

CAWP: A COMBINATORIAL AUCTION WEB PLATFORM

A MASTER'S THESIS

in

Computer Engineering

Atılım University

by

İBRAHİM CERECİ

NOVEMBER 2009

CAWP: A COMBINATORIAL AUCTION WEB PLATFORM

**A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
OF
ATILIM UNIVERSITY
BY
İBRAHİM CERECİ**

**IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF MASTER OF SCIENCE
IN
THE DEPARTMENT OF COMPUTER ENGINEERING
NOVEMBER 2009**

Approval of the Graduate School of Natural and Applied Sciences, Atılım University.

Prof. Dr. K. İbrahim Akman

Director

I certify that this thesis satisfies all the requirements as a thesis for the degree of Master of Science.

Prof. Dr. K. İbrahim Akman

Head of Department

This is to certify that we have read the thesis “CAWP: A Combinatorial Auction Web Platform” submitted by “İbrahim Cereci” and that in our opinion it is fully adequate, in scope and quality, as a thesis for the degree of Master of Science.

Assoc. Prof. Dr. Mohammed Rehan

Co-Supervisor

Examining Committee Members

Prof. Dr. Ali Yazıcı

Asst. Prof. Dr. Murat Koyuncu

Asst. Prof. Dr. Sanjay Misra

Asst. Prof. Dr. Hürevren Kılıç

Supervisor

Date: 16.11.2009

I declare and guarantee that all data, knowledge and information in this document has been obtained, processed and presented in accordance with academic rules and ethical conduct. Based on these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

İbrahim Cereci

ABSTRACT

CAWP: A COMBINATORIAL AUCTION WEB PLATFORM

Cereci, İbrahim

M.S., Department of Computer Engineering

Supervisor: Asst. Prof. Dr. Hürevren Kılıç

Co-Supervisor: Assoc. Prof. Dr. Mohammed Rehan

November 2009, (70) pages

In this thesis an online platform, which enables users to both sell and buy items in a Combinatorial Auctioning environment is designed and implemented. The platform adopts a Sealed-Bid Single-Unit Combinatorial Auction. The platform can be used to auction online, without having any knowledge about solving winner determination problem known to be important for profit maximization of the auctioneer in Combinatorial Auctions. Combinatorial Auctions are rarely developed for online auctioning. Combinatorial Auction Web Platform fills that void, it gives general consumer a platform to buy and sell goods in a combinatorial trading environment. Unified Modeling Language models are used for describing the system. Platform called Combinatorial Auction Web Platform is implemented using open source tools and technologies. Case studies are developed to show capabilities and functionalities of the platform.

Keywords: Online Auctions, Combinatorial Auctions, e-commerce, winner determination problem

ÖZ

CAWP: BİR KOMBİNATORİYEL İHALE WEB PLATFORMU

Cereci, İbrahim

Yüksek Lisans, Bilgisayar Mühendisliği Bölümü

Tez Yöneticisi: Yrd. Doç. Dr. Hürevren Kılıç

Ortak Tez Yöneticisi: Doç. Dr. Mohammed Rehan

Kasım 2009, (70) sayfa

Kullanıcılarının kombinatoriyel ihale ile alım ve satım yapmasını sağlayan bir internet platformu tanımlanmıştır ve geliştirilmiştir. Platform, açık artırmadaki her ürüne kalem bazında teklif verilebilen, kapalı zarf usulü çalışan bir kombinatoriyel ihale sistemi geliştirme altyapısıdır. Platform kullanıcılara, kazanan ihalelerin tespitinde satıcının gelirini azami mertebeye çıkarma probleminin bilinen bir algoritma vasıtasıyla otomatik olarak çözümüne olanak tanımaktadır. İnternet üzerine kombinatoriyel ihale tabanlı çalışan nadir sayıda sistem bulunmaktadır. Çoklu ihale internet platformu (Combinatorial Auction Web Platform) bu boşluğu doldurmakta ve genel internet kullanıcılarının da kombinatoriyel ihalelerle mal alıp satabileceği bir ortam yaratmaktadır. Birleşik modelleme dili (Unified Modeling Language) yardımıyla sistem tanımlanmış ve platform açık kaynak teknolojiler kullanılarak geliştirilmiştir. Sistemin işlevselliğini ve yeteneklerini göstermek amacıyla da örnek olay incelemeleri (Case Study) yapılmıştır.

Anahtar Kelimeler: Çevrimiçi ihale, kombinatoriyel ihale, e-ticaret, kazanan belirleme problemi

To My Parents

ACKNOWLEDGMENTS

I would like to thank my thesis supervisor Asst. Prof. Dr. Hürevren Kılıç for his direction, assistance, and guidance. Without his contributions, this thesis would not have been completed to the full. Thanks for always being there.

Also I want to thank all my past instructors and my dear friends, for putting this hunger for knowledge in me and making me the computer engineer I am today.

Last but not least, I would like to dedicate this thesis as a gratitude to my family, for their unlimited support and motivation.

TABLE OF CONTENTS

ABSTRACT	iii
ÖZ	iv
ACKNOWLEDGMENTS	vi
TABLE OF CONTENTS	vii
LIST OF FIGURES	ix
LIST OF TABLES	x
LIST OF ABBREVIATIONS	xi
CHAPTER 1	1
INTRODUCTION	1
CHAPTER 2	4
ONLINE AUCTIONS.....	4
2.1 Auctions	4
2.3 Electronic Auctions.....	6
CHAPTER 3	10
COMBINATORIAL AUCTIONS.....	10
3.1 Combinatorial Auctions and Their Types.....	10
3.2 Winner Determination Problem.....	12
3.3 Solutions to Winner Determination Problem.....	14
3.3.1 Combinatorial Auction Structured Search (CASS) Algorithm:.....	14
3.3.2 Combinatorial Auction Branch on Bids (CABOB) algorithm.....	16
3.3.3 CPLEX Solver.....	17
CHAPTER 4	18
COMBINATORIAL AUCTION WEB PLATFORM.....	18
4.1 Overview of the Product	18
4.2 Technical Details.....	21
4.3 Business Protocol and Policies.....	30
4.3.1 Policies	30

4.3.2 Business Protocol	30
4.4 Performance of the System	32
4.5 Risk Management.....	34
4.5.1 Business Risks and Solutions.....	34
4.5.2 Technical Risks and Solutions	35
4.6 Some Limitations about the CAWP.....	37
CHAPTER 5	38
CASE STUDIES	38
5.1 Custom Built Furniture Auction	38
5.2 Web Service Combinatorial Auction	43
CHAPTER 6	44
CONCLUSIONS.....	44
REFERENCES.....	46
APPENDIX A	49
APPENDIX.....	49
A. Privacy Policy	49
B. User Agreement.....	53
C. Optimizer Handler written in Java	58
D. Web Site Code Samples	65

LIST OF FIGURES

Figure 1. Use case diagram of user authentication	22
Figure 2. Use case diagram of auction creation process	22
Figure 3. Use case diagram of advertising upcoming auctions.....	23
Figure 4. Use case diagram of bidding process.....	23
Figure 5. Activity Diagram for Winner Determination Process	24
Figure 6: Use case diagram of auction results	25
Figure 7. Use case diagram of trade settlement	25
Figure 8. Use case diagram of general system usage.....	26
Figure 9. Deployment Diagram of the system	27
Figure 10. Database Entity Relation Model Diagram of CAWP	27
Figure 11. Initial Screen of the CAWP	39
Figure 12. Modification of CAWP parameters	39
Figure 13. CAWP screen after changing title picture and CSS	40
Figure 14. Adding a product to CAWP.....	41
Figure 15. Creating an auction.....	41
Figure 16. Solver console report for solving WDP problem generated by CAWP ...	42
Figure 17. Result of the Auction of the given scenario for John Smith.....	42
Figure 18. Result of the Auction of the given scenario for Dave Murray	43

LIST OF TABLES

Table 1. Results of the CASS tests for variable problem sizes.....	33
Table 2. Performance results for multiple solvers running on a single machine	34

LIST OF ABBREVIATIONS

CA	-	Combinatorial Auction
CAS	-	Combinatorial Auction Systems
WDP	-	Winner Determination Problem
NP	-	Nondeterministic Polynomial time
NP-complete	-	Nondeterministic Polynomial time complete
B2B	-	Business to Business
B2C	-	Business to Consumer
C2C	-	Consumer to Consumer
DoS	-	Denial of Service
CASS	-	Combinatorial Auction Structured Search
CAWP	-	Combinatorial Auction Web Platform
UML	-	Unified Modeling Language

CHAPTER 1

INTRODUCTION

Auction is a trading process where auctioneer provides goods or services and buyer bid to these goods or services. At the end the highest bidder wins. Online auctions are the auctions which are held over the internet. Rapid growth of internet makes online auctioning important, as it reduces the time and space cost of the offline auctioning mechanisms.

Combinatorial Auctions (CA) are the auctions that bidders place bids on combinations of items rather than a single item [1]. CAs are commonly used in sectors like transportation[2], bus routes, airport landing rights[3], power exchanges, carbon permits, and radio spectrum for wireless communications services[4][5].

CA history goes back to 70's[6], with the increase of computation capabilities of the computers, their automation become more and more popular. Another reason of CAs being popular is that it gives the bidders capability to fully express their preferences. Especially if the items in the auction are complements, set of items may be valued more than the sum of values for each individual item. At the same time auctioneer has great benefit from using CA, because allowing bidders express themselves in combinatorial manner, often improves economic efficiency and causes greater auction revenues [7].

Automation of the CA is also important, because sellers over the internet may also want to maximize their revenues and let their bidders flexibly express themselves while bidding to the items in the auction. Every online auction mechanism including

online CAs, are example uses of e-commerce, because e-commerce deals with buying and selling of goods or services over electronic systems, such as internet.

One of the main problems in the online CA is deciding which bids will be the winner bids in favor of the auctioneer. CA allows bidder to bid bundles of items in an auction, these bundles may overlap, our aim is to find the subset of all given bids which will maximize the revenue of the seller. In literature, this problem is known as Winner Determination Problem (WDP) which is known to be NP-complete problem [8]. NP-Complete problems are hard problems which don't known to have solution in polynomial time. As the problem size increases, time required to solve the problem increases very rapidly.

