2009 International Conference on Computer Engineering and Applications
IPCSIT vol.2 (2011) © (2011) IACSIT Press, Singapore

An Application Tool for Collecting Real Auction Data

Rodel Balingit[†], Jarrod Trevathan, Yong Jin Lee and Wayne Read

Discipline of Information Technology, School of Business, James Cook University Australia

Abstract. Online auctions are a rapidly growing platform to exchange items of just about anything, from common to collectors' items in different region around the world. Researchers continue their efforts to understand further the real behaviour and bidding patterns of buyers and sellers, the fraudulent traits and even devise countermeasures to prevent auction fraud. Most often, the researchers' noble work ends with frustration due to unavailability of real auction data. Online auction sources are unwilling to provide the real auction data and often cite "commercial, security and privacy" as reasons. This paper presents an application tool that collects real auction data from an ongoing auction for a given criteria, and automatically returns later to collect the data from a recently completed auction. From the knowledge of the authors, the work presented here is the first serious attempt to create an openly available application tool that will be free for use by other researchers.

Keywords: e-Commerce, Software Design, Parsing, Reporting Features, Statistics.

1. Introduction

eBay[1], the world's largest online auction service is becoming more popular each day. Millions of people use eBay for its convenience and ability to sell commonly available and rare items, not typically found in regular retail stores of that region. However, making bidders' identity completely anonymous is very much harder to determine whether a specific bidder (or group of bidders) that is responsible for submitting the majority of the bids. Additionally, hiding bidders' ID has the effect of difficulty to understand bidder behaviour and the tracking of fraudulent bidders, as it would be simply easy for them to get away from any fraudulent activities. This fraudulent activity might include artificially inflating the auction's price using fake bids (referred to as *shill bidding* [1]), selling stolen items, or misrepresenting an item.

Several anti-fraud countermeasures are being designed to detect and investigate fraudulent activities (see [1], [2], [3], [4]). In order to elevate the effectiveness of the proposed countermeasures, the researchers require real auction data from online auctions. This data can be use to analyse bidding trends that suggests to fraudulent behaviour and to see the affect on buyers' and sellers' strategies given the presence of anti-fraud mechanisms. However, obtaining a significant amount of real auction data is not simple.

Most commercial online auctioneers are unwilling to provide data and often cite "commercial, security and privacy" as reasons. It is true that these reasons are valid concern. However, the type of data required does not affect these factors (i.e., auctioneers' reputation, sales, buyer's and seller's identity confidentiality, etc.). All that is required is the bidding history (e.g., Bidder's ID, price, time and date submitted). The bidders and sellers identities do not need to be disclosed, as long as they are uniquely identified. In addition, auctioneers need no fear, as researchers are very much willing to cooperate and abide to auctioneers' set of legal guidelines (e.g., how to use the auction data, the exposition of a fraudster (or a con artist) once found from the auction data provided).

Although online auction sources are hesitant to provide the real auction data, researchers left no choice but to continue collecting real auction data from various online auction sources. There are existing methods used to acquire this real auction data. For example, some researchers have manually "cut and pasted" data from auction pages (e.g., Jank and Shmueli [5]), whereas other researchers automate the process using software tools (see Rubin et. al. [3], Shah et. al [4]). Additionally, some researchers have conducted their own auctions (see Trevathan and Read [6]), or used software bidding agents to artificially generate auction data (see Hattori et. al. [7], Trevathan and Read [8]).

[1] http://www.ebay.com

[†] *Corresponding author. Tel.: (+617) – 4781 6913. E-mail address: rodel.balingit@jcu.edu.au.*

Furthermore, a commercial tool called *Auction Data Retriever*[2] that allows customers to harvest auction data directly from eBay web pages to their own customised auction management system. Customers can easily import the data to Microsoft Excel or Access. The system's fundamental problem is that it is sales-driven and only retrieves basic sales information for a seller (but doesn't actually retrieve the bid history for any particular auction). Another deterring factor is that it incurs costs to subscribe to the system and no means to customise the type and amount of data collected. It appears that commercial systems are not really ideal to conducting research into online auction fraud. Regardless of the collection means, there appears to be no standard or documented manner in which the data collection has occurred.

