Preprint of paper published in Journal of End User Computing, © Idea Group Publishing 2003.

CUSTOMER PERCEPTIONS OF A THIN-CLIENT MICRO-PAYMENT SYSTEM: ISSUES AND EXPERIENCES

Xiaoling Dai
Department of Computer Science
University of Auckland
Private Bag 92019, Auckland
New Zealand
xdai001@ec.auckland.ac.nz

John Grundy
Department of Electrical and Electronic Engineering and
Department of Computer Science
University of Auckland
Private Bag 92019, Auckland, New Zealand
john-g@cs.auckland.ac.nz

ABSTRACT

Two fundamental payment methods exist for on-line information purchase: macro-payment and micro-payment. Traditional macro-payment methods, like credit and charge cards and digital currency, are suitable for large-value, low-volume transactions. However, large-volume, low-value commodities, such as discrete units of information from a web site, better-suit a micro-payment model. In micro-payment, customers pay for large numbers of small value goods (e.g. per-web page view) with "e-coins", typically of very small value each. We have carried out an empirical assessment of micro-payment and macro-payment purchasing models for an on-line newspaper application. We report on the design of our experiment, the two kinds of micro-payment (client and server-side e-wallets) used, and customer feedback. We also carried out an assessment of customer effort and economic trade-off when using these services and compare the results of this assessment to a survey of customers using each system. We present directions for further on-line payment research aiming to improve the overall satisfaction and efficiency of payment models for end-users.

KEYWORDS

micro-payment, usability, e-commerce, e-money

INTRODUCTION

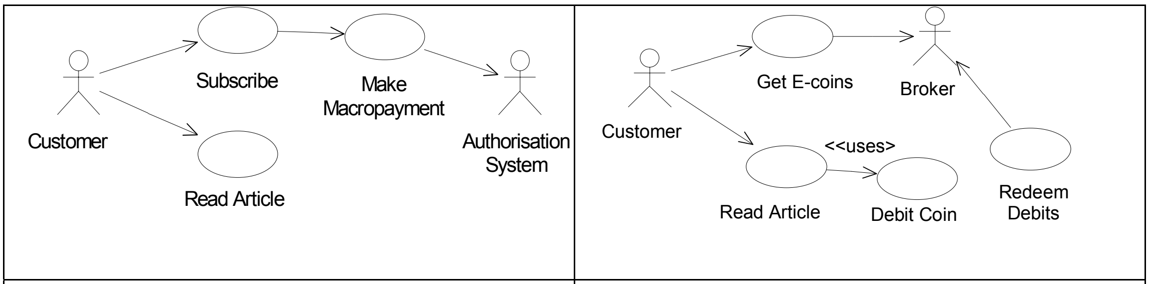
Most current E-tailing systems adopt a macro-payment model and architecture. A user makes a small number of on-line purchases that have a reasonably high purchase price. In order to pay for these purchases, a "heavy weight" interaction between the vendor of the product or service and an authorisation agent (bank, credit-card company etc) system is carried out. This typically involves the user supplying credit card details or "digital money" certificates, which are communicated to the authorisation system using complex encryption algorithms. Business processing logic and database updates are performed by the authoriser before the purchase is approved. The vendor system waits for approval before providing the customer with goods or services. This approach works well for relatively small numbers of transactions and relatively high purchase price (to offset the cost of authorisation) (Dai et al, 2001). However, in some e-commerce scenarios this approach has a number of fundamental flaws. It requires the authorisation system to always be on-line. High numbers of transactions or low-price purchase items are infeasible, due to bottle- necking or prohibitive cost per-transaction. In addition, with most approaches the customer's identity can not generally be hidden from the vendor. For example when using a subscription-based approach i.e. a single macro-payment for on-going supply of services, the vendor must be supplied with customer-identifying information. In addition, if the customer only makes use of a small fraction of subscribed services, they spend a comparatively high amount of money for what they use. Most subscriptions are to a single vendor's service and don't cover the purchase of low-value commodities from multiple vendors.

We describe the NetPay micro-payment model and architecture we have been developing. NetPay provides an off-line micro-payment model using light-weight hashing-based encryption. A customer buys a collection of "e-coins" using a macro-payment from a broker. These coins are cached in an "e-wallet" (stored either on the customer's machine or on broker and vendor server machines). The customer, when buying many small-cost items from a vendor, pays for these transparently by the passing of e-coins to the vendor. Periodically the vendor redeems the e-coins with the broker for "real" money. E-coins can be transparently exchanged between vendors when the customer moves to another site.

In this paper, we give an overview of the concept of micro-payment vs macro-payment models of e-payment. We outline our research methodology of assessing the perceptions of customers of micro-payment vs macro-payment models for E-tailing systems. We present the software architecture and design for NetPay, a new micro-payment system we have prototyped, for deployment with thin-client vendor interfaces for customers. We describe three kinds of experiments we have done on our NetPay prototype, to assess micro-payment vs macro-payment usability, performance and overall qualitative characteristics for E-tailing systems payment. We compared two kinds of NetPay-based micro-payment systems (client-side wallet and server-side wallet) and a subscription-based macro-payment system. We conclude with an outline of our further plans for research and development in this area.

MICROPAYMENT FOR E-TAILING SYSTEMS

Consider the scenario of customers wanting to browse on-line newspapers (Dai et al, 2001). Using the typical approach of subscription-based payment, the user would first have to subscribe to the newspaper by supplying personal details and payment details (credit card number etc). The newspaper system would then make an electronic debit to pay for their subscription by communicating with an authorisation server. The user would then normally go to the newspaper's site where they login with an assigned user name and password. The newspaper looks up their details and provides them access to the current edition if their subscription is still current. If the user's subscription has run out, they must renew this by authorising a further macro payment from their credit card. *Figure 1* (a) outlines the key interaction use cases for this scenario. Problems with this approach are that there is no anonymity for the user (the newspaper system knows exactly who they are and when and what they read), they can not browse other newspapers without first subscribing to them too, and they must pay for the whole newspaper, even if they want just one or two sections or articles. These issues apply to many other information sources on the internet where vendors want to charge for content (Blankenhorn, 2001; Herzberg, 1998).



(a) Typical macro-payment interaction model.

(b) Possible micro-payment interaction model.

Figure 1. Two on-line newspaper interaction scenarios.

