Wednesday, October 20, 2004
Meeting Log

Purpose: Protocol / Collision Handling Meeting
Time: October 20, 2004 Wednesday, 8:00 PM
Location: Sun Lab
Secretary: Jason Segal
Present: Brandi, Jason, Matt, Steven

Items Discussed:

Change of Meeting Agenda

- Jonas could not be present for the protocol meeting, so we continued conducted a limited discussion of protocol options, then considered collision detection and change ordering arrangements.

Preliminary Protocol Discussion

- We identified the primary questions that need to be answered regarding the protocol extension/use:
    1. Where can our information be placed without causing the server to reject the message? (Where should our branch of the XML be inserted?)
        a. Must it be sent as a text message? This seems like it would almost certainly be possible, but could cause difficulty in the future, and should therefore be avoided if possible.
        b. Can it be sent as a new branch of the XML tree? (A branch such as "<Inkboard> *Information* <Inkboard>", inserted high in the tree)
    2. Where should our branch be placed in order to avoid conflicting with current uses of the protocol? (Where do other applications place theirs?)

Message Ordering and Collision Detection

- Two arrangements were discussed: a random back-off system, and a nominal-host system

- Random Back-off: Every user keeps a copy of the shared document in its current state. Each time a change to the document is sent by any user, it is given an identifying number, which is incremented each time a change is made. After a user sends a change message, a fixed time is given in for detecting conflicts with the message. If the user receives a new message with the same ID number, the system will recognize a collision. If a collision is detected, all users who involved in the collision (those who both sent and received a message involved in the collision) will send an error message to all other users, informing them of the collision state. All instances of the application involved in the collision will then engage in "random back-off" behavior, each waiting a random time to see if they can receive the conflicting message before sending their own. When a new user begins to share the document, his or her instance of Inkscape would

accept the current version of the document from the first responder, saving all change messages until the document is ready.

- Nominal-Host: Under a Nominal-Host system, each user is assigned a "host order number" upon beginning to share a document. One user's application is currently the "nominal host", and decides the order in which changes will be made to the document based on the order in which it receives messages. All change messages are sent to all instances of Inkscape sharing the document, and each message is given an ID that is unique when considered in combination with the sender (ex: Message 3 from User 5 can be differentiated from Message 3 from User 2). No instance sharing the document implements any change until instructed to do so by the Nominal-Host. After a fixed period of time or number of messages, whichever comes first, the host sends an "order message" that identifies the messages the host has received and the order in which they are to be implemented. The non-host instances, then implement the changes in that order. If the host ceases sharing the document, then the instance with the lowest "host order number" becomes the new Nominal-Host. New users beginning to share the document would receive the document from the Nominal-Host (or would be directed to request it from another sharing instance by the host).

- Upon review of both potential systems, several things were determined. First, the Random Back-Off system would be much easier to implement, if no other reason to favor one over the other can be found. Second, the Random Back-Off system has better "best-case" efficiency (1 message to all users except sender per change, if no collisions occur, while the nominal-host system must periodically send order messages no matter what). Third, the Nominal-Host system has a better "worst-case" efficiency (One message from each user to each other user, then an extra from the host to each other user, vs. the potential for a spiraling collision-backoff-expanded collision chain that could have one collision sequence per user before being sorted out). Fourth, the Random Back-Off system would see "collisions" for any two messages sent at the same time, not just any two messages that give conflicting information, while the Nominal-Host system effectively prevents any collisions from occurring by enforcing an arbitrary order. (This problem with the Backoff technique might be corrected by analyzing the contents of messages, but such a step would not be necessary for the host system.)

- After comparing the potential benefits of the systems, it was decided that the Random Back-Off system would be our first choice for implementation.

Compilation of Inkscape

- Steve has successfully compiled a version of Inkscape
- No one has yet succeeded in compiling Inkscape under Microsoft Visual Studio. It was suggested that just using MSVS for editing and compiling with a different tool might be a better approach.

Tasks assigned:

Complete Document Revisions by 10:00 PM on Friday (10/22)

- Complete your review and alterations of your assigned documents and send them to the team

Next Meeting / Scheduling:

Monday at 5:20 PM (Dr. Azhar's office) – Protocol discussion will continue afterward (time limit: 1 hours)