This paper presents an application tool for collecting real auction data from eBay. Once a user conducts a search on eBay, s/he enters the URL containing the auction listings. The system scans through the listed auctions and extracts the item information and bid history. The application tool collects real auction data from an ongoing auction of a given criteria and automatically returns later to collect the data of a recently completed auction (without user intervention). The data is parsed and then stored in a database. The system also provides reporting features and statistics on the collected data. From the knowledge of the authors, the work presented here is the first serious attempt to create an openly available application tool that will be free for use by other researchers. (Note that in this paper we will be focusing on auctions run by eBay as an example auction site, however, the system can be adapted to extract auction data from other online auctioneers.)

This paper is organised as follows: Section 2 describes the core components of the application tool, the process to collect and store the auction data, and also the basic reporting features to analyse the collected data. Section 3 gives a basic performance analysis of the system and Section 4 describes some implementation drawbacks. Section 5 provides some concluding remarks and avenues for future work.

2. Software Model and Design

This section presents the software model and design of the application tool that extracts auction data. It describes the process of collecting the auction data, the core components and storage of the data, and reporting capabilities.

2.1. High Level Design

Figure 1 presents the high level software design of the application tool. The tool sits on a user's computer and interacts with eBay across the web. When commenced, the tool remains online until the extraction of the last auction is completed. For auction that has not completed during the extraction process, the tool records the finish time and then revisits the auction once it has completed to extract its final data. The results (i.e., the auction data) are written to a database. At any time the user can interact with the software by a user interface. Once started, the user can leave the application tool running as a batch process. The tool can also generate and write data to an Excel file in preparation for further statistical analysis.

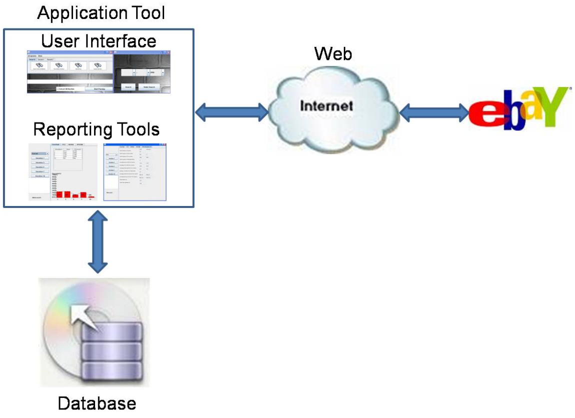


Fig. 1: High Level Software Design.

2.2. Data Collection Process

eBay offers differing buying formats (e.g., *Auctions, Fixed Price, Advertisement*) for buyers and sellers to conduct their business (see [9] for details). The auction data is collected from *Auctions* buying format as it

[2] http://rajeware.com/

depicts the interaction of bidding among competitive buyers. The application tool categorises an auction as either an *ongoing auction* or a *completed auction*. Generally, the tool collects data from an ongoing auction by individually parsing the *Auction Items page* list, then parses/extracts the *Individual Auction Item* details and bid history information. The tool can perform automatic updates of a recently completed auction while the tool is busy extracting data from other auctions.