An alternative approach is to use a "micro-payment" model. There are several approaches to micro-payment (Furche and Wrightson, 1996; Herzberg and Yochai, 1996; Hwang et al, 2001; Manasse, 1995; Rivest and Shamir, 1997; Stern and Vaudenay, 1997) - we outline the basic interactions of the NetPay model we have developed (Dai and Grundy, 2002). *Figure 1* (b) outlines the key interaction use cases for this scenario. The user first goes to a broker and purchases "E-coins" using a single macro-payment. These are stored in an "E-wallet", either on the user's machine or on the broker server. The user can then visit any vendor site they wish, for example an on-line newspaper. Each time they need to purchase a small-value item e.g. view an article (or section or page, depending on the item charged for) they give the vendor one or more E-coins of specified value to pay for this service. The vendor redeems these E-coins with the broker (for "real" money") periodically e.g. each night/week. The user can move to another site and unspent money associated with their E-coin is transferred from the first vendor to the second. If coins run out, the user communicates with the broker and authorises another macro-payment debit.

The standard macro-payment methods cannot be effectively or efficiently applied for buying inexpensive information goods, like single articles of an on-line newspaper, because transaction costs are too high (Furche and Wrightson, 1996; Hwang et al, 2001; Domingo-Ferrer and Herrera-Joancomarti, 1999). Encryption mechanisms used are slow and each transaction typically "costs" a few cents. Macro-payment suits spending small numbers of large amounts. In contrast an internet micro-payment system allows the spending of large numbers of small amounts of money at web sites in exchange for various content or services, as in the E-newspaper scenario above. The design of micro-payment systems is usually quite different from existing macro-payment systems, since micro-payment systems must be very simple, secure, and efficient, with a very low cost per transaction (Dai and Grundy, 2002). This must also be taken into consideration for transaction security: high security leads to high costs and computation time. For micro-payments a lower overall security threshold can be applied. Additional benefits of using a micro-payment approach include preserving the anonymity of the customer – the vendor needs have no information about customers in order to accept their e-coins for service payment.

RESEARCH METHOD

After having developed a new model for micro-payment for E-tailing applications, and building a prototype of this system, NetPay, we wanted to assess its worth compared to macro-payment systems. We had also developed two models for managing electronic coins ("e-coins") in our NetPay system – managing e-coins in a client-side electronic wallet ("e-wallet"), where the encoded coin information resides on a customer's computer, or in a server-side e-wallet, where the coin information resides on vendor servers and can be exchanged from vendor to vendor.

To carry out an evaluation of NetPay and compare its two e-wallet support approaches to traditional macro-payment based E-tailing payment methods we wanted to assess the characteristics of NetPay-based micro-payment systems from several perspectives. We wanted to assess and understand customer perceptions of using NetPay and the advantages and disadvantages they saw with the system, as well as gain an understanding of the usefulness of the approach for information vendors. Basically a micro-payment system should provide a lightweight mechanism for paying for on-line content where there are a large number of payments for quite small units of information (Dai et al, 2001; Hwang et al 2001; Domingo-Ferrer and Herrera-Joancomarti, 1999). Such a system needs to support both the buying of coins, or electonic money, by customers and per-click debiting of coins by vendors (Furche and Wrightson, 1996). The impact on the vendor system of debiting coins and tracking payments must be light to make the system practicable. We developed a new micro-payment model aiming to ensure light-weight, low-cost e-coin encryption via hashing, off-line micro-payment (a broker server doesn't need to be involved in every transaction made), protection from double-spending and vendor or customer forgery of coins or debits, and anonymous payment i.e. vendors do not know who customers are (Dai and Lo, 1999). This is achieved by the use of a hashing-based encryption scheme implemented by a central e-coin broker where customers buy coins and vendors redeem coins. Coins are encrypted as chains of complex numbers and the use of "touchstones" ensures that e-coins can be verified quickly and efficiently. We implemented prototypes of NetPay that augment vendor web sites to manage coin debiting on a pay-per-click basis (Dai and Grundy, 2002).

To evaluate this prototype we determined that three types of evaluation would be required: a usability evaluation, to assess the users perception of NetPay-provided features, and to compare these against conventional macro-payment subscription-based payment models (Neilsen, 1992; Preece, 1993). We focused on assessing usability via a using survey-based approach with representative target users of NetPay. A performance evaluation of our NetPay prototype was carried out to determine if adding it to typical vendor web servers would be viable for large transaction loading. Finally a qualitative assessment of NetPay and conventional macro-payment approaches was done to determine how well our model and prototype compare using some common assessment criteria. We analyses the results from these three evaluation approaches to determine if (1) NetPay is usable as far as target users are concerned; (2) the performance overhead of NetPay micro-payment would be acceptable to information vendors; and (3) that NetPay meets the requirements for a micro-payment system for E-tailing applications as outlined above. We describe in detail these evaluations and report on their results at the end of this paper.

OVERVIEW OF NETPAY

In this section we briefly describe our NetPay micro-payment protocol. We also outline the architecture of our NetPay prototype implementation and give some examples of an E-newspaper site augmented with NetPay-based micro-payment support.

NetPay Model

NetPay is a micro-payment model that allows customers to purchase information on a pay-per-click basis from vendors on the WWW (Dai and Lo, 1999). NetPay, a secure, cheap, widely available, and debit-based protocol of a micro-payment system, is used for purchasing on-line services via the WWW. NetPay differs from previous micro-payment protocols in the following ways: NetPay uses “touchstones” signed by the broker and coin index’s signed by vendors which are passed from vendor to vendor. The signed touchstone is used by a vendor to verify the electronic currency – the “paywords” encoding E-coins, and the signed index is used to prevent double spending by customers and to resolve disputes between vendors. There is no dependency on customer trust required with this approach.

A NetPay micro-payment system includes customers (e.g. newspaper customers), vendors (e.g. on-line e-newspapers) and a broker. In our approach we make the assumption that the broker is honest and is trusted by both the customers and the vendors. The micro-payments only involve customers and vendors, and the broker is responsible for the registration of customers and for crediting the vendors’ account and debiting customers’ accounts. *Figure 2* outlines some of the key NetPay system interactions.

