Homework 3

This assignment must be submitted **electronically** using the Yale Classes server by **5pm EST, Thursday, March 6, 2003**. It covers readings and lectures through Tuesday, March 4, 2003. Late homeworks will not be accepted.

Please be sure to type your name and e-mail address at the top of your homework assignment. In addition, please include your NetID and "hw3" in the name of your file, *e.g.*, **`vr48-hw3.doc`**. Detailed instructions for submitting your homework are available on the course webpage at http://zoo.cs.yale.edu/classes/cs155/spr03/submit.html.

1. eBay AND ITS LEGAL RESPONSIBILITIES (25 points)

The following parts of the eBay User Agreement were discussed in class:

> [Section 3.1] "We are not involved in the actual transaction between buyers and sellers. As a result, we have no control over the quality, safety or legality of the items advertised, the truth or accuracy of the listings, the ability of sellers to sell items or the ability of buyers to buy items. We cannot ensure that a buyer or seller will actually complete a transaction."
>
> [Section 3.3] "… because user verification on the Internet is difficult, eBay cannot and does not confirm each user's purported identity. Thus, we have established a user-initiated feedback system to help you evaluate with whom you are dealing…"

Do you think that eBay should be legally responsible for illegal acts by eBay users that it profits from? Examples of illegal acts include sales of counterfeit items or stolen items. Justify your answer on technical, legal or moral grounds using at most one page. The number of points that you receive for your answer will depend on how well you justify your position.

2. PORTALS AND PRIVACY (35 points)

In Chapter 7 of **Trust and Risk in Internet Commerce**, Camp presents (and refutes) the business strategy put forth by Hagel and Armstrong of *NetGain*. One point she makes is that a business that focuses intensely on collection of customer information, rather than delivery of customer service, is positioning itself as an intermediary between its users and other businesses; by definition, relatively few service providers will succeed as mass-market intermediaries, and so the *NetGain* focus on customer surveillance cannot be a very generally applicable business strategy. Camp then goes on to discuss the objectionable privacy implications of customer surveillance as a centerpiece of Internet business.

The few Internet businesses that *have* flourished as intermediaries include *portals*; the example discussed in class was Yahoo! Recall that the Yahoo! slogan is "The only place anyone needs to go to find anything, communicate with anyone, or buy anything." Recall further that the Yahoo! business model (and, indeed, the concept of "portal") has evolved considerably in parallel with the evolution of the Internet as a mass-market communications medium.

This homework question asks you to contemplate the following proposition: In the current Internet environment, user privacy is a growing concern. Therefore, a portal company could successfully differentiate itself from its competitors by committing to aggressive protection of customer privacy.

A (15 points). For five points each, explain how such a portal company might make use of *or be thwarted by* three of the following technological components or ideas that we have covered in this class:

- Encryption
- Digital signatures
- P3P
- Cookies
- The HTTP protocol
- Digital-rights-management languages

B (20 points). Do you agree with the proposition that aggressive privacy protection would be a successful strategy for a portal company? Why or why not? Limit your answer to at most one page. As usual, the number of points that you get will depend on how well you justify your answer.

3. VENTURE CAPITAL (20 points)

Consider *temp-numbers.com*, a start-up company with first-round funding from one VC firm. In this first round, Jane (the founder) owns 100 shares priced at $10 each, and the VC owns 50 shares priced at $10 each.

A (4 points). The pre-money valuation of *temp-numbers.com* was $2 million. How much money did the first-round VC invest?

In a subsequent bridge round, Jane issues 10 new shares, priced at $2 each. Under the terms of the first round of funding, the first-round VC has "anti-dilution protection."

B (8 points). For four points each, compute both the new total number of shares and the new number of shares owned by the first-round VC if these numbers are computed using a weighted ratchet.

C (8 points). For four points each, compute both of these numbers using a full ratchet.

4. XML AND DOCUMENT EXCHANGE (20 points)

Document exchange is a natural way to think about doing business. Business interactions can often be represented by an exchange of XML documents. For example, ordering something from a manufacturer involves the following interaction:

> Buyer sends Manufacturer a *purchase order*.
> Manufacturer sends Buyer an *invoice* for the order.

In the above example, the *purchase order* and *invoice* are the documents exchanged, and the information contained in these documents represents the order requested and fulfilled. To automate this process, the computers that are handling the exchange of information must be able to *parse* the documents and extract the contents in a useful form.