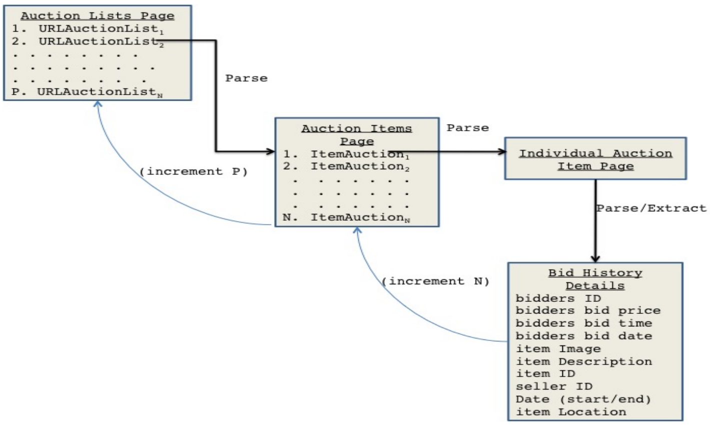


Fig. 2: The process involved with collecting data from an online auction source.

Figure 2 illustrates the process by which the tool collects auction data from eBay. Once the system has been initialised (i.e., set to the search page), the tool constructs a *URLAuctionList* page. This is done by extracting the URLs for each auction from the raw HTML in the search page. For each URL page, the *Individual Auction Item* list page is parsed and the following information is extracted:

- ***Item ID***, ***Seller ID***, ***Item image***, ***Item description***, ***Date (start/end)*** and ***Item location.***

Next, the link to the item's *Bid History* is followed and the following information is extracted for each bid submitted:

- ***Bidder's ID***, ***Price***, and ***Time and date submitted.***

This data is cached until the last *Individual Auction Item* on that page is completely parsed and extracted. The cached data is written individually to a text file and then moves to the next auction in the *URLAuctionList page* list. This process is repeated until the last *URLAuctionList page* is reached.

2.3. Core Components

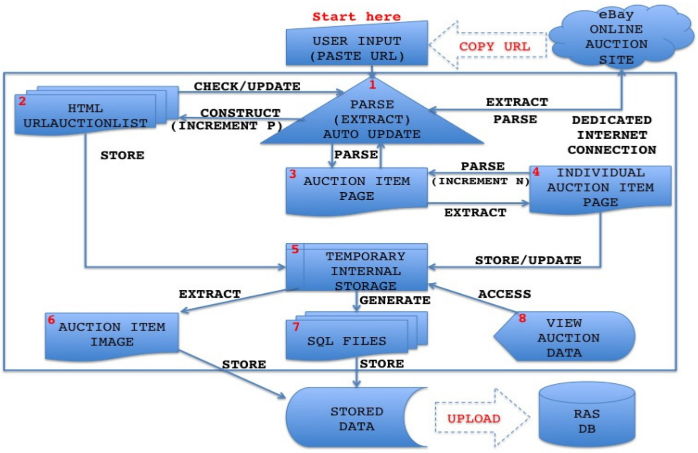


Fig. 3: The internal architecture of the application tool.

Figure 3 illustrates the internal architecture of the processes involved for extracting and storing the auction data. There are several main components:

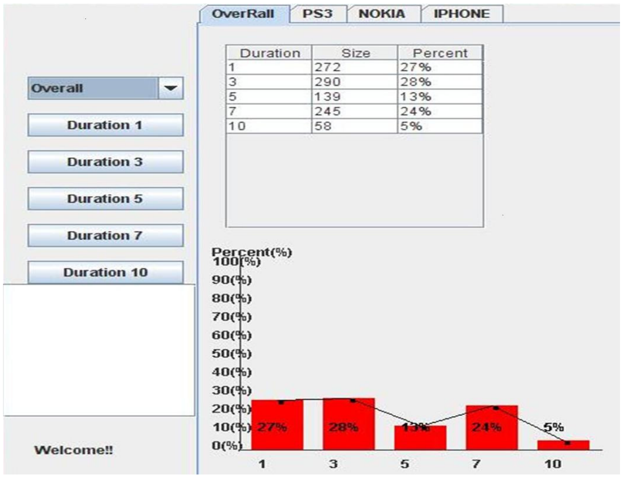
1. *PARSE/EXTRACT/AUTO UPDATE (PEAU)* - This parses eBay's auction HTML documents, extracts auction data for each item on auction, and performs an auto update of a recently completed auction. Both the parse and extract processes are performed simultaneously. During this process, PEAU can detect if a specific auction has completed, and then automatically

updates the previously extracted auction data. The auction data is cached during the parse/extract process and written to *TEMPORARY INTERNAL STORAGE*.