To the best of author's knowledge, there are no CA platforms for public consumer auctions, because of the hardness of the WDP problem, online CAs cannot perform well for the large problem scales. But with giving limitation to maximum number of items in an auction problem size can be reduced to a solvable instance. In this thesis, an online customizable Combinatorial Auction Platform which is named Combinatorial Auction Web Platform (CAWP) is developed. This platform will be used by the general consumers who want to sell or buy goods in a combinatorial way. These general consumers are not expected to know detailed information about how to solve the WDP.

Every commerce site should have some privacy policies and a user agreement about its usage. CAWP adapted the privacy policy and user agreement from a known e-commerce site eBay. The policies are altered to fit the area of CA trading. An e-commerce site like eBay does not consider any specific policy about optimal winner determination since the highest bidder is always the winner. But in CAWP, we solve the problem in such a way that maximizes the revenue of the auctioneer.

CAWP is developed using open source tools and technologies. CAWP is independent from any operating system, the sellers and buyers will interact with the system through a web browser, no addition program installation is required, and these make CAWP free and reachable from any operating system and on any browser. PHP is used in the system for web development. For database management system MySql is selected. The WDP solver is a C program named CASS. Java is used to implement the solver handler which connects database to solver. There are a

lot of successful e-commerce sites for general consumer on the internet, but to the best of the author's knowledge none of them has CA capabilities, so if a user wants to trade in an online CA platform CAWP should be the choice.

Two case studies are generated to show the CAWP functionalities, first one is a web site where a craftsman makes custom furniture and sells them in a CA. The second case study is for selling the usage rights of web services in hourly basis. Bidders can bid for combinations of hours of usage.

The remainder of the thesis has the following organization. In Section 2, Auctions and their automation are reviewed. In Section 3, Combinatorial Auctions, Winner Determination Problem are explained. Different solutions to WDP are discussed. In Section 4, Combinatorial Auction Web Platform is defined with both its functionalities and technical details. In Section 5, example case studies are given; a trading scenario is generated and shown over the case studies. The last section includes the conclusion and future work.

CHAPTER 2

ONLINE AUCTIONS

Emergence of the Internet has changed how people buy and sell goods. Even new types of electronic marketplaces have been developed to create more efficient markets [9]. Online Auctions have been the one of the most successful electronic markets. Most known example of online auction site is eBay which has a net income \$1.677 billion, and over 16200 employees by the year 2009 [10]. Success of the online auctions comes from the capabilities it gives to both buyers and seller. As a buyer you can bid on large number of items from different sources and you have the potential to find goods in lower prices. As a seller you reach great number of potential buyers. When you put your items for auctions to the online market, you eliminate the cost of time and place of the offline market. In this chapter, general definitions of auctions, online auctions and combinatorial auctions will be given, and different kinds of auctions are introduced.

2.1 Auctions

Auction is a trading process, where buyers bid to the goods or services that seller provides. At the end of the process highest bidder wins. Auctions are important for practical, empirical, and theoretical reasons. First, they are practically important because huge amount of goods are sold through auctions. Auctions generally used in markets such as antique business, collectibles, wine sales, real estate, second hand goods, and radio spectrum auctions.

Second, auctions are empirically important, because they have been the basis of major empirical researches on the game theory field.

Finally, auctions are theoretically important because they are the basis of theoretical works such as, efficiency of a given auction design, optimal and equilibrium bidding strategies, and revenue comparison [11].

There are four basic auction types which are widely used. Most of the other auctions have been derived from these four auctions [11].

First one is the **First-Price Sealed Bid Auction**. In this auction all bidders submit their valuations in sealed bids at the same time, by doing that it is guaranteed that no bidder knows the bid of the others. The highest bidder, who pays the price and gets the good.

Second one is the **English Auction**. It is also called “open-cry ascending price” auction. English auction is the most common form of auction used today [12]. In this form bidders bid openly against each other. Each bid should be higher than the previous one. The auction ends when no bidder is willing to raise the final bid or bidding period is over. The highest bidder gets the good.

The third form of auction is the **Dutch Auction**. It is also called “open-cry descending price” auction. This auction type is similar to the English auction. The auctioneer begins with setting a high price to his good. Initial price is gradually lowered until one bidder accepts to pay that amount. Last announced amount is paid by the bidder.

Last basic type of auction is **Vickrey Auction** [13]. It is also called sealed-bid second-price auction. It is identical to the first-price sealed-bid auction except that winning bidder pays not his bid amount but the second highest bidder’s amount.

These four auctioning types can be compared from different perspectives. Dutch auction or first-price sealed bid should be chosen if the aim is to maximize revenue of an auctioneer.[14] This is because the bidders want to insure themselves by bidding slightly more for the good. If the social welfare is wanted to be maximized, Vickrey auction should be chosen, because winner pays does not pay the maximum bid. Another way to compare these auctions is how resistant they are to the lies of the auctioneer or the bidder. For example in Vickrey auction auctioneer may lie about

the price winner have to pay. It can be avoided by taking signatures of the bidders with the bids and prove the winner that the amount you want is the second best. In open-cry auction since all bidders see others bids there is no way for lying. The first-price sealed-bid auctions are not affected from lying of the auctioneer, because winners know the amount they offered.

There are also secondary types of auctions which can be combined with the basic auctions.

All-Pay Auction is the auction, in which all bidders should pay their bid even if they won't win the auction [15].

Buyout Auction is an auction where seller sets an amount to the good, which you can pay and win the auction instantly [16].

No-Reserve Auction is the auction where there is no base price for bidding. Item will be sold whatever the final bid is given. This may be below the actual value of the item [17].

Reserve Auction is the type of auction if the bid to the item below certain amount, it may not be sold. This reserve price may or may not be announced to the bidders [17].

Reverse Auction is the auction where sellers are trying to supply a service to the buyer by decreasing the amount of the price. The auction ends when no seller wants to provide the service less than current price [18].

Top-Up Auction is a variant of all-pay auction, where each bidder should pay the difference between their bids and the next lower bid.

2.3 Electronic Auctions

On the last decade the Internet grew rapidly. Both bandwidths and system availability are increased. As a result of this, the trading done on the Internet grew too. Internet allowed businesses to reach potential customers which would not be reachable otherwise. Since sellers and buyers do not need to be at the same place for trading to happen, digital trading also reduced the cost of time and place of the classic offline trading. Electronic trading improves Business to Business (B2B),

Business to Consumer (B2C) and especially Consumer to Consumer (C2C) trades. Even a simple person who wants to sell his second hand goods may reach huge potential customers by using online trading.

Electronic auctions became an important part of the electronic trades. The most commonly used online auction is English auction [19]. Market leader eBay also uses English auction on its trades. Auctions generated in eBay can have both reserve price, and buyout price.

In general, complete trading process of any online auction has the following basic steps [20]:

1. **Initial buyer and seller registration:** In this step authentication of the parties are done.
2. **Setting up a particular auction event:** Creating the goods, describing them on the site and setting auction rules and starting the auction.
3. **Scheduling and advertising:** Upcoming auctions are notified to attract potential buyers.
4. **Bidding:** Collecting all the bids, controlling the bid validity is done until bidding period is over.
5. **Evaluation of bids, closing the auction:** Apply the auction closing rules and determine the winners of the bids, notify winners and losers.
6. **Trade settlement:** Payment and good delivery are realized at this step.

These steps may vary from one auction method to another; some of them like bidding period may be repeated.

CAWP contains these steps; each step is shown in Section 4, using use case diagrams.

Most of the electronic auctions use open-cry auctions (English auctions) and they totally trust the fairness of the auctioneer [21].

However, there might be some abuses in electronic auctions such as:

Bid shielding: It occurs when a high bid is withdrawn at the last minute so a lower bid is accepted as a winner. Note that an electronic auctioning system may allow withdrawn of bids, but should also prevent it being a fraud.

Bid siphoning: It happens when a bidder connects the auctioneer and makes the trade outside of the electronic platform, to avoid paying the commission.

Shilling: is bidding just to increase the price of the good. Usually this person is the seller himself with another account or an accomplice. Shilling is hard to detect, but it can be data mined after having enough data about the sellers all auctions and the bidders.

Sniping: is waiting to bid for the last second, trying to avoid other bidders to increase the price of the auction. If it is combined with a Denial of Service (DoS) attack, which is an attempt to make computer services unavailable for its intended users, it becomes a serious problem.

Identity theft and unauthorized access: If there are weak passwords set user accounts in the electronic auctions, or the system is not well coded that it allows unauthorized operations to be done, than such person can mess with the users, bids and auctions.

Misrepresented or non-existent items: An auctioneer may claim to have a good, start an auction and at the end he may not have that specific good or any good at hand. This is a common problem in online auctioning. To avoid misrepresented or non-existent items, auction sites come up with some security measures. Some of these measures are as follows:

- Give positive or negative feedback to buyers and sellers.
- Register the credit card information of the sellers and buyers, prevent multiple identity usage.
- Offer insurance services to cover losses caused by fraud.
- Escrow services are one of the most common security measures. It prevents payment to be done before buyer has the product. Usually buyer pays the electronic auction provider. Seller is notified with that payment. Seller sends

the good to the buyer. When buyer say good has reached and it is legitimate, the money is transferred to the sellers account.

For the sake of standardization every electronic auction platform are expected to have the following properties [12]:

1. **Anonymity:** Loser bidders should not be identifiable.
2. **Non-cancelability:** A winner is always identified that he cannot deny having bid to the auction.
3. **Public verifiability:** Anybody could publicly verify the winning bid is really the highest value and valid.
4. **Unforgeability:** Impersonation of sellers and bidders should be prevented.
5. **Robustness:** Auction process should not be interrupted, even with invalid bids.
6. **Fairness:** Every bid should have the same priority; there should be no favor to any individual's bids.
7. **Efficiency of bidding:** The computation of determining a winner bid and verifying that should be practical.

