cis3.2 — electronic commerce — 2 feb 2006 — lecture # 3

today

topics:

- internet facilities
- issues in e-commerce development
- open systems
- network concepts

internet facilities

- the world wide web
  - HTML = hypertext markup language
  - *hyperlink*
  - *browser*
  - *web page*, *web site*, *web server*
- ftp (file transfer protocol)
  - *download*
  - *upload*
- email
- newsgroups
  - *posting*
  - *thread*
- mailing lists

distributed systems

- architecture of an online system:
  - *user* or *client* uses a *browser*
  - *server*
  - *database*
- remember our case study (music store) — how would we define the requirements for the above components? how would they be connected?

issues in e-commerce development

- legacy technology
  - not enough space
  - no "state"
  - static web pages
- security and privacy
- programming and abstraction — portability, compatibility
- speed of development
- structure and data
- distributed transactions
- design
- quality of posted material
- speed of servers

open systems

- "a system whose architecture is not a secret"
- prime example: UNIX, LINUX — runs on many hardware platforms; LINUX is free
- *protocol*
  - rules for how computers communicate with each other
  - TCP: transmission control protocol (computer ↔ computer)
  - UDP: user datagram protocol
  - IP: internet protocol
  - HTTP: hypertext transfer protocol (computer ↔ browser)
- *distributed system*
  - multiple computers are distributed geographically
  - example: banking system

network concepts

- network topologies
  - bus (linear model; inexpensive to run cables, but not robust to node failure)
  - ring (example: IBM token ring)
  - star (book calls it "hub"; can be expensive to run cables, but robust to node failure)
- layered models: OSI (below) is the classic model

Open System Interconnection (OSI) reference model

- also called the 7-layer model:
  1. application layer (displays data, communicates with lower layers via presentation layer)
  2. presentation layer (link between app and lower layers; converts application layer data to forms understandable by other layers, and back; translates the "meaning" of the bits)
  3. session layer (exchange of data between applications – "dialog" – and synchronization between applications)
  4. transport layer (transfer of data through network; effects flow control; provides some error recovery)
  5. network layer (physical routing of data from one computer to another; facilitates sender finding receiver)
  6. data link layer (manages transmissions of low-level data; detects and corrects transmission errors)
  7. physical layer (sends electronic signals, or "bits" – 0's and 1's) (usually linked to above)
- Here's a phrase to remember the layers: "**A**ll **p**eople **s**eem **t**o **n**eed **d**ata **p**rocessing." where the first letter of each word refers to a layer (above).

- information formats
  - "information units" are passed from one layer to another; "headers" are added as information passes from upper to lower layer
  - terminology is defined below (it is often not used clearly or precisely):
    - *frame* = information unit whose source and destination are the *data link* layer
    - *packet* = information unit whose source and destination are the *network* layer
    - *segment* = information unit whose source and destination are the *transport* layer
    - *message* = information unit whose source and destination are the *application*, *presentation* or *session* layers
    - *datagram* = information unit in a "connectionless" network (see below)

connection-oriented vs connectionless networks

- "connection-oriented"
  - like a telephone
  - first a connection must be established, then data is transferred, finally the connection is closed
  - e.g., TCP
  - monitors for lost packets and re-sends if necessary
  - more overhead than connectionless, but more reliable
- "connectionless"
  - like sending a letter in the mail:
  - there is no guarantee that the recipient ever gets the letter;
  - e.g., UDP
  - less overhead than connection-oriented, and less reliable