2. *HTML URLAUCTIONLIST (HURL)* - This holds a constructed *URLAuctionList page* during the parse/extract process. The *URLAuctionList* page is appended with additional *URLAuctionList* data as the parse/extract process moves to the next page (*increment P* in Figure 2) until the operation is completed. The constructed list is used as reference to monitor and update the extracted auction data from recently completed auctions.
3. *AUCTION ITEM PAGE (AuIP)* - This holds a list of items on an auction as per the *URLAuctionList* page. Each item parsed is checked to determine if submitted bids are found. Then, it moves to the *IvAuIP*, otherwise it continues to parse the next auctioned item (*increment N* in Figure 2), or goes to the next *URLAuctionList page* as the last auctioned item is reached.
4. *INDIVIDUAL AUCTION ITEM PAGE (IvAuIP)* - An IvAuIP holds the bid history details. At this point, an extraction of data commences and is then stored in *TEMPORARY INTERNAL STORAGE*.
5. *TEMPORARY INTERNAL STORAGE* - This is a central repository of an extracted auction data from eBay. The data stored are organised and prepared for file creation like *url.txt*, *auction.sql*, *bidinfor.sql*, *updatebidinfor.sql*, and image file. The *url.txt* file is a list of URLs collected from eBay. The files *auction*, *bidinfor* and *updatebidinfor* are SQL command procedures used for inserting the data directly into the database.
6. *AUCTION ITEM IMAGE (AII)* - This extracts the images for each item auctioned.
7. *SQL QUERY FILES* - This creates a SQL stored procedure to insert extracted auction data into a RAS-style auction database. It creates three files: *auction.sql*, *bidinfor.sql* and *updatebidinfor.sql*.
8. *VIEW AUCTION DATA (VAD)* - This allows a user to view the extracted auction data. The view feature includes auction data, URL list and updates of recently completed auctions.

2.4. Reporting Features

The real auction data collected is spanning from the three categories of items (i.e., *Playstation 3, Nokia N95 8GB* mobile phone, *Apple iPhone*).



Duration	Size	Percent
1	272	27%
3	290	28%
5	139	13%
7	245	24%
10	58	5%

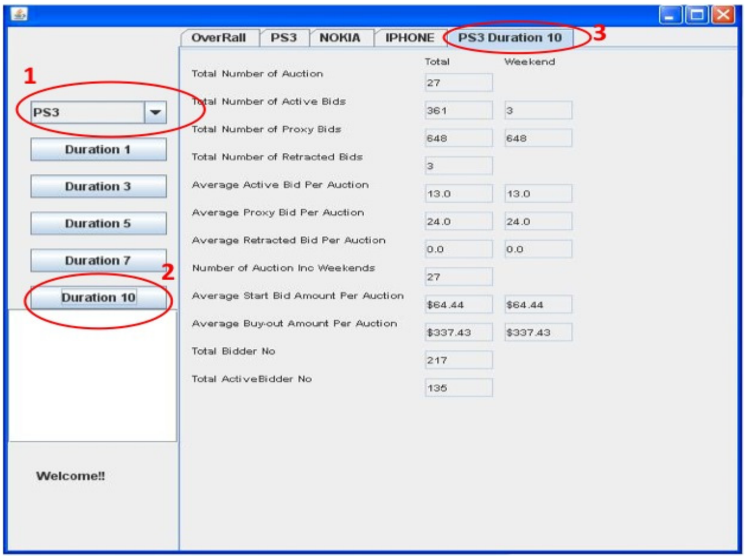
Fig. 5: The overall report of auction data collected from the three categories of items.