CAWP contains most of the above properties to some degree. Loser bids are not identifiable. Bidder login to the system before bidding so a winner is always identified. CAWP is not publicly verifiable, only seller can verify that winning bids combination is the highest amount. Impersonation of another bidder is not possible without stealing their user name and password. A person can impersonate another person not in the CAWP, and jeopardize the fairness of the trade, but after single act this person will be notified and receive bad comments, or it can go to account suspension. CAWP prevents bidders to place invalid bids, and keeps the auction process uninterrupted. In CAWP bids or bidders have no priority, so fairness is obtained. CAWP deals with WDP of CAs, so determining winner is not practical, but CAWP limits the number of items that can be in a single auction so we can get a response from the WDP solver in an acceptable time.

CHAPTER 3

COMBINATORIAL AUCTIONS

It is hard to decide which bids are the winners in CAs. We allow bidders to express themselves freely and put any combinations of items in the auction, but it comes with an enormous possibility in the solution space. In this chapter, we will look at the winner determination problem, how to solve it, solution used in this thesis and quality of this solution.

3.1 Combinatorial Auctions and Their Types

Combinatorial Auctions are the auctions where multiple goods are sold simultaneously and bidders are allowed to bid any combinations of those goods. Bids been made to the bundle of goods are won or lost at the same time. These types of auctions are desirable when a bidder's valuation to the bundle of goods is not equal to sum of his valuations of the individual items. A bidder may want a good if and only if he won another good on that auction. In CAs, bidders can still bid to the individual goods in the auction. So one can think CAs as a special case of one-shot sealed-bid auctions, where bidders can express themselves more flexibly.

In [8] it is stated that

“Formally, a combinatorial auction is a pair $\langle I, B \rangle$,

where

$I = \{I_1, K, I_m\}$ is the set of items the auctioneer has to sell, and

$B = \{B_1, K, B_n\}$ is the set of bids from the buyers interested in the items in I .

Each bid B_i has

The form $\langle \text{item}(B_i), \text{pay}(B_i) \rangle$, where $\text{pay}(B_i)$ is a rational number denoting the price a buyer offers for the items in item $B_i \subseteq I$.

An outcome for $\langle I, B \rangle$ is a subset b of B such that $\text{item}(B_i) \cap \text{item}(B_j) = \emptyset$, for each pair B_i and B_j of bids in b with $i \neq j$.”

Combinatorial Auctions are being used in many real world situations. For example in 2002, Nigeria sold regional fixed wireless access licenses on a sealed-bid combinatorial auction [22]. Also web services are allocated using CAs. Nature of the web services may require more than one web service to run at the same time, so bidders bid combinations of web services [23]. CAs also used in Supply Chain Formation. The supply chain formation has difficult coordination issues for distributed negotiation of the protocols. Parties must negotiate for multi level production relationships with important interdependencies among inputs and outputs of each level. CAs addresses this problem by global optimization over expressed offers to engage in compound exchanges. [24]

CAs can be categorized according to certain criteria described below:

Categorizing CAs with regard to bidding style:

Open-Bid Combinatorial Auctions: In this type of combinatorial auction, bidders are aware of competing bidders' bids. All bids are publicly announced.

Sealed-Bid Combinatorial Auctions: They are the auctions where each bidder is only aware of his bids. After bidding process is completed and winner is determined, it can be announced. Hiding the bids is necessary during bidding time.

Categorizing CAs according to the number of goods:

Single-Unit Combinatorial Auctions: are the auctions where amount of the each individual item is one. For example if there are five identical items, they must be placed to the auction as different items which have the same product information.

Multi-Unit Combinatorial Auctions: are the auctions where amount of the individual items may be more than one.

Categorizing CAs based on pricing:

Reserve Combinatorial Auctions: are the auctions where seller may put a base acceptance price on each item while auctioning. Since in CA bundles of items are bid together, amount of bid should be more than the sum of the items' base prices in the bundle.

Non-Reserve Combinatorial Auctions: On these auctions seller cannot put a base acceptance price to their items in the auction. Winners pay the amount they bid and get the items even they are below their original value.

Reverse Combinatorial Auctions: CAs can also be used in reverse auctions. Buyers may want series of items and suppliers bid group of that items. Least expensive bids are accepted.

CAWP adopts a Sealed-Bid Single-Unit Combinatorial Auction mechanism where items can have reserved prices, set by the Auctioneer. This choice limits the capability of CAWP; bidders can not see their competitors' bids. Every item put in an auction is single unit, if two identical items will be put in an auction, they should be placed separately.

3.2 Winner Determination Problem

Winner Determination Problem is the problem of deciding the allocation of winner bids, in a set of bids placed to the auction, so that the revenue of the auctioneer will be maximized.

Formal Definition of the Problem [25]:

Denote the set of goods by G and the set of non-negative real numbers by R^+ .

A bid $b = (p_b, G_b)$ is an element of $S = R^+ \times (2^G - \{\emptyset\})$. Let B be a subset of S .

A set $F \subseteq B$ is said to be feasible if $\forall b, c \neq b \in F, G_b \cap G_c = \emptyset$. Denote the set of all feasible allocations for B by $\Phi(B)$. Further, let $G(B) = \cup_{b \in B} G_b$ be the set of goods contained in the bids of B .

Problem: Find an allocation $W \in \Phi(B)$ such that $\forall F \in \Phi(B) \sum_{b \in F} p_b \leq \sum_{b \in W} p_b$. Such allocation is said to be optimal or revenue maximizing.

WDP is hard because one would need to check all subset of the bids, whether they are feasible (no conflict of items) and how much revenue they provide. A feasible subset of the bids that has the maximum revenue is an optimal solution. There are 2^k subsets of bids, k being the number of bids; this makes the problem hard [26].

WDP can be explained with an example problem.

Example 1:

There is an auction A with three items I_1, I_2, I_3 , there are four bids $B_1 - B_4$, a bid consists of, list of items and a bid value.

These are the bids:

$B_1: (\langle I_1, I_3 \rangle, 30)$, $B_2: (\langle I_2, I_3 \rangle, 20)$, $B_3: (\langle I_2 \rangle, 15)$, $B_4: (\langle I_1, I_2, I_3 \rangle, 30)$

Problem here is to decide which bids to choose as winners, so that they will maximize the revenue of the seller. Remember every item in the auction is single unit, so if two bids contains same item in their list, they can not both be winners. If the item and bid size is small enough this problem can be solved by hand. As the numbers increases, since problem is NP-Complete in nature, it becomes a hard problem even for computers.

Winner bids in Example 1 should be B_1 and B_3 , so the auctioneer will gain the maximum possible amount which is 45.

3.3 Solutions to Winner Determination Problem

Following Subcategories lists most known and widely used WDP solvers. There are three main factors affecting the complexity of a given WDP. These three factors are number of goods, number of bids and distribution of these bids. If there are some dominant bids in the system solution will be found in dramatically shorter time, since dominant bids are the bids which contain large number of items and significantly high bid amount. When you accept a dominant bid it helps maximizing the auctioneer revenue and reduces the solution space, causing a solution to be found faster.

CASS is chosen in CAWP. CASS opens its sources to everyone and is a known and efficient algorithm [25]. It includes heuristics which fits the nature of CA problems. It reduces the search space by applying four improvements. Even for the big problem sizes, it uses the system resources minimum in order to avoid unexpected crashes.

The following alternative algorithms have not been traced using example problems, because some parts of their algorithm are not well defined in related papers. For example there are sentences asking to do something without giving methods of doing them, such as "do the upper bounding", "prune the path if it leads to no better solution", or "k variable is calculated at each run time".

3.3.1 Combinatorial Auction Structured Search (CASS) Algorithm:

The algorithm uses exhaustive search for determining optimal solution. CASS is basically a brute-force search approach with four significant improvements [25]. CASS structures the search space to avoid conflicting bids which are the bids that some or all items in them overlap with each other, keep the result of the searches done up to that point and prunes the search tree. A test suite is developed to create sample auction setups and test the systems performance. This test suite is known as CATS [7] [27].

Brute Force Algorithm in CASS:

"Suppose there are $|G|$ goods $1, 2, \dots, |G|$ and $|B|$ bids $1, 2, \dots, |B|$."

First, bids that will never be part of an optimal allocation are removed. That is, if for bid $b_k = (p_k, G_k)$ there exists a bid $b_l = (p_l, G_l)$ such that $p_l > p_k$ and $G_l \subseteq G_k$, then b_k is removed because it can always be replaced by b_l , increasing revenue. Then for each good g , if there is no bid $b = (x, \{g\})$ a dummy bid $b = (0, \{g\})$ is added. The brute-force algorithm examines all feasible allocations through a depth-first search. Let x be the first bid and y be the last bid. The implementation follows:

1. If x does not conflict with the current allocation, add x to the current allocation
2. Increment x
3. If more bids can be added to the allocation, go to 2.
4. Update best revenue and allocation observed so far.
5. If y is contained in the current allocation, remove it, set $x = y + l$ and repeat from 2.
6. Decrement y .
7. If y is not the first bid, go to 5.”

Improvement made to the Brute Force Approach in CASS

Improvement #1: Bins

CASS checks for the conflicting bids and allocate conflicting bids accordingly in the search space. It puts conflicting bids into a respective bins, and if a bid is selected from a bin, all the other bids in that bin is directly avoided, since they will cause conflict. Just continue taking single bid from each bin.

Example:

Assume Bid 1 bids for items 1, 2, and 3 in the auction and the bid price is 100.

Assume Bid 2 also bids for items 1 and 2 in the same auction and the bid price is 50.

Improvement 1 puts Bid 1 and Bid 2 into same bin, and if one of them is chosen at any result configuration, the other one is directly omitted.

Improvement #2: Caching

Cache the partial allocations with a threshold of k which is defined at each runtime. Threshold is set because caching can be infeasible more than k goods exist. Since we see solution space as a tree of bids, some sub trees may be the same these are the

partial allocations in the solution space. Caching occurs on these trees if they do not exceed an item threshold which is k , calculated by the runtime for each bin.