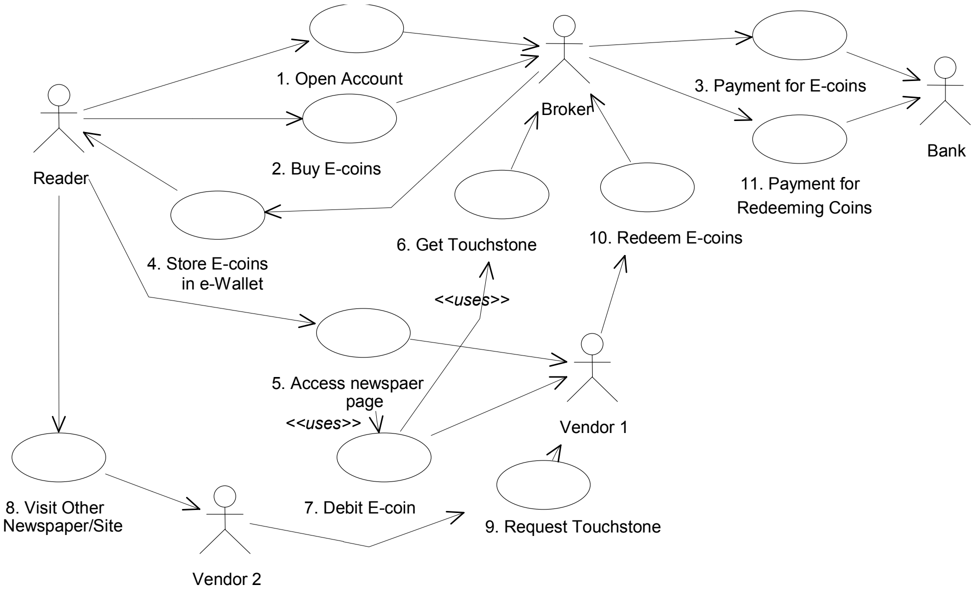


Figure 2. Basic NetPay component interactions.

Software Architecture

We have developed a software architecture for implementing NetPay-based micro-payment systems for thin-client web applications (Dai and Grundy, 2002). NetPay micro-payment transactions involve three key parties: the Broker Server, the Vendor Server, and the Customer browser. This architecture is illustrated in *Figure 3*.

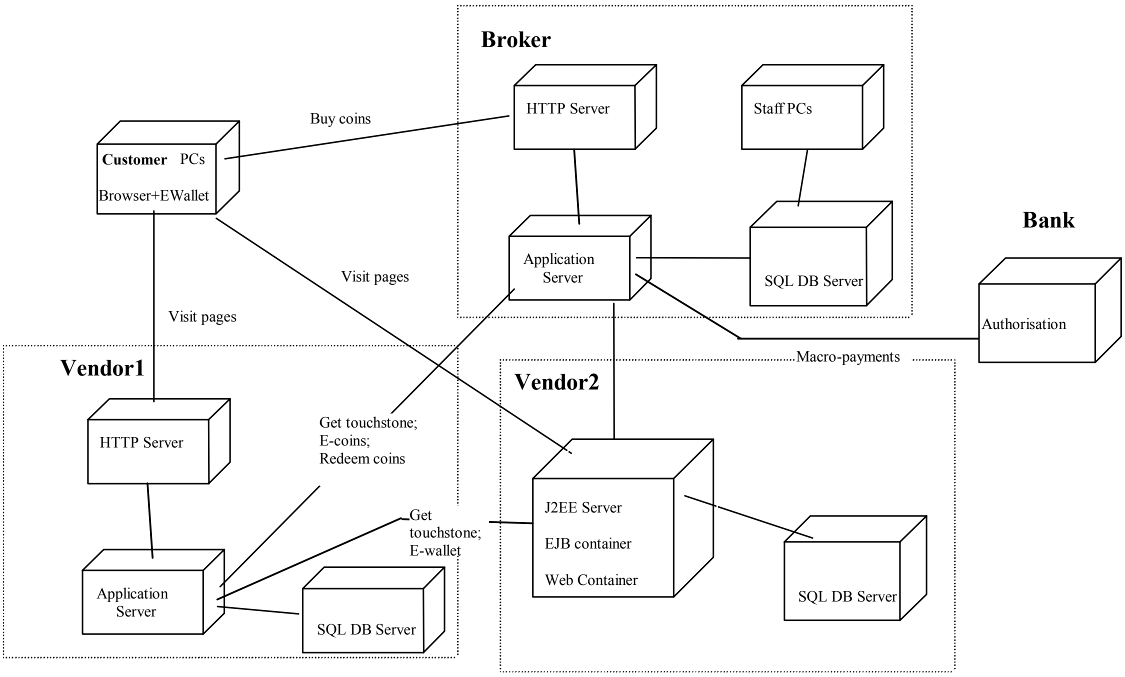


Figure 3. Basic NetPay software architecture.

