Model-Based Specification of Complex Bidding Strategies in Agent-Based Online Auctions

Prof. Haiping Xu
Computer and Information Science Department
University of Massachusetts Dartmouth
Email: hxu@umassd.edu

Acknowledgement

- Graduate Students:
  - Benjamin J. Ford
  - Christopher K. Bates
- Collaborators:
  - Sol M. Shatz, Professor, UIC
  - Fei Dong, Ph.D. Candidate, UIC
- Sponsor:
  - National Science Foundation
Online Auctions

- Different types of auctions
  - Increase-price auction (English auction)
  - Decrease-price auction (Dutch auction)
  - Second-price sealed-bid auction (Vickrey auction)
- English auction has become the most popular one in online auction houses (e.g., eBay).
- However, it is time-consuming for a human user to search and place bids on an auctioned item.
- There is a pressing need to introduce agent technology into online auction systems.

Agent-Based Online Auctions

- It consists of an auction house and a number of clients.
- It is designed as a multi-agent system.
- The auction house is managed by auction house administrator.
- Agents at the client side work on behalf of human users.

*Auction House* (Server)

- Auction Agent
- Auction Agent
- Auction Agent
- Bidding Agent
- Bidding Agent
- Bidding Agent

*Bidding agents* can place bids on behalf of human users, and they are typically running on different machines.
Specification of Bidding Strategies

- Provide an interface for human users to specify bidding strategies
  - Complex and flexible bidding strategies
  - Easy-to-use interface
- Develop a formal model for bidding strategy specification
  - Separation of concerns: a layered approach
  - Adopt a visual modeling language, e.g., UML activity diagram
  - Convert it into a formal rule-based bidding model
- Advantages of allowing customized bidding strategies
  - Simulate human bidding behaviors more precisely
  - Support generation of real-time auction data for analysis and testing purpose

Bidding Agent Architecture

- Layered Bidding Strategy Model (LBSM)
  - specify
- Rule-based Bidding Strategy Model (RBSM)
  - decision making
  - converted into
- Bidding Agent Interface
  - next action
- Reasoning Module
Layered Bidding Strategy Model (LBSM)

Examples of Bidding Actions
- DynamicBidAction
- ChangeDynamicBidIncrement
- RandomPauseBidding(2700~3600)

Simple Strategy and Complex Strategy

Simple Strategy S1 (aggressive)
- ChangeDynamicBidIncrement(10)
- DynamicBidAction()
- RandomPauseBidding(2700~3600)
    - highestBidder && highestBid + 10 <= bidLimit
    - else

Complex Strategy C2 (aggressive/cautious)
- numOfActiveBidders > 5
- Simple Strategy S3
    - else

Simple Strategy S3 (cautious)
- numOfActiveBidders <= 5
- Simple Strategy S3
    - else
Rule-Based Bidding Strategy Model

<production rule>::= <strategy rule> | <action rule> | <initial strategy rule> | <initial action rule>
<strategy rule>::= <s-domain> <bidding strategy> <condition> -> <bidding strategy>
<s-domain>::= <s-domain>.<complex strategy> | <complex strategy>
<bidding strategy>::= <simple strategy> | <complex strategy>
<condition>::= <compound condition> | <arithmetic condition> | <comparison condition> | <boolean condition>
'action rule>::= <a-domain> <action> <condition> -> <action>
<a-domain>::= <s-domain>.<simple strategy> | <simple strategy>
<action>::= <basic bid> | <change bid limit> | <change dynamic bid increment> | <dynamic bid> | ... | <pause> | <stop>
<initial strategy rule>::= <complex strategy> -> <initial strategy>
<initial action rule>::= <simple strategy> -> <initial action>

Definition of bidding strategy language (BSL) in Backus-Naur Form (BNF)

Algorithm: Model Conversion

function convertToRuleBasedStrategyModel (LBSM lbsm)
    if lbsm is a complex strategy
        add a new initial strategy rule:
        lbsm → lbsm.initialStrategy
        for each StrategyTransition at in lbsm
            set up s-domain according to the strategy hierarchy
            add a new strategy rule: s-domain, st.startStrategy, st.condition → st.endStrategy
        end
    for each strategy s in lbsm
        convertToRuleBasedStrategyModel (s)
    end
    else if lbsm is a simple strategy /* base case */
        add a new initial action rule: lbsm → lbsm.initialAction
        for each ActionTransition at in lbsm
            set up a-domain according to the strategy hierarchy
            add a new action rule: a-domain, at.startAction, at.condition → at.endAction
        end
    end
end function
Algorithm: Reasoning Engine

```plaintext
function findNextAction (Domain domain, Action currentAction)
    if currentAction == null
        if domain is a ComplexStrategy
            Search for initial strategy rule isr for domain that leads to initial strategy
            return findNextAction (isr, null)
        else if domain is a SimpleStrategy
            Search for initial action rule iar for domain that leads to initial action ia
            return ia
        else if currentAction != null
            Remove and process the first element fe of domain, and let the remaining domain be r-domain
            if fe is a ComplexStrategy
                /* strategy transition */
                Retrieve all strategy rules for the first element of r-domain and store them in a list
                while the list is not empty
                    Remove and process strategy rule sr at list head
                    if the condition for sr is true
                        Let s be the conclusion part of sr
                        return findNextAction (s, null)
                    else if sr is a SimpleStrategy
                        /* action transition */
                        Retrieve all action rules for the currentAction and store them in a list
                        while the list is not empty
                            Remove and process action rule ar at list head
                            if the condition for ar is true
                                return the conclusion part of ar
                            return currentAction
                return findNextAction (r-domain, currentAction)
            else if fe is a SimpleStrategy
                /* action transition */
                Retrieve all action rules for the currentAction and store them in a list
                while the list is not empty
                    Remove and process action rule ar at list head
                    if the condition for ar is true
                        return the conclusion part of ar
                    return currentAction
            end function
```

Bidding Agent Interface

---

04/28/2009  CIS Dept., UMass Dartmouth
Designing Bidding Strategies

A fictitious auction: an item with an estimated auction price of $1000.

Six bidding agents: Bidder 1 to Bidder 6
- Bidder 1-5 are active through the whole auction
- Bidder 6 joins at the middle of the auction

Run the auction three times with Bidder 2-6 using a normal strategy, but Bidder 1 using one of the three different bidding strategies:
- Aggressive strategy C1 – composed of simple strategy S1 and S2.
- Aggressive/Cautious strategy C2 – composed of simple strategy S1 and S3.
- Cautious strategy S3 – simple strategy.
Experimental Result - 1

Bidding Rates with Different Strategies

- Aggressive Strategy (strategy C1)
- Aggressive/Cautious Strategy (strategy C2)
- Cautious Strategy (strategy S3)

Strategy transition in C2

Experimental Result - 2

Bidding Prices with Different Strategies

- Aggressive Strategy (strategy C1)
- Aggressive/Cautious Strategy (strategy C2)
- Cautious Strategy (strategy S3)

highest final price
lowest final price
Conclusions and Future Work

- Provided a model-based specification approach for complex bidding strategies.
- Demonstrated how bidding agents can automatically place bids according to bidding strategies.
- Used a case study to show how our prototype can support analysis of agent-based online auctions.
- For our future work, we plan to improve the GUI for visual specification of complex bidding strategies.
- Use agent-based online auction system as a test bed for shill detection.
- Implement a trustworthy agent-based online auction system.

Questions?

The slides for this talk can be downloaded from

http://www.cis.umassd.edu/~hxu/Projects/ATM