Improvement #3: Pruning #1

If a search path can be proved to not lead a new best allocation it is pruned.

Example:

When a new best solution is found all the previous cached branches can be pruned.

Improvement #4: Pruning #2

When it is proven that no more bids can be added to the current allocation search tree can be pruned there.

Example:

If a bid from all the bins is selected, then there is no other bid to add to the current solution, algorithm keeps the revenue this tree branch provides and prunes that branch.

Formal analysis of the CASS's computational complexity is not completed by its inventors. However they have empirical results against a brute force algorithm. As expected their heuristics decrease the time necessary to get an optimal result of the problem.

3.3.2 Combinatorial Auction Branch on Bids (CABOB) algorithm

CABOB is a tree search algorithm where tree is branched on bids [28]. CABOB makes a dept-first branch and bound search on the tree. Branch and Bound search is a search algorithm consist of a systematic enumeration of all possible solutions and prunes the large subset of fruitless candidate solution by using upper and lower bounding [29].

In the algorithm G is a bid graph.

In CABOB a data structure called bid graph is generated. The nodes on the graph are the bids; two bids share vertices if they share items in the bids. While searching down the tree, vertices are removed as well as the edges connected to those vertices this narrows down the search space.

CABOB uses upper and lower bounding. Upper bounding occurs when the solution can not be extended to a new optimal solution, and CABOB prunes that path. CABOB generates an acceptance level for each bid between 0 and 1. Then it takes all the bids higher than 0.5 and tries to put other bids by decreasing acceptance level, this gives them a lower bound. If the lower bound is high, it leads to more pruning and increase the speed of the solver.

3.3.3 CPLEX Solver

CPLEX is an optimization engine that belongs to a firm called ILOG [30]. It has a capacity to solve problems that has up to millions of constraints and variables. It is the leader of the optimization industry. Over 1300 corporations and government agencies use ILOG's CPLEX. Over 100 market leaders also use SAP and Oracle are examples of them. All the algorithms developed in the field compares themselves with the CPLEX solver, state their strengths and weaknesses against it [28, 31, 32].

CPLEX solver is an optimization package which includes solvers for Integer Programming and Linear programming problems. Integer problems are the problem which some or all the variables are restricted to be integral. Linear programming is to calculate the best outcome in a mathematical model which has a list of requirements represented as linear equations.

CPLEX solver can be used to solve WDPs, because WDPs can be formulated as integer linear programming formulations [26].

CHAPTER 4

COMBINATORIAL AUCTION WEB PLATFORM

In this section CAWP will be introduced in detail, with its functionalities according to the system and the user requirements. In Section 4.1, an overview of CAWP is given. In Section 4.2, technical details are explained using UML diagrams. Technology behind the CAWP is introduced. In Section 4.3, the adopted business policies and the protocols that we developed are covered. In Section 4.4, system performance results are given. In Section 4.5, risk factors and precautions for them are listed. In the last subsection discussions about CAWP is made.

4.1 Overview of the Product

The platform can be used to create and participate in Combinatorial Auctions. Product is called **Combinatorial Auction Web Platform (CAWP)**.

The platform consists of four main components. First one is the “web component” by which users can create auctions, make bid. The winner determination problem is generated by this component and results are notified via this component, also.

Second component is the “solver component”. Solver component takes the problems generated by the Web component, solves them and returns the solution.

Third component is the “database component” which is the glue that holds together web component and solver component. Every bid, auction, bidder information is kept

in the database and solver component gets necessary data from Database to create the problem file. After solver solves the problem it inserts results to the database.

Last component is the customization component. This component is necessary to take the CAWP and use it in your business without knowing web programming. You can modify some parameters so that site name, labels in the system, error messages according to platform user's needs.

Below is the list of CAWP properties based on system/user requirements that are considered during development:

1. Authentication of the users is a must for both Bidding and creating Auctions.
2. Any internet user can enter the CAWP site and view the products information, open auctions, but user must be logged in to have bidding and auctioning capabilities.
3. There is no hierarchy or priority between users of the system.
4. The system gives auctioneers set some parameters of the auction like auction name, auction end time, items in the auction, items' base prices.
5. Bidders can bid any bundle of items as long as they are at the same auction.
6. The system assists bidder the minimum acceptable amount of the bid which is the sum of base prices of items in the bid.
7. The system gives detail information about the bids and auctions. At any time user can see the state of an auction, whether it is in still bidding phase, or bidding phase finished but it is not solved yet.
8. All users can reach the information of an item which is in an auction that is in bidding phase.
9. Users can withdraw their bids until bidding phase is over.
10. Auctioneer can drop an auction if there are no bids placed on the auction.
11. Buyers and sellers can write comments about the people they trade.
12. Platform has an internal messaging system. Sellers and buyers can send private messages to each other.

13. The system does not require users to install any software to their computers. It is fully usable from a web browser; all the operations are held on the server side.
14. CAWP can run on both Windows and Linux machines. Only open source technologies are used creating CAWP.
15. Number of items that can be on a single auction is maximum 30.
16. An auction can be cancelled, if there are no bids placed on any item on that auction.
17. Real procurement of the goods is realized between seller and buyer. CAWP only gives buyers and sellers capability to comment about their trades, and give bad comments about the other side.
18. CAWP is a customizable product. Anyone can take and customize it using built-in customization tool. Anyone having knowledge about web technologies and CSS can further modify CAWP because it is a General Public License version 3 (GPLv3) licensed product.

There is another CA system created by Liping Fang and Yucheng Wang, named Online Iterative Combinatorial Auction System (OICAS) [33]. It is a prototype implemented in Visual Basic and Microsoft Access. OICAS has following characteristics:

- The auctioneer can determine who can participate in an auction.
- Only legitimate users (bidders) can participate in an auction.
- Bidders can bid for any bundle of items with acceptable price.
- The system shows bidders the required minimum winning price of their bids automatically.
- The system displays winning bids in real time.
- Bidders can trace the history of the auction any time.
- Bidders can see information about an item any time.

The main difference between OICAS and CAWP is that OICAS is an offline prototype implemented in Visual Basic language and Access database, CAWP is a real web application implemented using PHP for server side scripting, MySql for database management system, java for solver database integration, and C for WDP solver. These components can be deployed to different machines and run distributed.

OICAS lets auctioneer to choose bidders, CAWP believes in fairness for the bidders and does not let an auction to be available for only a certain bidders. In order to compare both programs' performances both should be online and given same problem data. As explained before the nature of the problem, bids can change the solution speed dramatically.

4.2 Technical Details

In Section 2.3 we listed the complete auction process for electronic auctions. In this section each step of auction process will be retraced and the actors and their available actions on the system will be displayed. UML diagrams will be used to describe the model of CAWP. In UML, use case diagrams are used to describe the interaction between user and the system. Activity diagrams are used to display the workflow. Deployment diagrams are used to show physical distribution of the system components over a network of computer. As one can see from below figures there is no role like administrator in CAWP. Administration should be done from the phpMyAdmin database administration panel.

Initial buyer and seller registration: In this phase both bidder and auctioneer should be authenticated by the system. Figure 1 shows both actors authentication. CAWP uses password authentication. Users of the system should login with their corresponding usernames and passwords. Every page at the CAWP site checks if the user is logged in or not. It avoids identity theft by this way. There is no additional authentication method to make sure the real identity of the person signing up to the system.

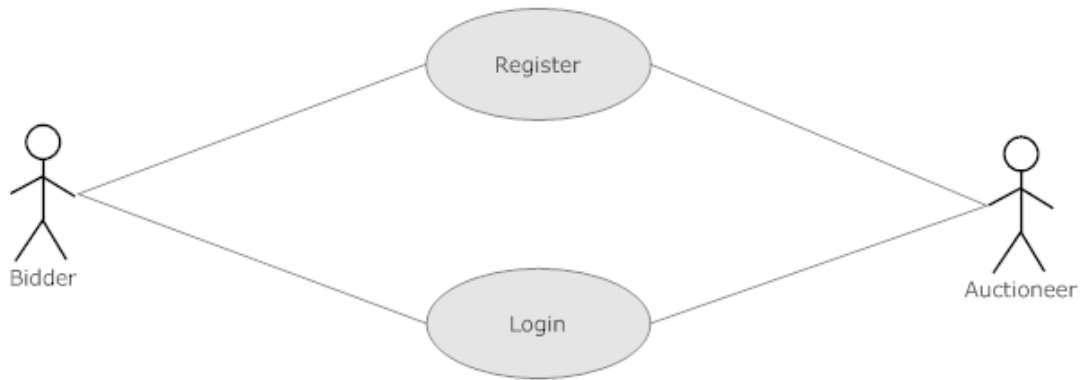


Figure 1. Use case diagram of user authentication

Setting up a particular auction event: Figure 2 shows the phase where an Auctioneer creates, edits, and deletes products that will be put to an auction. Auctioneer also creates and starts an auction. If no bids are placed, auctioneer can delete an auction.