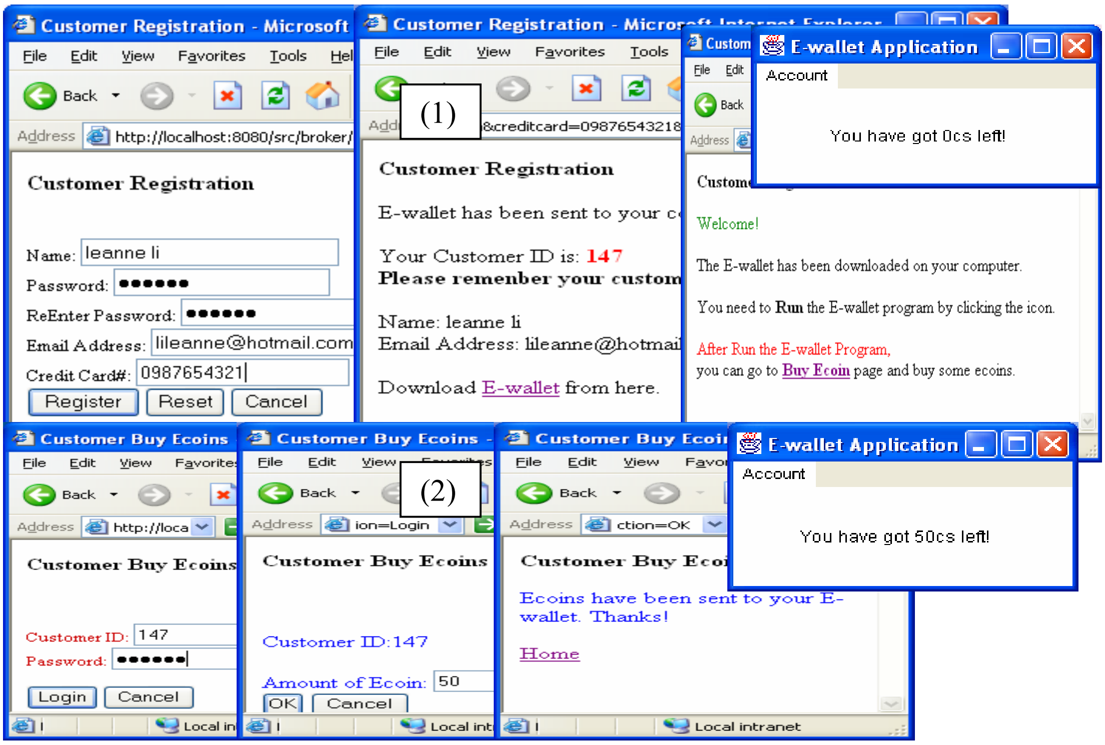
The **Broker** provides a database holding all customer and vendor account information, generated coins and payments, redeemed coins and macro-payments made (buying coins and redeeming money to vendors). The Broker application server provides a set of software interfaces vendor application servers communicate with to request touchstones and redeem e-coins. This server also communicates with one or more bank servers to authorise macro-payments (customer buying coins or broker paying vendors when redeeming spent coins). The Broker web server provides a point of access for customers to buy e-coins and check their e-wallet balances and transaction history.

The **Customer** runs a web browser that accesses the broker and vendor servers, and may also contain an e-wallet. In our current NetPay prototype we support the use of two kinds of e-wallet: one held server-side and one held client-side. The client-side e-wallet is an application running on the client PC holding e-coin information. The server-side e-wallet resides on the vendor server and is transferred from the broker to each vendor in turn the customer is buying content from. Each has its own advantages and disadvantages. When buying e-coins the Broker's application server updates the customer's e-wallet (cached e-coin information). When purchasing information using micro-payment, the vendor's web server accesses e-coin information using the customer's e-wallet.

The **Vendor** sites provide a web server and possibly a separate application server, depending on the web-based system architecture they use. The Vendor web server pages provide content that needs to be paid for and each access to these pages requires one or more e-coins from the customers' e-wallets in payment. In our architecture Vendor application server accesses the Broker application server to obtain touchstone information to verify the e-coins being spent and to redeem spent e-coins. They communicate with other vendor application servers to pass on e-coin indexes and touchstones. Vendors may use quite different architectures. In the example above, Vendor #1 uses a web server, custom application server and relational database. Vendor #2 uses a J2EE-based architecture with J2EE server providing Java Server Pages (web services) and Enterprise Java Beans (application server services), along with a relational database to hold vendor data. We use the open standard CORBA distributed objects to support broker and vendor interactions (Dai and Grundy, 2002).

Customer Interaction Examples

Initially a customer accesses the broker's web site to open an account and acquire a number of e-coins from the broker (bought using a single macro-payment). With a client-side e-wallet the broker sends an "e-wallet" that includes the e-coin ID and e-coins to the customer and the


Customer Registration - Microsoft
File Edit View Favorites Tools Help
Back
Address http://localhost:8080/src/broker/
(1)
(2)
Customer Buy Ecoins
E-wallet Application
Local intranet

Customer Registration

Name: leanne li
Password: ●●●●●●
ReEnter Password: ●●●●●●
Email Address: lileanne@hotmail.com
Credit Card#: 0987654321

Register Reset Cancel

&creditcard=09876543218

Customer Registration

E-wallet has been sent to your c

Your Customer ID is: 147
Please remenber your custom

Name: leanne li
Email Address: lileanne@hotmai

Download E-wallet from here.

Welcome!

The E-wallet has been downloaded on your computer.

You need to Run the E-wallet program by clicking the icon.

After Run the E-wallet Program,
you can go to Buy Ecoin page and buy some ecoins.

Account

You have got 0cs left!

Customer Buy Ecoins

Customer ID: 147
Password: ●●●●●●

Login Cancel

Customer Buy Ecoins

Customer ID:147

Amount of Ecoin: 50

OK Cancel

Customer Buy Ecoins

Ecoins have been sent to your E-wallet. Thanks!

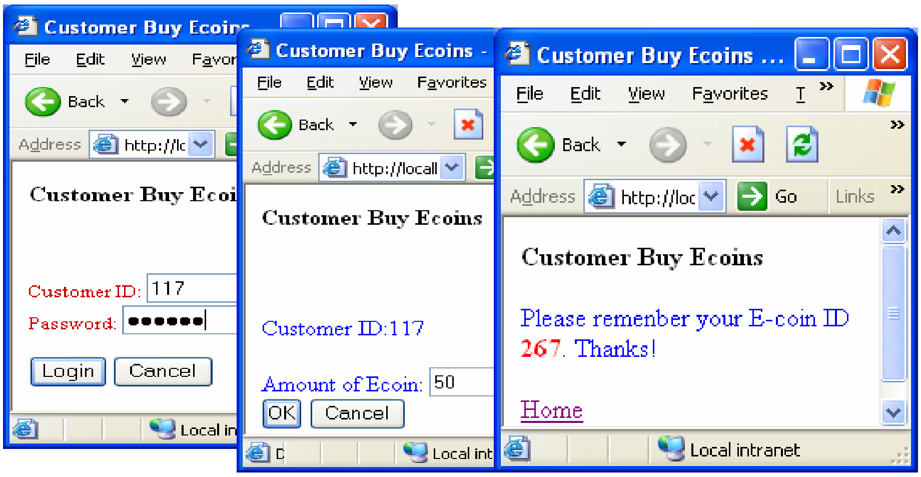
Home

Account

You have got 50cs left!

Figure 4. Interactions between customer and broker for client-side NetPay.

customer’s PC caches this information. The HTML interface for client-side NetPay used by customers to register and purchase e-coins is shown in *Figure 4*. The customer can register with the broker, download and run e-wallet application software (1). When needing to buy some e-coins, the customer authorises macro-payment by the broker who debits the customer’s supplied credit card to pay for the coins (2).



