CS351 Auction House Project - Auction House Message System Outline

Andrew Geyko

This document serves to outline the how the Auction House utilizes the messaging system in order to provide a (hopefully) acessible format for creating components that interact with it. I believe that it would be easier for the bank to specify it's own standard of how it receives messages since it seems that the AuctionHouse/Agent are requesting things for the bank to do, and specifying something like that here would just put unneccessary constraints on the bank code.

This document is a work in progress, please let me know if some change in this format would make it easier to write your code, we can work on changing it

Initial Connection

An initial connection is established when a socket is first opened. When the AuctionHouse receives the following message, it will send a response aknowledging that the connection was successful. If the response is not received, something has probably broke.

Message (Agent→AH) Format

Origin: Origin.AGENT

Type: Type.ESTABLISH_CONNECTION

Body: Will probably need to specify the bank account number here, not sure yet

Info: N/A

Message (AH→Agent) Format

Origin: Origin.AUCTIONHOUSE

Type: Some kind of Acknowledgement enum (not implemented yet)

Body: N/A

Info: N/A

Main Logic

Getting Items

The get items message will be used by the agent to get a list of the items being auctioned along with their ID numbers (which will be used to specify which item to bid on).

Message (Agents→AH) Format

Origin: Origin.AGENT

Type: Type.GET_ITEMS

Body: N/A

Info: N/A

Message (AH→Agent) Format

Origin: Origin.AUCTIONHOUSE

Type: Probably some kind of send_items enum, not implemented yet

Body: N/A

Info: I don't have a complete idea for this yet, but here are the three main ones I've thought of:

(a) Just send the Auction object (probably a bad idea for encapsulation reasons)

(b) Send a string with the data needed (probably a bad idea because it would be annoying to parse)

(c) Create a new "AuctionInfo" class to hold the information agent would need, and send that (probably the best idea)

Bidding

The bid message will be used to make a bid (clearly). It will be necessary to specify the amount that the agent is bidding, and the ID of the auction the agent wants to bid on. In response, the agent will receive a message stating whether the bid was accepted or denied.

Message (Agents→AH) Format

Origin: Origin.AGENT

Type: Type.MAKEBID

Body: This will be two separate lines (separated by newline character). The first line will specify an integer amount for the amount to bet. The second line will be an integer id of the auction to bet on (this id is sent in the "get items" message)

Info: N/A

Message (AH→Agent) Format

Origin: Origin.AUCTIONHOUSE

Type: BIDSUCCESS — BIDFAILED (self-explanatory)

Body: Maybe a string outlining the reason for the failure, but maybe not necessary.

Info: N/A

Winning a Bid

In this case, the agent does not send a message, but just gets one whenever it wins. No response is necessary (yet), so the agent just needs to process this on its own.

Message (AH→Agent) Format

Origin: Origin.AUCTIONHOUSE

Type: probably an enum like "bid_won", not implemented yet

Body: N/A

Info: Will probably be an "AuctionInfo" object as described in the getting items message, still need to work out a format for that.

Outbid by other agent

I haven't though about or implemented too much of this feature, but will update in future.

Closing Connection

I haven't though about this one too much either, will update in future.