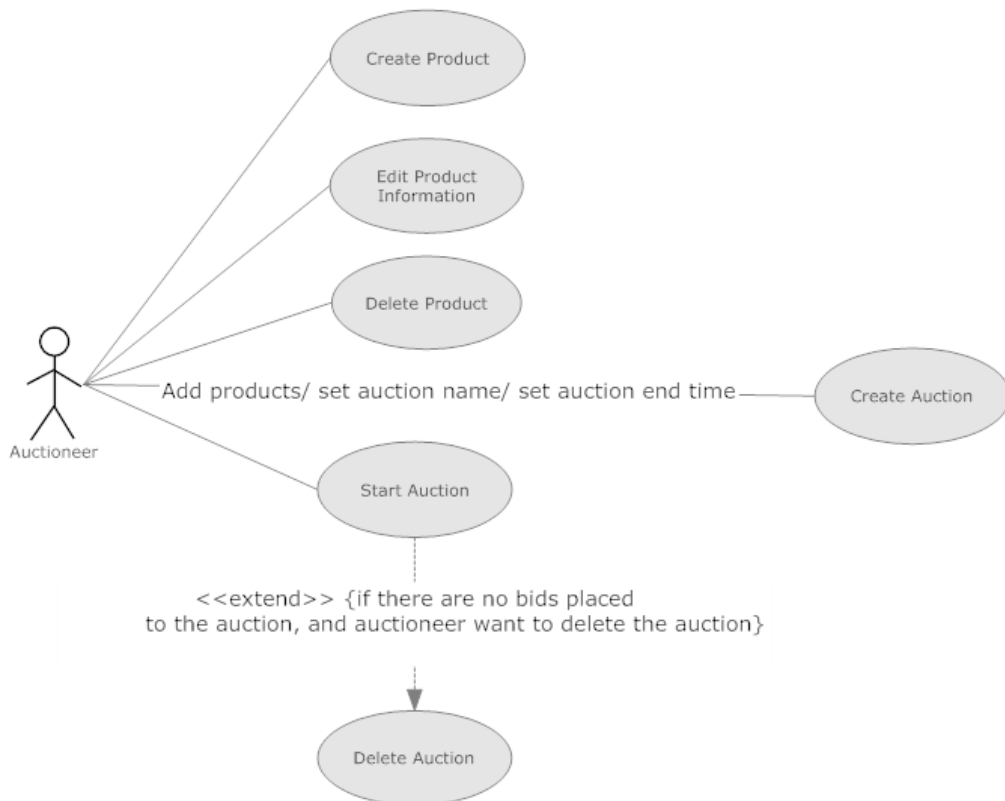


Figure 2. Use case diagram of auction creation process

Scheduling and advertising: CAWP lists the auctions which are about to end soon.

Figure 3 shows how upcoming auctions are advertised in CAWP.

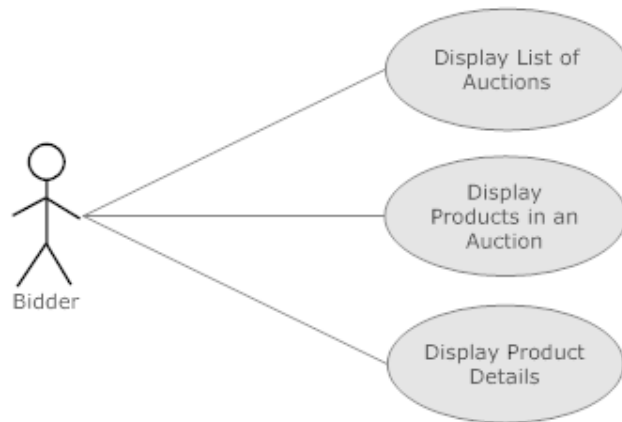


Figure 3. Use case diagram of advertising upcoming auctions

Bidding: Figure 4 shows the bidding process in CAWP. This is the period where bidding to a specific auction is available. Agents can perform these actions if and only if that bidding period is still continuing.

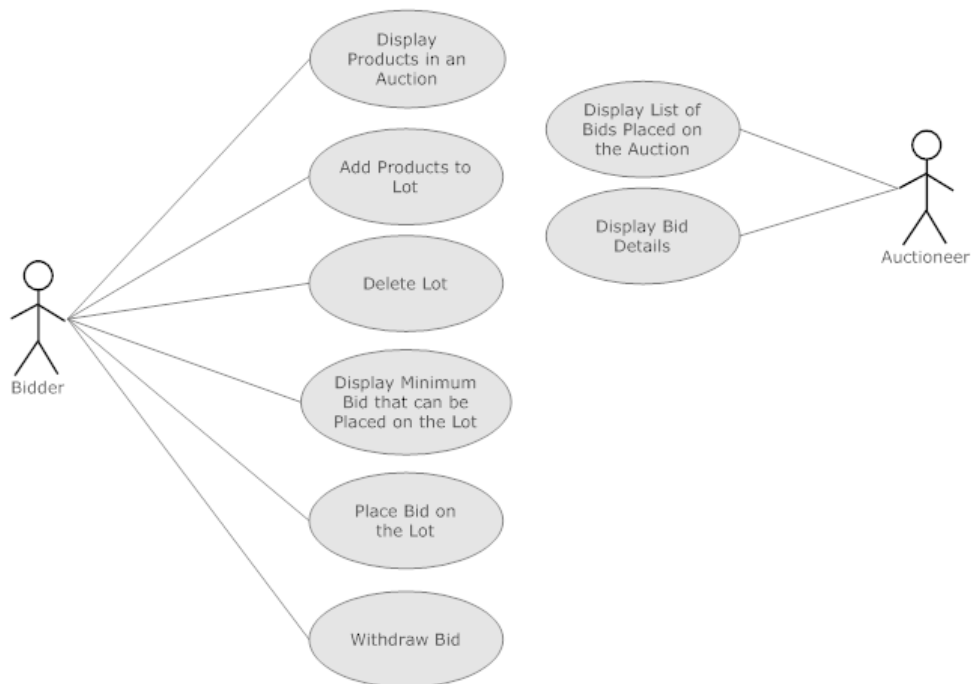


Figure 4. Use case diagram of bidding process

Evaluation of bids, closing the auction: Auction bidding time is ended and winner determination is started. This process will be handled on the background and the results are announced. Figure 5 shows how WDP solver program interacts with the remaining components of the platform. Figure 6 displays the winner and loser bidders' actions after an auction is finished.

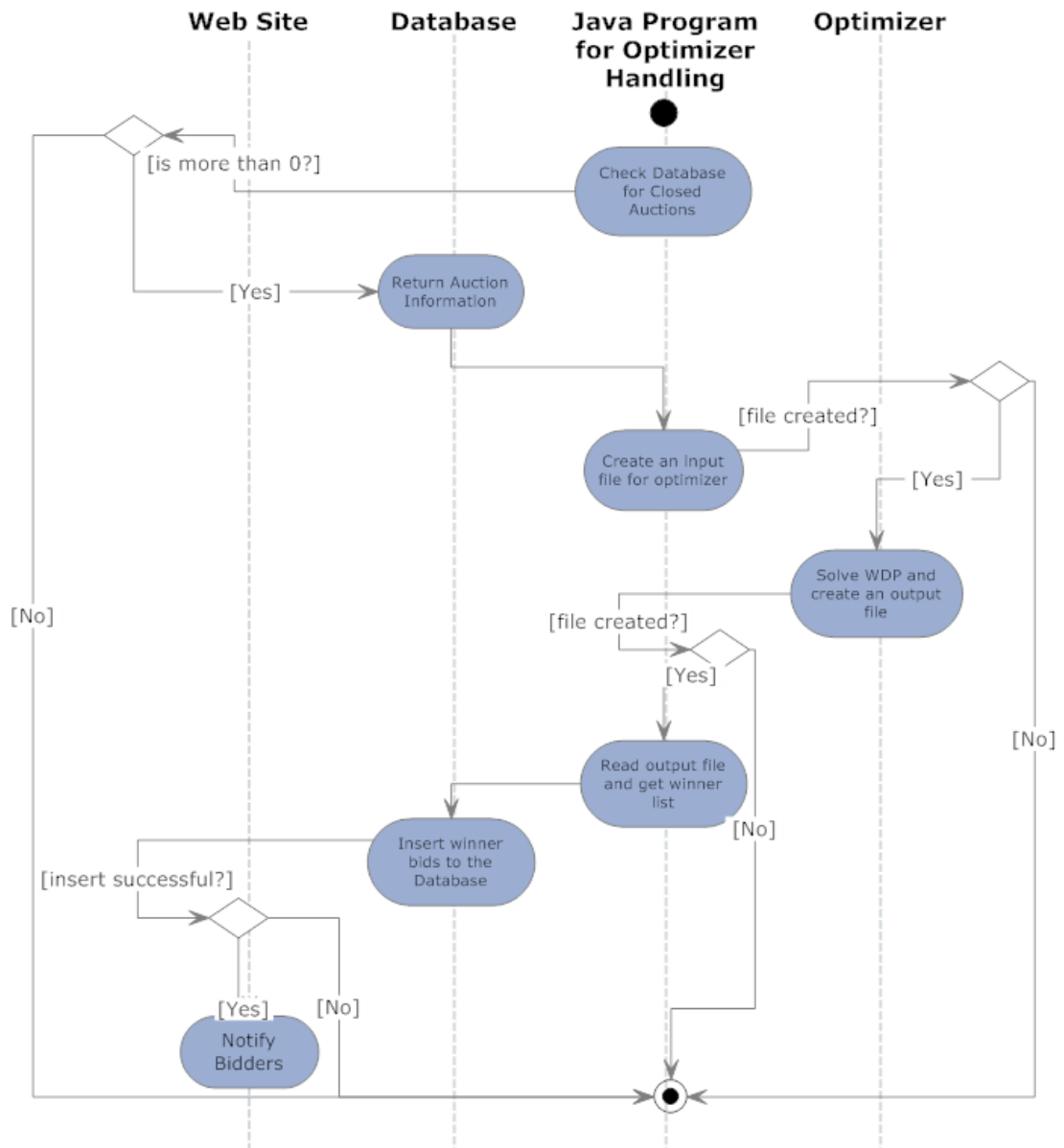


Figure 5. Activity Diagram for Winner Determination Process

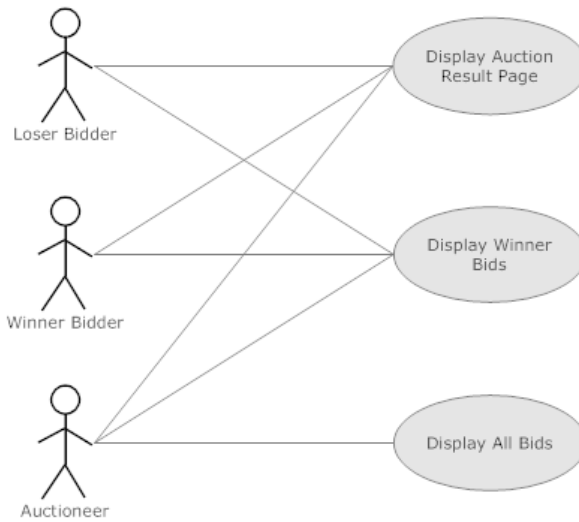


Figure 6: Use case diagram of auction results

Trade settlement: Figure 7 shows the phase where actual trading occurs between winner bidders and the auctioneer.