Customer Buy Ecoins

Customer ID: 117
Password: ●●●●●●

Login Cancel

Customer Buy Ecoins

Customer ID:117

Amount of Ecoin: 50

OK Cancel

Customer Buy Ecoins

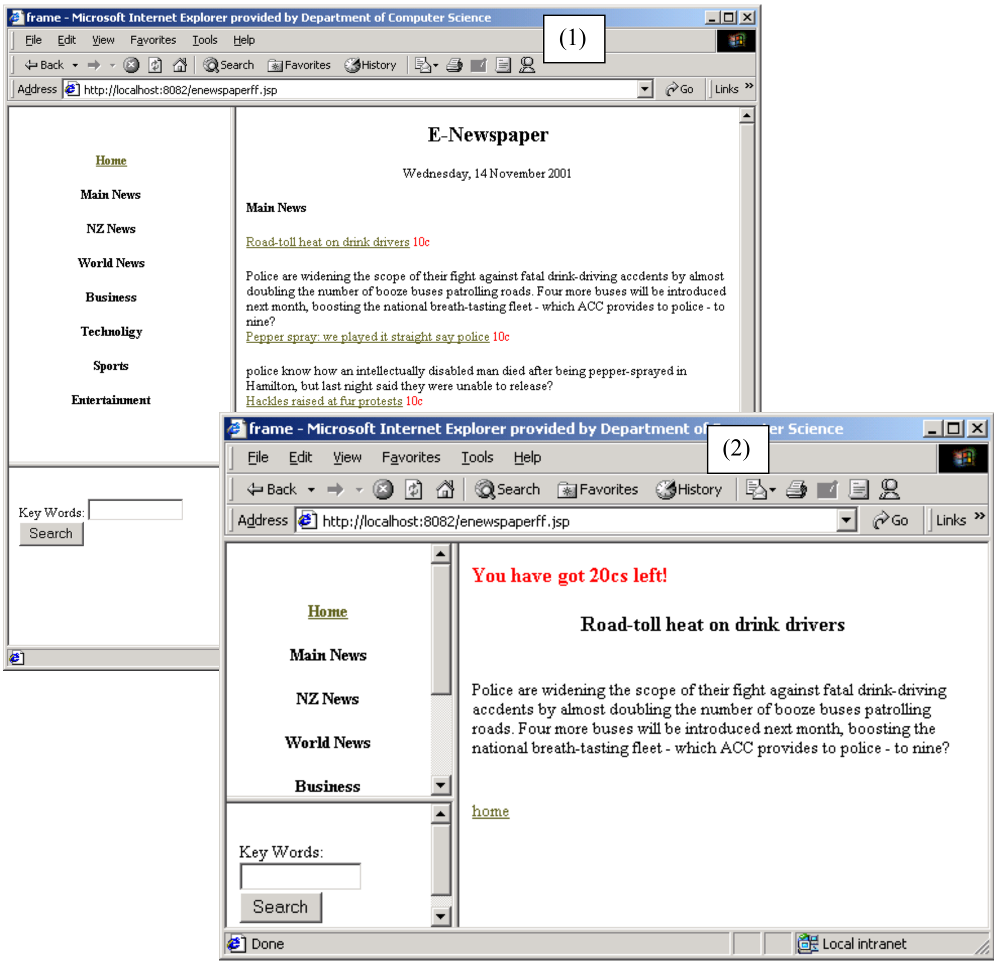
Please remember your E-coin ID 267. Thanks!

Home

Figure 5. Interactions between customer and broker for server-side NetPay.

The HTML interface for server-side NetPay used by customers to register with the broker is same as client-side NetPay, but it does not need to download and run e-wallet software. Customers purchase e-coins is shown in Figure 5. The differences between client-side and server-side are that there is no need for customer to download the e-wallet but the customer needs to remember the e-coinID and the customer needs to login and input their e-coinID and password when accessing a newspaper site for server-side NetPay E-wallets.

Unlike regular, subscription-based newspaper sites a micro-payment-based newspaper not only provides searching, browsing and newspaper content for customers, but also indicates article cost, as shown in *Figure 6* (1). When wishing to read the details of an article, a customer clicks on the article heading. The newspaper site debits the customer's e-coins (e.g.10c) provided by the customer's e-wallet and verifies if these e-coins are valid by using of a "touchstone" obtained once only from the broker. If the payment is valid (coin is verified and sufficient credit remains), the article is displayed on the screen. The customer may browse other articles, his or her coins being debited (the index of spent coins incremented) each time an article is read.



frame - Microsoft Internet Explorer provided by Department of Computer Science

Address http://localhost:8082/enewspaperff.jsp

E-Newspaper

Wednesday, 14 November 2001

Main News

Road-toll heat on drink drivers 10c

Police are widening the scope of their fight against fatal drink-driving accdents by almost doubling the number of booze buses patrolling roads. Four more buses will be introduced next month, boosting the national breath-tasting fleet - which ACC provides to police - to nine?
Pepper spray: we played it straight say police 10c

police know how an intellectually disabled man died after being pepper-sprayed in Hamilton, but last night said they were unable to release?
Hackles raised at fur protests 10c

Key Words:

frame - Microsoft Internet Explorer provided by Department of Computer Science

Address http://localhost:8082/enewspaperff.jsp

You have got 20cs left!

Road-toll heat on drink drivers

Police are widening the scope of their fight against fatal drink-driving accdents by almost doubling the number of booze buses patrolling roads. Four more buses will be introduced next month, boosting the national breath-tasting fleet - which ACC provides to police - to nine?

home

Key Words:

Figure 6. Customer spending E-coins at an E-newspaper site.

If a customer's E-coins run out, the customer is directed to the broker's site to buy more. When the customer changes to another online newspaper (or other kind of vendor using the same e-coin broker currency), the new vendor site first requests the current e-coin touchstone information from previous vendor's site. The new vendor contacts the previous vendor to get the e-coin touchstone and "spent coin" index and then debits coins for further news articles. At the end of each day, the vendors all send the e-coins to the broker redeeming them for real money (done by macro-payment bank transfer from the broker to vendor accounts).