XML allows content in a document to be *tagged*, identifying what information is being given. For example, the line “10/30/2001” may be tagged in a purchase order as the date the request was made. XML documents reference a *Document Type Definition*, or DTD, which gives the rules for what tags are allowed and what content may be contained in each tag. Using the same example, a DTD for the purchase order may contain a rule that every purchase order must contain a request date, specified by the tag `<date>`, and that the date is given by three numbers representing the month, day, and year of the request. Given a DTD, an *XML parser* can obtain and process information from the XML document in a useful way.

Here is an example business exchange:

> A customer calls the Acme Radio Company and says, “I would like a radio that gets both FM and AM stations, has an alarm, and costs less than $30. Can you send me a list of radios like that?”

This exchange can be automated by an exchange of XML documents. The *information request* and *catalog* are two documents in the exchange above. The request can be used to ask about products; the catalog can be used to list the products meeting the specs of the request. A DTD for one of these documents describes the type of information contained in it. The following is an example DTD for the information request:

```
<!DOCTYPE INFOREQUEST [
<!ELEMENT request (minprice?, maxprice?)>
<!ELEMENT minprice (#PCDATA)+>
<!ELEMENT maxprice (#PCDATA)+>

<!ATTLIST request

      band (FM | AM | both) ‘both’
      alarm (yes | no) #IMPLIED
      cd (yes | no) #IMPLIED
>

<!ATTLIST minprice currency CDATA #REQUIRED>
<!ATTLIST maxprice currency CDATA #REQUIRED>
]>
```

The <!DOCTYPE *name* […]> declaration gives the definition for the document type *name.* The brackets contain the elements (and their specifications) used in the document.

The <!ELEMENT *name* (*contents*)> declaration specifies a way to tag information with *name.* The *contents* field indicates what the element can contain (*i.e.*, what can legally appear within the tag—see discussion of `<date>` above). The contents can be other elements or a character string or both. Note the following conventions for the element declaration:

- `#PCDATA` means a character.
- + means one or more occurrences, so `(#PCDATA)+` means one or more characters, *i.e.*, a character string.
- * means zero or more occurrences.
- ? means optional. `minprice?` means that the element `minprice` may be present one or zero times.

Each element can also have an <!ATTLIST *name list…*> declaration to go along with it. This specifies the attributes assigned to every tag. Consider the element `minprice`, which is used to tag the lower end of the price range. The element contents may just be a number. A required attribute for this element is `currency`, the unit of the amount provided. (Without it, `<minprice>10</minprice>` could mean 10 dollars or 10 cents.) In XML, attributes are written inside the tag, *e.g.*, `<minprice currency=”dollars”>` … `</minprice>`.

Note the following conventions for the attribute-list declaration:

- You can define more than one attribute for an element; they are just listed one after the other. (In the above example, the attribute list for `request` has three attributes defined. The attributes are put on separate lines for readability.)
- `CDATA` means a character string.
- `#REQUIRED` means that the element tag must contain a value for the attribute. For example, a `minprice` tag without a currency attribute (*e.g.*, `<minprice>` as opposed to `<minprice currency=“dollars”>`) is not allowed.
- `#IMPLIED` means that the attribute is not required.
- (*value1* | *value2* | …) means that the attribute can be one of the values in the list, specified by *value1*, *value2*, etc. If there is a value after the parentheses, that is the default value assumed if the attribute is not given. In the above, a request tag with “`band=`…” omitted would be assumed to have “`band=both`” specified.

Given the above DTD, here is an example XML information-request document that follows it:

```
<request band=’both’ alarm=’yes’>
      <maxprice currency=’dollars’> 30 </maxprice>
</request>
```

Tagged content starts with <*name*> and ends with </*name*>.

For a more detailed explanation of XML and examples of XML documents and DTDs, follow the links on the course webpage to “Examples of Document Type Definition in XML,” by Sheng Zhong, Yale University, March 2001 (http://zoo.cs.yale.edu/classes/cs155/spr03/ex-xml.pdf).

Consider the following business exchange:

> A customer searches through the list of available flights from her hometown to a vacation spot. She contacts an airline company with a reservation request for a flight, including any preferences she might have. The company responds with a ticket confirming the details of the flight and the order.

A (6 points). For three points each, identify two XML document types in the above exchange.

B (8 points). Choose one of the types you identified in part (A), and write a DTD for that document type following the rules and the example given above. Include tags for information that you think should be present in the document. (This is open-ended. You can be as detailed as you want to be, but you should be somewhat realistic with your document definition.)

C (6 points). Give a sample XML document for the DTD you wrote in part (B).