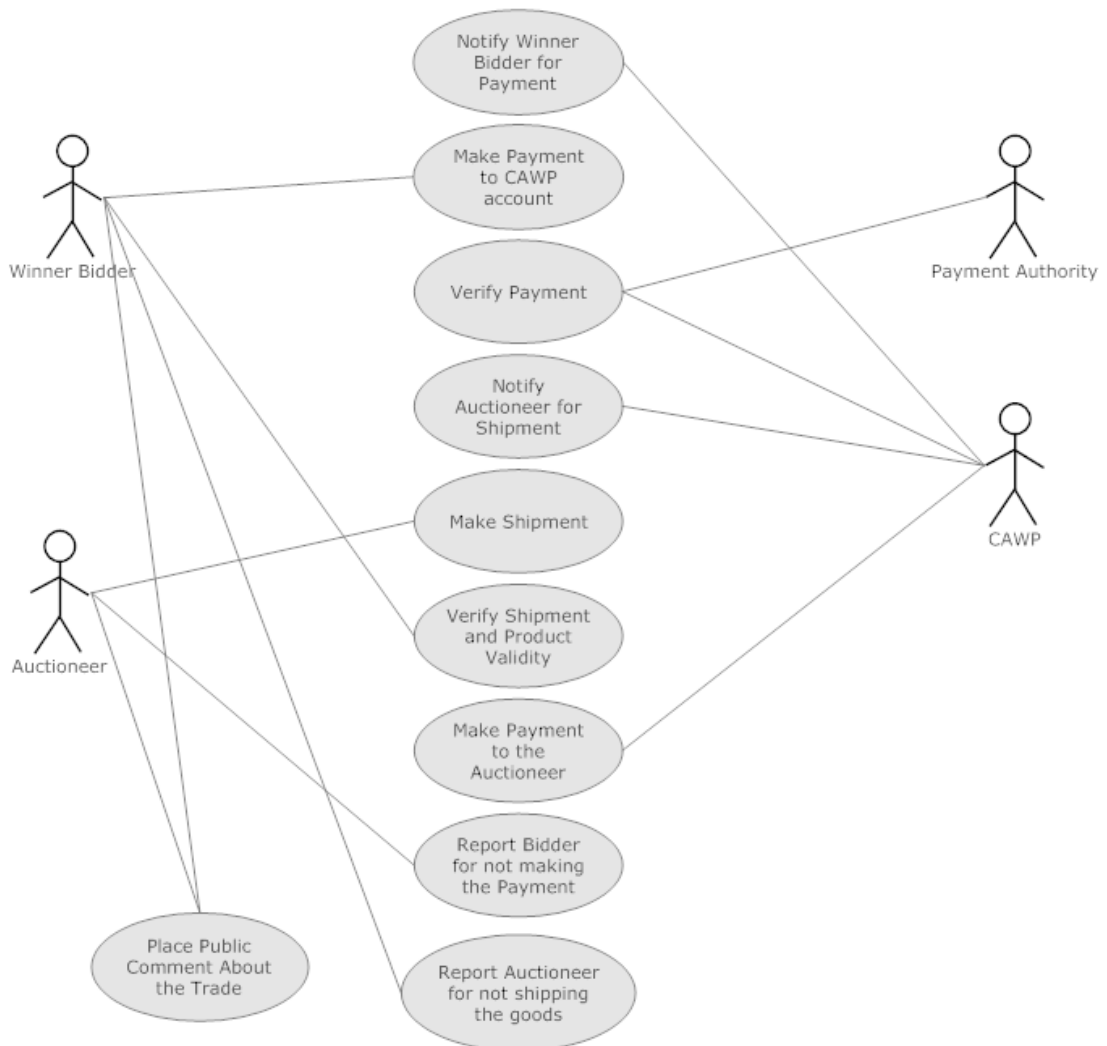


Figure 7. Use case diagram of trade settlement

These six phases covers the entire usage of the system by the actors through the trading process. Regardless to these six phases users can also interact with the system to make following actions shown in Figure 8, any time they want. In Figure 8, all the actions make by both bidder and auctioneer at any phase of the trading is shown. Figure 9 shows the deployment of the entire platform. It can be seen that web component, database component, solver component may reside on different machines. Figure 10 shows the Database design of the system.