EXPERIMENT DESIGN

As discussed in research method section, we decided to carry out three kinds of evaluation of our NetPay-enabled e-newspaper prototypes to determine their suitability for providing E-tailing system payment support:

- A usability evaluation surveyed users of the prototype to assess their impressions of the approach when carrying out information purchasing tasks using a micro-payment vs a macro-payment protocol.
- A performance evaluation assessed the performance of NetPay-enabled web sites to determine the overhead of the micro-payment extensions made to the software, particularly in regard to user response time.
- A qualitative evaluation assessed factors such as customer effort in using a NetPay-enhanced web site from a customer's perspective, along with the cost/benefit of the system for customers, vendors and brokers.

In this section we summarise the design of each of these experiments, and in the following section report on the results of each experiment and draw conclusions from these as to the utility of a NetPay-based micro-payment system for an E-newspaper E-tailing application domain.

Usability Evaluation

We evaluated participants' user satisfaction, navigational efficiency, effectiveness and general preference for the three payment systems – subscription-based macro-payment, server-side and client-side NetPay micro-payment with two newspaper sites (Preece, 1993; Rubin, 1994). These measures are the standard ones for determining how "usable" an interactive system is, and allow us to make judgements on the suitability of the interface for the tasks being carried out (Neilsen, 1992). Efficiency was measured by the degree of ease to change different newspaper sites and the speed of article content loading. Effectiveness was measured by assessing operations needed by the customer to complete their purchases. Satisfaction was a subjective measure assigned by each participant in the experiment.

We identified 10 participants to carry out a set of information purchasing tasks from our three E-newspaper prototypes, one using subscription-based macro-payment; one using a client-side NetPay e-wallet and one using a server-side NetPay e-wallet. After completing these information searching and access tasks with each of these payment systems, participants answered a post-test questionnaire. We choose to use a task list and post-survey questionnaire rather than other usability evaluation approaches do to the ease of setting up the experiment and also because we felt this would provide the best usability measures for such a problem domain (Preece, 1993).

After completing the tasks with all three payment systems, participants ranked the systems in order of preference. The application servers used are: newspaper1 and newspaper2 providing subscription-based macro-payment; broker, newspaper1, and newspaper2 providing server-side NetPay micro-payment; and broker, newspaper1, newspaper2 providing client-side macro-payment. These application servers were deployed for this experiment on the some host on the Windows XP network. The participants in the experiment used other PCs connected this network to carry out a set of tasks including registering, subscribing, buying coins, reading articles and reading articles over multiple sittings.

Performance Impact/Evaluation

Our three prototypes providing subscription-based payment, server-side and client-side NetPay micro-payment have been tested for application server performance and client response time. The key aim was to test how long a newspaper site takes to serve client requests when extended to use each of the three payment systems, from the time the customer clicks the title of an article to the time the article is fully displayed on screen. In order to do this we developed a pseudo-web browser to perform large numbers of requests to the web server and to time the response time of the web server. The macro-payment subscription-based approach makes one expensive macro-payment debit for pay for the initial subscription and then simply checks the whether a customer, after login, has a valid subscription. The micro-payment systems need to carry out an E-coin debit of the customer's e-wallet with each purchase of an article. We measured the CPU time taken by the vendor's web server and the overall time taken to action the page display. The longer the delay to display a page, the more problematic for the customer in terms of vendor information response time. The more CPU consumed by the server-side, the less overall client requests and lower response time overall can be supported.

Qualitative Comparison to Macro-payment

We carried out a third, qualitative assessment of our three prototype E-newspaper web sites to assess various factors associated with their costs and benefits for customers, vendors and the broker organization. The assessment criteria included:

- Number of customer interactions with the web site(s) needed to read articles

- Information retention needed by customers to use web site(s)
- Cost to customers depending on subscription and article pricing and article usage
- Cost to vendors of subscription authorization and e-coin redemption
- Cost to brokers of providing e-coins and redeeming coins with banks

The results for these analyses were obtained from analyzing the performance of each payment method in order to satisfy a payment scenario.

EXPERIMENTAL RESULTS

Usability Evaluation

Ten participants volunteered for our usability study of NetPay. They were an equal mix of non-IT specialists and graduate students, the later who were frequent users of on-line information portals. All participants were familiar with using E-tailing web sites, particularly for purchasing books, CDs and clothing. Participants were asked to complete five tasks with each system.

- Subscribe with the newspaper site or register and buy e-coins with a broker
- Read 3 articles on newspaper1 site
- Change to newspaper2 site and read 3 articles
- If subscription expired or e-coins run out, the user must renew it
- Read articles on the two vendor sites a second time, subsequently to the first use of the system

In the post-test questionnaire, we used a 5-point rating scale (1= Strongly Disagree, 5=Strongly Agree) to rate each tested characteristic. We also included open questions to gain user feedback to help in the qualitative analysis evaluation work. We presented the average ratings for the tested characteristics in a bar chart form as shown in *Figure 7*. The tested characteristics are:

a. *Ease of use***:** Payment system is ease to use.
b. *Efficiency1***:** It is easy to move around different newspaper sites.
c. *Efficiency2*: The speed of article content loading is fast enough.
d. *Efficiency3***:** It is easy to deal with subscription expired or e-coin run out.
e. *Preference***:** You are preferred to use the system widely.

We choose to use these assessment criteria as they test key requirements of micro-payment systems as indicated in our earlier work on micro-payment system requirements development (Dai et al, 2001), and in other micro-payment work (Hwang et al, 2001).

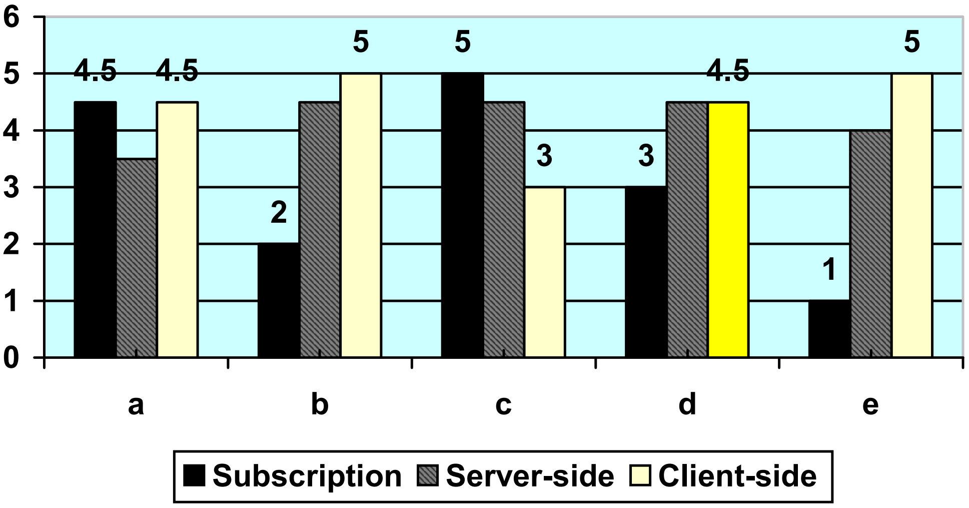


Figure 7. Three payment systems usability test results.