Figure 5 shows the overall report of the total number of auctions extracted by the application tool that are of 1-day, 3-day, 5-day, 7-day and 10-day durations. Once selected, the tab shows individually the information and a graph of the total number of auctions collected under a certain category of item. For each item, the total number of auctions collected is distributed in percentage across of a 1-day, 3-day, 5-day, 7-day and 10-day duration.

Furthermore, Figure 6 shows a basic statistical report of a collected auction data for a certain category of item in certain duration. For example, Figure 6 shows the basic statistical report of *Playstation 3* that is held in a 10-day duration. In order to obtain the basic statistical report, the user must follow these steps:

1. The user selects the particular item (e.g., PS3, Nokia, iPhone) from the dropdown menu. The application tool processes immediately the selected category of item.
2. The user selects a button (for a certain duration) the auction is held. The application tool generates the basic report and display the results individually (refer to Figure 6, *no. 3*) and;
3. The user selects the tab generated by the system to display the details of the report.

The details of the report includes the following: *total number of Auction, total number of active bids, total number of proxy bids, total number of retracted bids, average active bid per auction, average proxy bid per auction, average retracted bids per auction, number of auctions held on weekends, average start bid amount per auction, average final bid price per auction, total number of bidders and total number of active bidders*. The reported information is then used to further our study in analysing bidding trends, and to detect and investigate online auction fraud.



	Total	Weekend
Total Number of Auction	27	
Total Number of Active Bids	361	3
Total Number of Proxy Bids	648	648
Total Number of Retracted Bids	3	
Average Active Bid Per Auction	13.0	13.0
Average Proxy Bid Per Auction	24.0	24.0
Average Retracted Bid Per Auction	0.0	0.0
Number of Auction Inc Weekends	27	
Average Start Bid Amount Per Auction	$64.44	$64.44
Average Buy-out Amount Per Auction	$337.43	$337.43
Total Bidder No	217	
Total ActiveBidder No	135	

Fig. 6: The basic statistical report of a collected auction data from *Playstation 3* that is held in a 10-day duration.

3. Complexity Analysis

This section performs a basic complexity analysis of the application tool that collects real auction data from online auctions. The factors that influence the system's running time includes: 1) the number of auctions to extract; and 2) the finishing times of the auctions. Assuming that all auctions have completed, the execution time is directly proportional to the number of auctions extracted. Given ***n*** (the number of auctions), the run time can be denoted as ***O(n)***.

Prior to commence the extraction of auction data, the tool parses/extracts first the auction's URLs from auctioneer's page. The tool individually open the HTML file and sequentially scan the HTML tags to remove irrelevant information. This process is very similar for extracting the auction data and bidding history of every auction.

Note that the application tool extracts data from ongoing and completed auctions. When it encounters an ongoing auction, the tool must later return to that auction to extract the final auction data. The tool does this by recording the auction's closing time, and then returns later to extract the final data when the user's computer's system clock indicates that the time on the auction server has passed the auction's closing time. Using this approach, the system does not need to continually poll the auction server (as in the approach by Shah et. al. [4]). The only load the system places on the auction server is a one off extraction of the search page and each relevant auction page. Given ***n*** auctions with ***m*** ongoing auctions (where ***m*** <= ***n***), the system places a maximum load on the auction server of ***O(n+m)*** (in addition to the extraction of URLs from the search page).

4. Implementation

This section briefly describes the specifics of the implementation and drawbacks encountered in using the application tool. The tool is developed and implemented using the Java programming language. Several versions were implemented and unit tested prior to the commencement of a concerted acquisition attempt to obtain the auction data. The tool is run on several standard desktop computers to extract the auction data, simultaneously. Each computer extracted the data from different search criteria for different auction. At the end of the process, all of the auction data was collated and stored in a central repository.