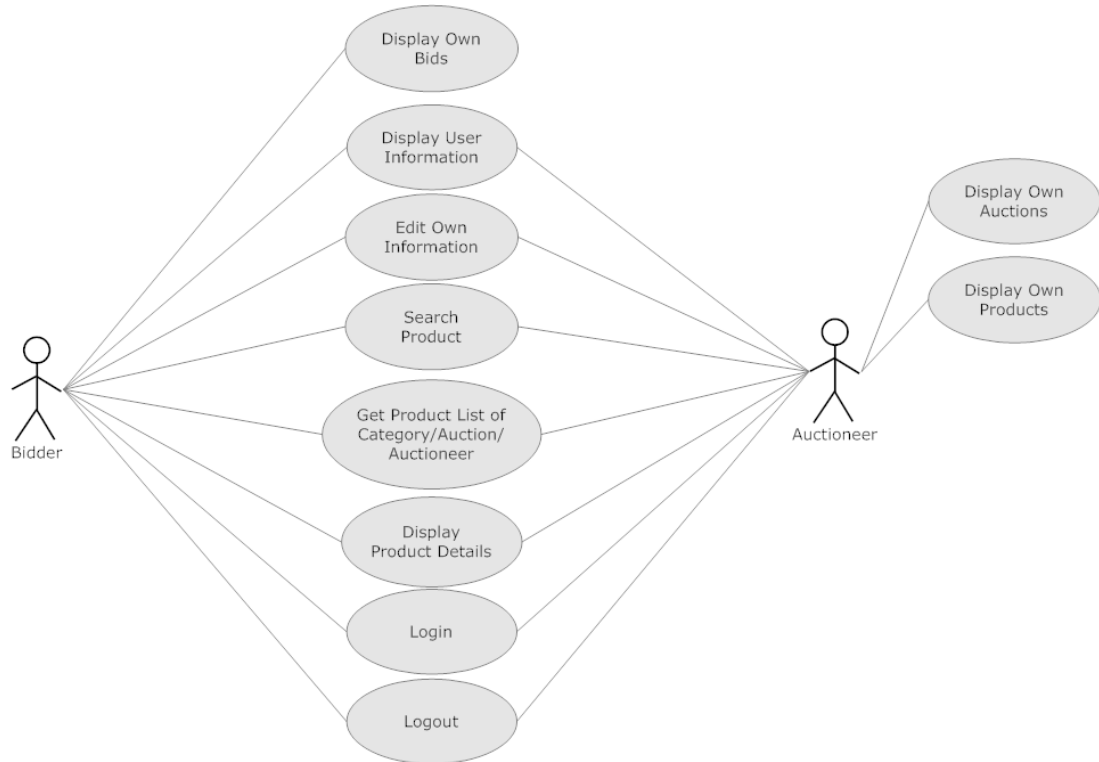


Figure 8. Use case diagram of general system usage

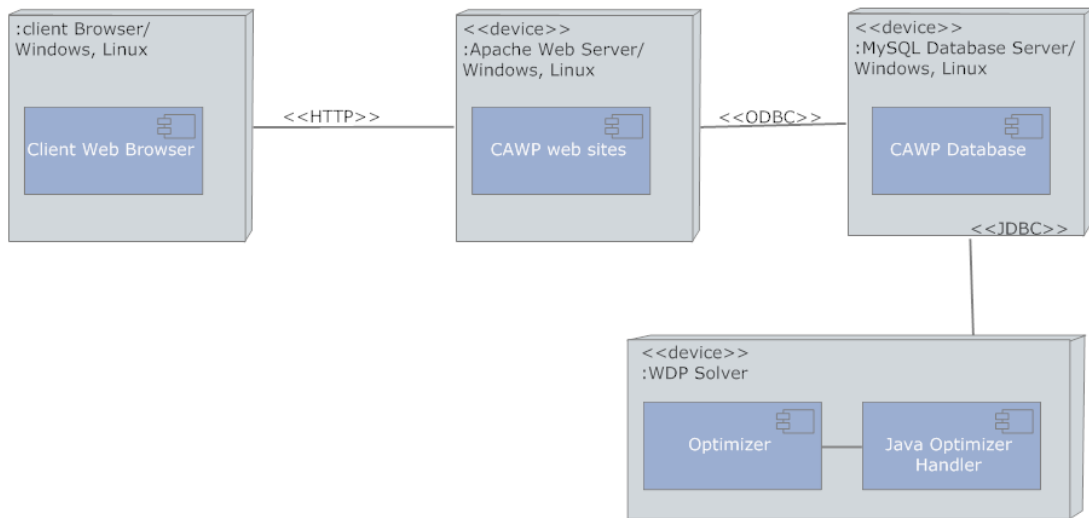


Figure 9. Deployment Diagram of the system

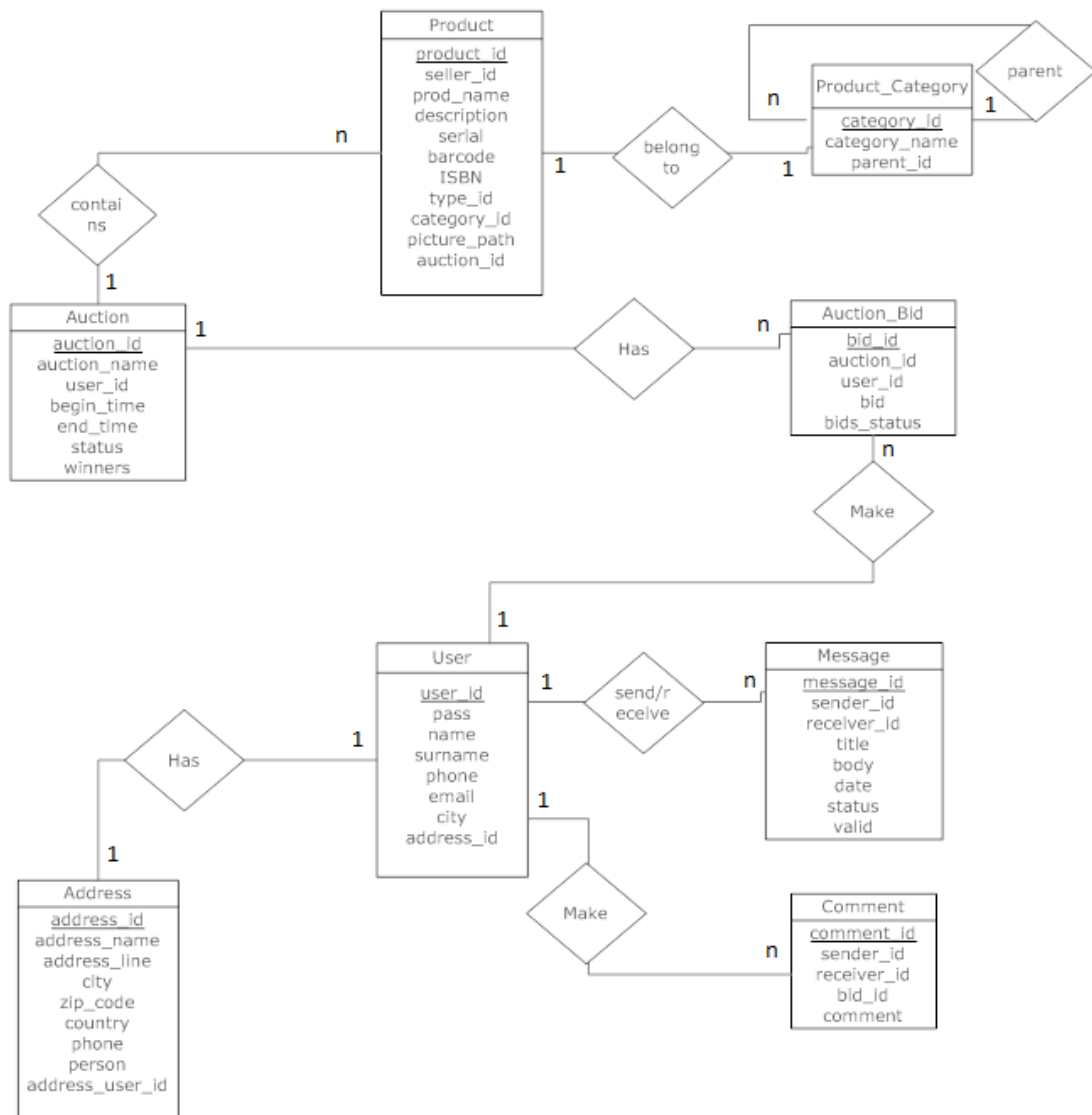


Figure 10. Database Entity Relation Model Diagram of CAWP

Database Design: Figure 10 shows the table structure and their relations in the database designed for CAWP. The database tables are driven from the real objects of the system, like users of the system, auctions and bids. Normally multi-valued attributes are not stored in single column of a table. But in our database the winners of an auction and bids are multi-valued attributes. This is because the nature of the WDP solver. WDP solver takes input as a list and returns winner ids as a list. Since this database is not transitionally active, we keep them as multi-valued attributes in order to reduce the construction times of that lists.

All technologies used in CAWP are open source. These open source technologies are free and independent from a specific operating system. These properties enable the deployment and distribution of the CAWP to most know operating systems, without any extra cost or effort.

The web component is written in PHP, we used Apache web server, but it can be deployed on any web server supporting PHP. A few example web site codes are given in Appendix D. Solver is a C program. Its source code is open so it can be compiled and run on both Windows and Linux systems. Also another program is needed to control the Solver, create and dispose the input and output files of the optimizer. A Java program is written for this purpose, this program should run on the same machine with the Solver. Java can again run on Windows and Linux system. Java Program code is given in Appendix C.

As a database management system MySQL, which has multi platform support, is chosen. Client side only needs a web browser to run the system. No additional program need to be installed on the client computer to use the CAWP. Clients won't know the operations held on the server side.

In order to use CAWP your computer should have proper installations of the following programs:

- Apache web server
- MySQL Database Management System
- Java Software Development Kit

- Eclipse Integrated Development Environment

To Install CAWP you have to follow these steps:

- Download the RAR file from <http://www.atilim.edu.tr/~icereci/cawp/cawp.rar>
- Extract the contents of the RAR file. Extraction will reveal three folders, webserver, database, and optimizer.
- Copy the contents of the web server file to the web server folder in your computer.
- Optimizer folder includes an eclipse Java project and C optimizer. Copy this folder to your workspace and create an eclipse project with the same name.
- Database folder contains the initial table structure needed by the CAWP. Use phpMyAdmin to import these database and tables to your computer.

Modification of CAWP:

- Web server, database and optimizer components can be deployed on the same computer as well as separate computers. All configurations in the CAWP are done assuming they are installed on the same computer. If you want to distribute them there are several things you should do.
 - In the web server there is a config.php file. That file contains database username and password and also points to the database address. Change them to the correct ones.
 - In the optimizer java program, the database address and username is also stated. Change them to your database address, username and password.
- After deployment there are modifications you can do to the CAWP. All of them are in a page which is in the MODIFICATION folder in the web server.

After you complete necessary modifications, delete MODIFICATION folder before deployment.

4.3 Business Protocol and Policies

4.3.1 Policies

Policies are crucial for any online e-commerce site. Otherwise you may be responsible for any unwanted situation. All users should agree to the user agreement and privacy policy of CAWP before registering to the system. These policies are taken and adapted from the privacy policy [34] used in eBay. These Policies are effective on August 09, 2009, for current users, and upon acceptance for new users.

See Appendix A. for the privacy policy of CAWP.

See Appendix B. for the user agreement of CAWP.

4.3.2 Business Protocol

Business Process of the entire system has been defined using UML diagrams in section 4.2. Business Protocols and Policies that are driven from the process are given below.

Business Protocol for the Auctioneer:

1. Register to the system.
2. Login to the system.
3. Create new product by giving it a name, category, description, serial number (optional), barcode (optional), ISBN number (optional), picture (optional).
4. Repeat step 3 until you create all your products in the system.
5. Pay the fixed amount of creating an auction.
6. Create a new auction by giving it a name, bidding period (3-7-10 days), and by selecting the product you want to put in this auction. The auction is

created after you submit all these information and it is instantly available for bidding.

7. If there are no bids placed on your auction user can delete it.
8. Wait for the bidding period to end. By that time you can see the bids placed on your auction, but they are anonymous to you.
9. After bidding period is over, you may check the site to see the status of your auction. Whether it is finished and waiting or being solved by the solver. After the winners are determined you will be notified with mail, you can also check the web site to see the results.
10. After bidding period is over, winning bidders should make payment in 3 days to the CAWP, the system will notify you when this payment is realized. You should make the shipment within 3 days after the payment is realized. If the payment is not done within this time period that means the goods are not sold to the bidder. You may put a complaint on them via the contact e-mail and you can comment on their public comments page, so other people will also know about it. After your products reach to the bidder and he approves the shipment money will be transferred from CAWP to your account after taking 10% commission.
11. Step 9 is realized for all the winning bids, after that your auction is finalized with its trading period.
12. At anytime you may start another auction with different products. The bidding and trading periods of different auctions can overlap.

Business Protocol of the Bidder:

1. User may wander in this site, check product, auction details without logging in to the system. You must be logged in before placing a bid to any group of items.
2. Find an item that you are interested and put it in your lot (similar to a shopping cart in e-commerce sites).
3. You may repeat step 2, as long as new items are in the same auction.

4. You may delete your lot anytime you want.
5. Place a bid for all the items in your lot. Minimum bid price which is the sum of the base prices of all the items in that lot is given to you. You cannot place any bid lower than the minimum bid price.
6. You can repeat these 5 steps as much you wish. You can have any number of bids at any given time.
7. After bid period is over and winners are determined you will be notified with email. You can also use the site to see the results.
8. If you are not one of the winning bidders, this auction is over for you. If you are one of the winning bidders you will continue with step 9.
9. You will pay for the bids you won in 3 days after the announcements of the winners. If you fail to make payment to CAWP in this period, you lose your winner rights over these products, and auctioneer may give bad public comments about you. If you are decided to be a malicious user your registration of the system can be cancelled. If you make the payment in time, auctioneer will have 3 days to ship the goods to you. If auctioneer fails to make shipment in 3 days, the trade is off and your money is refund by CAWP. You may write public comment about the auctioneer, so other people can see them before bidding his other auctions in the future.

4.4 Performance of the System

System performance depends on the performance of the technologies it uses. In [35] performance of the MySQL database system is explained. In [36] performance issues of Apache are discussed. Aside from these, the most important component that needs to perform well is the solver. Performance of the solver is comparatively represented in [32].

The performance of the CASS solver is tested. Computer used on these test has Intel® Core™2 Duo 2.66 GHz CPU, 4GB Ram, Windows XP Professional. We

created 10 different scenarios for every different number of goods and bids and Table 1 gives us the average completion time of the optimization in seconds. These test cases are generated with CATS [7] [27]. CATS generates well distributed non-dominant bids, which make the problem harder to solve. Thousands of people can use the system at the same time; both Apache and MySQL allow it. We limit the number of items that can be in the same auction to 30. Even if there are 5000 bids to a single auction, the solver still produces an answer within minutes. CAWP design lets more than one solver to run and solve problems at the same time, so if need to increase the throughput of the system you can always run more optimizer on different machines.