In this study, ease of use, efficiency, and satisfaction/preference results mainly favoured the client-side e-wallet NetPay system. However, this approach incurred an extra delay in page display due to communication from the vendor to the customer PC's e-wallet application, which the other systems don't have. Participants stated the article contents at different newspaper sites were easy to read without log in and the balance can be checked any time. The server-side NetPay system allowed users to read articles on different computers, but customers needed to remember e-coin IDs and had to log into the new newspaper site when change vendor. The article content loading was very fast on subscription-based system, but the users found that it is not as convenient to change vendor. The users generally needed to spend more money in order to subscribe to the whole newspaper provided by each site. Open question results revealed that client-side NetPay was found to be significantly preferred over a subscription-based system. In addition, server-side NetPay was more preferred than subscription-based system for this E-tailing application domain.

Performance Evaluation

We ran two sets of performance evaluations, one on our original NetPay prototypes and a comparable macro-payment subscription-based system and one on modified versions of the NetPay prototypes, optimized to provide lower e-coin management overhead. This was done due to the large database overhead the original prototypes incurred for debiting e-coins.

The results of the first set of performance evaluations are shown in *Table 1*. The response time measures how long it takes for a page to be returned from the vendor site. The server CPU time measures the time spent in the vendor's server debiting NetPay e-coins.

System	Response Delay Time (average)	Server NetPay CPU Time Usage (average)
Subscription-based	16ms	-
Server-side NetPay	80ms	64ms
Client-side NetPay	950ms	-

Table 1. Initial prototype performance

From *Table 1*, the server-side NetPay takes 80ms-16ms=64ms for e-coin debiting per article and Client-side takes 950ms-16ms=934ms total time, though the time to debit coins is taken by the client's e-wallet application, not the vendor's application server. The large overhead in the server for the server-side NetPay prototype is due to the database transactions it carries out to record coin updates and debits to redeem to the broker.

To reduce the e-coin debiting time, we created a transaction temporary file recording the data for redeeming instead of the redeeming database. Because of the application of such a temporary file, the e-coin debiting time decreases dramatically especially for server-side NetPay system. The results are shown in *Table* 2. At the end of each day, the system redeems the coins or updates the database, and then deletes the records in the transaction temporary file. From *Table 2*, Server-side NetPay takes 30ms-16ms=14ms for e-coin debiting per article and Client-side takes 900ms-16ms=884ms after the application of the temporary file. The impact of the NetPay micro-payments on the vendor application server are greatly reduced, but the client-side e-wallet still incurs considerable response time delay due to the additional vendor->customer PC communication with it.

System	Response Delay Time (average)	Server NetPay CPU Time Usage (average)
Subscription-based	16ms	-
Server-side NetPay	**30ms**	14ms
Client-side NetPay	**900ms**	-

Table 2: Prototype performance after using a temporary file

Qualitative Analysis

With our qualitative assessment we wanted to measure the different characteristics of macro-payment and micro-payment approaches, and to analyse the differences between our two NetPay e-wallet models. The results of this assessment are summarized in *Table 3*.

Summary of Results

In summary, a macro-payment approach is more beneficial for the customer if they typically read a large portion of the on-line newspaper articles, or if a comparable micro-payment approach has a high-cost per article for the user. However, the micro-payment approach wins out when the customer typically users a small portion of the articles, articles are low-priced and if the customer reads articles from multiple newspapers and can use their e-coins across any of these vendors. There is a performance cost for the vendor in providing a micro-payment approach in terms of time taken to track e-coin spends and redemption. However, there is also normally a high cost to the vendor of providing macro-payment support for subscription purchase.

One interesting issue is whether vendors would "buy in" to a micro-payment system approach. By using subscription-based macro-payments to access information, vendors can lock in customers i.e. achieve "brand capture" and discourage customers from moving to other information sources e.g. other E-newspapers as they have already made a significant financial commitment to one newspaper. Similarly, there needs to be sufficient vendors sharing the same micro-payment system and "currency" to allow useful movement by customers from vendor to vendor. In addition, the software maintenance overhead of installing a micro-payment system must be considered by vendors. One approach is to adopt a portal-based approach to accessing multiple vendors through a single micro-payment enabled portal which does the debiting and redeeming of spending on behalf of multiple vendors.

Other alternatives do exist to providing macro- and micro-payment models as described in this paper for E-tailing systems. The most common is the provision of services to users for free but the use of advertising embedded within pages or as pop-up windows. However, studies have suggested that vendors would prefer a payment mechanism which is on a per-usage basis, either subscription-based or micro-payment-based to provide greater reliability of income. A key outstanding challenge with micro-payment systems is being able to spend currency (e-coins) at a wide range of vendors – if the customer must purchase different "currencies" for different groups of vendors then this will be both inefficient in terms of expenditure and incur overheads of the customer memorizing different usernames, e-coin IDs and passwords. One approach is to support inter-broker micro-payment e-coin exchange transparently when the customer visits a vendor that uses a different broker's currency.

