text formats are on the clipboard, the unambiguous, structured parsers are used instead.

The Citrine-specific clipboard format, "Citrine Clipboard Format" shown in Figure 3, serves as the means of communication between the parsers and format generators and is also created and used by the application plug-ins. The format is based on XML and is easily parsed by plug-ins. It consists of a list of structures labeled by type (such as contact or citation). Each structure contains the pre-defined values for each type, and has room for extra user-defined names and values as well. The format can accommodate multiple types of structures and multiple instances of structures of any given type at the same time.

The format generators take the labeled information from the Citrine-specific clipboard format and, if it contains a relevant structure, use it to generate other structured clipboard formats. This is much the same as the task of the structured-format parsers, but in reverse.

Application Plug-Ins
We created plug-ins for the Palm Desktop, Outlook, Internet Explorer and Excel. The plug-ins were written using their applications' respective APIs and communicate with each other and the main Citrine application only through the Citrine-specific clipboard format on the clipboard.

FUTURE WORK
Citrine is being created as part of the RADAR project (http://www.radar.cs.cmu.edu/). Through the work of this project we hope to have access to many intelligent parsers (e.g., [1]) that use state of the art machine learning techniques and can be integrated with Citrine. These techniques could learn the structure of new types of data given examples by the user [2] and improve its parsing accuracy over time by observing corrections made by the user to recognized data. Such techniques could also be used to learn a user's preferences in interpreting parsed data—for example, how concise or verbose a user likes the subject field of an appointment to be.

CONCLUSIONS
The Citrine software has been available for general use for about six months, and we have found it very useful ourselves for entering calendar information, copying contacts, and adding new entries to our bibliography. You can try it for yourself by downloading it from http://www.cs.cmu.edu/~citrine/. We performed a small user study that showed that people took less time and made fewer errors when using Citrine to copy information, even when Citrine did not parse perfectly, and these results were statistically significant. As more features are added to Citrine, we expect that its utility will only increase.

REFERENCES
1. Cohen, W.W. and Sarawagi, S., "Exploiting Dictionaries in Named Entity Extraction: Combining Semi-Markov Extraction Processes and Data Integration Methods." 2004. Submitted for Publication.
2. Lieberman, H., Nardi, B.A., and Wright, D. "Training Agents to Recognize Text by Example," in *ACM Conference on Autonomous Agents*. 1999. Seattle, WA: ACM Press. pp. 116–122.
3. Microsoft, "Microsoft Office XP Smart Tags," http://www.microsoft.com/technet/treeview/default.asp?url=/technet/prodtechnol/office/officexp/maintain/xptags.asp.
4. Myers, B.A. "Text Formatting by Demonstration," in *Proceedings SIGCHI'91: Human Factors in Computing Systems*. 1991. N.O., LA: pp. 251-256.
5. Nardi, B.A., Miller, J.R., and Wright, D.J., "Collaborative, programmable intelligent agents." *Communications of the ACM*, 1998. **41**(3): pp. 96–104. Apple Data Detectors.
6. Pandit, M.S. and Kalbag, S. "The selection recognition agent: instant access to relevant information and operations," in *International Conference on Intelligent User Interfaces*. 1997. Orlando, FL: ACM Press. pp. 47–52.