At the time of writing, the application tool has collected data from over 1300 auctions spanning three categories of items. The data has taken one month to collect. There were several major impediments to the time it took to collect this data. The first constraint is the number of auctions available for the desired search criteria. Next the tool is constrained by the duration of each of the auctions. For example, some auctions can run for a day, whereas others may run for ten days. Obviously, more data can be collected for a large number of auctions that run for a short duration.

A significant limiting factor specific to our data collection was a proxy authentication system within our research institution. To gain external Internet access, each user (or process) must authenticate to the proxy. The authentication remains active for twelve hours of continuous use before requiring re-authentication. The proxy authentication essentially denied the application tool's external access to eBay when performing a batch job (i.e., it exceeded the twelve hours of activity), thereby forcing the application tool to be restarted.

5. Conclusions

This paper presented an application tool to collect data from eBay's new auction structure. This work is motivated by the uncooperative nature of commercial auctioneers to provide auction data for the purposes of testing auction fraud detection/prevention mechanisms that are proposed by researchers. The application tool parses/extracts auction data from both ongoing and completed auctions, and collects description information pertaining to the item auctioned as well as its bidding history. The collected data can be exported to an Excel spread sheet to graph for statistical analysis. Additionally, the auction data collected can be uploaded to a customised auction server. The work presented in this paper represents the first serious attempt at creating an openly available application tool and establishing a repository of online auction data that will be free for use by other researchers.

Future work involves opening the application tool to collaboration with other researchers to improve its performance (e.g., increase its efficiency to extract auction data, revise the software model and architecture design, etc.). The database of collected data will be place online for other researchers to copy and/or add their own extracted auction data. We also intend on allowing the application tool to extract all the completed auctions for a specific seller, and adding additional statistical functionality. Furthermore, there is a scope to implement basic data mining algorithms in an attempt to analyse previously unknown trends in the auction data over time.

6. References

[1] J. Trevathan and W. Read, "*Detecting Shill Bidding in Online English Auctions*," in *Social and Human Elements of Information Security: Emerging Trends and Countermeasures*: IGA Press, 2008, pp. 446 - 470.

[2] Y. Cheng and H. Xu, "*A Formal Approach to Detecting Shilling Behaviours in Concurrent Online Auctions*," in *Proceedings of the 8th International Conference on Enterprise Information Systems (ICEIS 2006)* Paphos Cyprus, 2006, pp. 375 - 381.

[3] S. Rubin, M. Christodorescu, V. Ganapathy, J. Giffin, L. Kouger, and H. Wang, "*An Auctioning Reputation System Based on Anomaly Detection*," in *the Proceedings of the 12th ACM Conference on Computer and Communication Security (CCS)*, Alexandria Virginia, 2005, pp. 270 - 279.

[4] H. Shah, N. Joshi, and P. Wurman, "*Mining Strategies of eBay*," in *SIGKDD 2002 Workshop on Web Mining for Usage Patterns and User Profiles*, 2002.

[5] Jank and Shmueli, "Dynamic Profiling of Online Auctions Using Curve Clustering," in *Proceedings of the 11th Annual Spring Research Conference (SRC) on Statistics in Industry and Technology*, 2004.

[6] J. Trevathan and W. Read, "Detecting Collusive Shill Bidding," in *the Proceedings of the 4th International Conference on Information Technology - New Generations*, 2007, pp. 799 - 808.

[7] H. Hattori, R. Yamada, T. Ozono, and T. Shintani, "*A Multiple-Bidding Support Framework for Bidding and Browsing Information*," Department of Intelligence and Computer Science Nagoya Institute of Technology, 2002.

[8] J. Trevathan and W. Read, "*RAS: a system for supporting research in online auctions*," in *ACM Crossroads*. vol. 12.4, 2006, pp. 23 - 30.

[9] eBay, "Guide to Buying Formats," http://pages.ebay.com.au/help/buy/formats-ov.html, Online, 04 August 2008.