Criteria	Macro-payment	Server-side Wallet	Client-side Wallet
Customer interactions	1.Subscribe (customer and credit card details) 2.Login to web site 3.Article read 4.Subscribe if move to another vendor +after subscribe/login simply read articles -must supply personal details -must subscribe for each vendor	1.Purchase e-coins from broker 2.Login to web-site 3.Article read 4.Login to new vendor +after login simply read articles +only login to new vendor -must recall e-coin ID, password -must supply for each vendor	1.Download wallet software 2.Purchase e-coins 3.Article read +don't need login/password for any vendor +simply read articles -must download, install and have running client-side e-wallet software
Information retention	Need to remember username/password. May avoid if use same PC always (can store in browser cookies)	Broker username and password needed to purchase coins. E-coin ID and password needed to access server-side wallet information	Broker username and password needed to purchase coins.
Subscription cost Low e.g. $10 High e.g. $50	Low – if use moderate number of articles, more cost-effective for customer. High – need to use substantial amount of articles for benefit. No cost savings for customers if use multiple vendor sites.	N/A	N/A
Article cost Low e.g. 2c High e.g. 10c	N/A	Low – customer likely to read more. Vendor needs more read to cover costs. High – customer likely to read less. Cost savings to customers. Can price articles differently.	Low – customer likely to read more. Vendor needs more read to cover costs. High – customer likely to read less. Cost savings to customers. Can price articles differently.
Article requests by customers Low e.g. <10 High e.g. >20	Low - No cost benefit for vendor High - Large number by many customers effects system performance	Low – if low cost, vendor makes little profit. High – if high cost to customer, may be more costly than macro-payment approach High numbers impact overall vendor server performance.	Low – if low cost, vendor makes little profit. High – if high cost to customer, may be more costly than macro-payment approach. Has large performance impact (in current implementation). High numbers impact overall response time of vendor server.
Vendor Benefit	"Brand capture" of customers due to use of subscription to each vendor site.	If large enough vendor community can encourage movement, partnerships. Customers need to login to access wallets.	If large enough vendor community can encourage movement, partnerships. No login for wallet access needed but need wallet installed on customer PCs.
Vendor cost	Need to buy macro-payment supporting software and pay bank for facility. Need to price subscription to adequately cover costs.	Need to allow broker to take portion of overall customer payments OR broker takes costs from customer direct. Need to price articles so cost per article/ and number of articles used cover costs. The performance overhead on the vendor server is significant.	Need to allow broker to take portion of overall customer payments OR broker takes costs from customer direct. Need to price articles so cost per article/ and number of articles used cover costs. There is little performance overhead on vendor server but response time reduction for customer.
Broker cost	N/A	May charge vendors for each e-wallet request or portion of redeemed coin amount. May charge customer for each e-coin purchase. Possibility of high number of e-coin requests from vendors.	May charge vendors for each e-wallet request or portion of redeemed coin amount. May charge customer for each e-coin purchase. Low overall e-coin requests as client-side wallet brokers these.

Table 3: Qualitative Assessment Summary

SUMMARY

We have developed a prototype architecture to support an efficient, secure and anonymous micro-payment system for high-volume, low-cost on-line E-tailing systems. This incorporates a broker used to generate, verify and redeem e-coins, a customer e-wallet stored either client or server-side, and vendor application server components. Our NetPay architecture provides for both secure and high transaction volume per item by using fast hashing functions to validate e-coin unspent indexes. NetPay is an off-line protocol allowing the vendors to interact only with customers after initial coin validation. We have assessed two variants of our micro-payment system deployed with an on-line newspaper web site and compared these with a macro-payment, subscription-based variant of the web site. These evaluations have indicated that for users the micro-payment approach has some appeal over traditional macro-payment approaches and that for some usage patterns the micro-payment approach is far more efficient in terms of cost to the customer. We are continuing to enhance our NetPay system implementation and are exploring its usage for other E-commerce systems with a high volume/low cost transaction mix. These possible application domains for micro-payment include pay-per-search specialized on-line databases, pay-per-item on-line music, magazine and book sites, and pay-per-article on-line sports and news headline sites.

REFERENCES

Blankenhorn, D. (2001). Charging for Content, E-commerce Times. http://www.ecommercetimes.com/perl/story/306.html.

Dai, X. and Lo, B. (1999). NetPay – An Efficient Protocol for Micropayments on the WWW. *Fifth Australian World Wide Web Conference*, Australia.

Dai, X., Grundy, J. and Lo, B. (2001). Comparing and contrasting micro-payment models for E-commerce systems. *International Conferences of Info-tech and Info-net (ICII)*, China.

Dai, X. and Grundy, J. (2002) Architecture of a Micro-Payment System for Thin-Client Web Applications, *In Proceedings of the 2002 International Conference on Internet Computing*, Las Vegas, CSREA Press, June 24-27, pp. 444-450.

Domingo-Ferrer, J., Herrera-Joancomarti, J. (1999) Spending programs: a tool for flexible micropayments. *Information Security. Second International Workshop*, ISW'99. Lecture Notes in Computer Science, Vol.1729. Springer-Verlag, Berlin, Germany, 1-13

Furche, A. and Wrightson, G. (1996) SubScrip – An efficient protocol for pay-per-view payments on the Internet", *The 5th Annual International Conference on Computer Communications and Networks*, USA.

Herzberg, A. and Yochai, H. (1996). Mini-pay: Charging per Click on the Web. http://www.ibm.net.il/ibm_il/int-lab/mpay

Herzberg, A. (1998). Safeguarding Digital Library Contents - Charging for Online Content. D-Lib *Magazine*, January*, ISSN 1082-9873.*

Hwang, M-S., Lin, I-C. and Li, L-H. (2001). A simple micro-payment scheme. *Journal of Systems & Software*, vol.55, no.3, Jan. 2001, Elsevier, pp.221-9.

Manasse, M. (1995). The Millicent Protocols for Electronic Commerce. *First USENIX Workshop on Electronic Commerce*, New York.

Neilsen, J. (1992). The Usability Engineering Lifecycle. *COMPUTER* 25 (3), March 1992, pp. 12-24.

Preece, J. (1993). A Guide to Usability: human factors in computing. Addison-Wesley.

Rivest, R. and Shamir, A. (1997) PayWord and MicroMint: Two Simple Micropayment Schemes. *Proceedings of 1996 International Workshop on Security Protocols*, Lecture Notes in Computer Science v. 1189, page 69-87. Springer.

Rubin, J. (1994). Handbook of Usability Testing: how to plan, design, and conduct effective tests, New York: Wiley.

Stern, J., Vaudenay, S. (1997) SVP: a Flexible Micropayment Scheme. *Financial Crypto '97*, Springer-Verlag, 161-171.