Below, Table 1 shows the completion time of the CASS WDP solver. Different sizes of goods and bids are given in the table. As a result, if the number of bids is small even the auctions with greater number can be solved quickly. But the main thing that effects the solution time is number of goods. When we keep the number of bids as 5000 and increase the number of goods from 30 to 50 the completion time of the problem increases from 151.86 seconds to 3713.24 seconds.

Table 1. Results of the CASS tests for variable problem sizes

Number of Goods	Number of Bids	Result in Seconds
10	100	0.002
10	1000	0.016
10	3000	0.21
30	100	0.003
30	1000	0.047
30	3000	12.84
30	5000	151.86
40	100	0.015
40	1000	6.552
40	3000	126.017
40	5000	177.721
50	100	0.020
50	1000	8.424
50	3000	3713.246

As seen in Table 2, increasing the number of concurrent solver running on a single machine reduces the overall performance of the solver. If aim is to increase the throughput of the system, concurrent solvers should run on different machines, single solver on a single machine is the best solution. Remember that Table 2 is constructed on a dual core machine, so two solvers run on different CPUs, but when we run four solvers, they must to share the same CPU, and it increases the completion time.

Table 2. Performance results for multiple solvers running on a single machine

Number of Solvers	Number of Goods	Number of Bids	Completion Time
2	30	3000	17.34
3	30	3000	23.87
4	30	3000	35.31

4.5 Risk Management

Every e-commerce site contains possible business and technical risks; they are inherited from the nature of trading and online systems. The aim of this section is to recognize possible risks which may arise using CAWP, and explaining cautions taken to prevent or minimize these risks.

4.5.1 Business Risks and Solutions

Some of the business risks of CAWP are listed below:

1. Identity theft: One person may give the CAWP information of another person, and try to cheat the system, steal from the bidders, or place false bids.
2. False products: Auctioneers may put products that are different than the description or they may create a product they don't have.
3. Fake Bidder: Bidding to any auction is free. So a person may create an account and bid to an auction without having intention to buy the products.
4. Auctioneer may not send a product that is won by a bidder.
5. A bidder may not pay a won bid.

Solutions to the business risks listed above:

1. The solution proposed by CAWP is to get a fixed amount of money from auctioneers before creating their auctions. The payment will reveal the true identity of the person and will prevent them to create accounts under false names.
2. Since CAWP takes the payment from the bidder first, after bidder validates the products sent by the auctioneer transfers the money from the auctioneer, an auctioneer gains nothing from putting false products to the system.
3. For the risks 3, 4, 5 we have a comment system in the CAWP. If a bidder or an auctioneer does not act in a good faith he will get negative comments from both CAWP and the other person in the trade. So the following trades of that person will carry the comments from the previous trades.

4.5.2 Technical Risks and Solutions

Some of the technical risks of CAWP are listed below:

1. Anonymity of the bidders and the bids.
2. Deleting, editing other peoples products, auctions or user information.
3. Network errors and site unavailability.
4. Problems during payment, loss of payments, over paying.
5. Error occurrence or non-optimality of the solution in winner determination and causing seller money loss.
6. Scalability
7. Performance issues due to concurrent solver running on a single machine.

Some proposed solutions:

1. Anonymity of the bidders and the bids are kept in the system. Only winner bids are displayed after an auction. User id is displayed with the bid, so if a

user real identity cannot be perceived from user name they are totally anonymous to other people on the system.

2. All the deletion/update/insertion operations in all the pages are done after logged user is valid and operation is related to him. So no one can reach and change a portion of data that isn't about themselves.
3. Site can crash and all the current auctions may be delayed or even canceled, these are stated in the user agreement. If a user accepts to use the system, he also accepts the integrity of the system cannot be trusted. If network failure happens or server is down, after things turn to normal CAWP continues with the remaining time of the auctions, if an auction is finished during that period, that auction is sent to the solver.
4. Payments are done over a trusted third party. History of all payment is kept. If there are any complaints about payment, it can be manually checked and corrected.
5. WDP Solver is optimization software, but like all software it may include bugs, or it can crash at any time. So if any misleading result is generated from the solver, CAWP is not responsible from it. It is clearly stated in the user agreement.
6. The solver is scalable. Because you can run multiple solvers on multiple computers and they may work concurrently. But you can not scale up or down of a single solvers performance.
7. Performance may decrease due to concurrent usage of solvers in a single machine. If there are not enough machines to run solver on them, WDP solver speed will decrease. Since on a single solver, only a single problem can be solved at a time, other finished auctions may get their winner solution late. In order to avoid this, one can trace the solvers performance through solvers output and add extra solvers on different machines when needed.

4.6 Some Limitations about the CAWP

The WDP solver program CASS, uses a lot of central processing unit powers. Running more than one instances of CASS on a single machine reduces the performance of the CASS solvers. There is no upper limit of solvers which can be on a single machine.

In order to serve an online community, solvers must be distributed on a grid and they should be remotely controllable. In CAWP you have to run the program on every machine they are controlled manually.

CAWP is not highly customizable. Customization effort mostly requires extra coding or updating of a code segment, which requires knowledge of the programming language. Customization component still requires some enhancements.

Current version of CAWP does not support any punishment/awarding mechanism about its usage. However, one can write down comments about their trading experiences. By means of a punishment mechanism user accounts can be suspended or cancelled automatically. Through a well designed awarding mechanism, trust among the users can be established better.

CHAPTER 5

CASE STUDIES

These case studies are chosen because they fit to the Combinatorial Auctioning mechanism. It is more logical to sell items which are complementary in a CA. Because people may want both items at the same time and may value them together much more sum than their separate values. This means that auctioneer will gain more money selling his items in a CA, and it makes the bidder to win which gives most value to the items.

5.1 Custom Built Furniture Auction

In this case study, there is a hypothetical carpenter who is making custom furniture. All the furniture he built is unique. He builds groups of furniture and categorizes them as breakfast, dining, bedroom and living room. For a customer who wants to decorate their house with this furniture, combination of furniture is more valuable oppose to a single one. Since there is no other copies of the items created here, the people who values most to the furniture will buy combinations of them and create a decorative look for their houses.

First we downloaded the source code from <http://www.atilim.edu.tr/~icereci/cawp/cawp.rar>. Then we create an eclipse project and put contents of optimization folder in it. We also created the database of the system using one of the scripts under the database folder in the rar file. Database operations are held on phpMyAdmin interface. We copied the content of the

webserver folder to our web server folder in the machine. After passing these steps, the initial screen of CAWP is displayed in Figure 11.

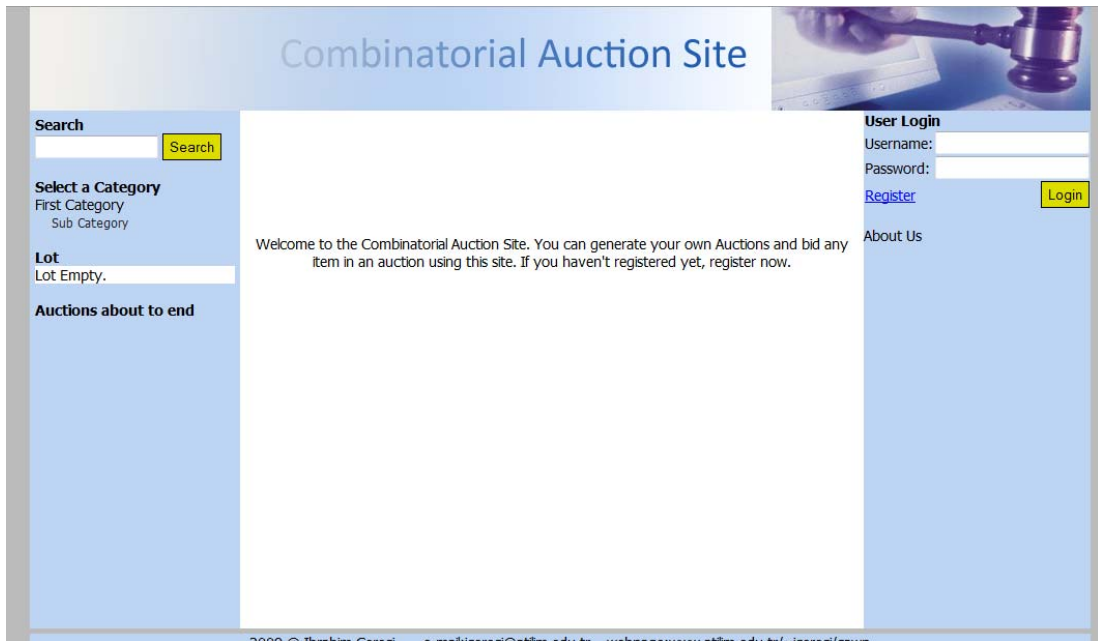


Figure 11. Initial Screen of the CAWP

In order to customize the system we use modification site included under MODIFICATION folder of our webserver. Figure 12 shows the modification window.

Owner Information	
Site Name	<input type="text" value="Shaker Furniture"/>
Site Owner Name	<input type="text" value="Ally Shaker"/>
Site Owner Company Name	<input type="text" value="Shaker Inc."/>
Contact Name	<input type="text" value="Alan Shaker"/>
Contact Email	<input type="text" value="alan@shaker.com"/>
Contact Phone	<input type="text" value="90123234345"/>
Owner Company Country	<input type="text" value="Turkey"/>
Owner Company Address	<input type="text" value="Dummy Address Line 1"/> <input type="text" value="Dummy Address Line 2"/>
Contact Company URL	<input type="text" value="www.shaker.com"/>
Labels	
Search	<input type="text" value="Search"/>
Select a Category	<input type="text" value="Categories"/>
Lot	<input type="text" value="Lot"/>
User Login	<input type="text" value="User Login"/>
Welcome Page	<input type="text" value="Welcome to Shaker Furniture Combinatorial Auction Site. In this site, you can bid to bundles of our homemade furniture."/>

Figure 12. Modification of CAWP parameters

The style of the CAWP can be changed by altering the Base.css which is under the webserver folder. Title picture can also be changed, it is a 1024x100 pixel picture which is under the webserver folder. Figure 13 shows the new look of the system after changing the css and the title picture.

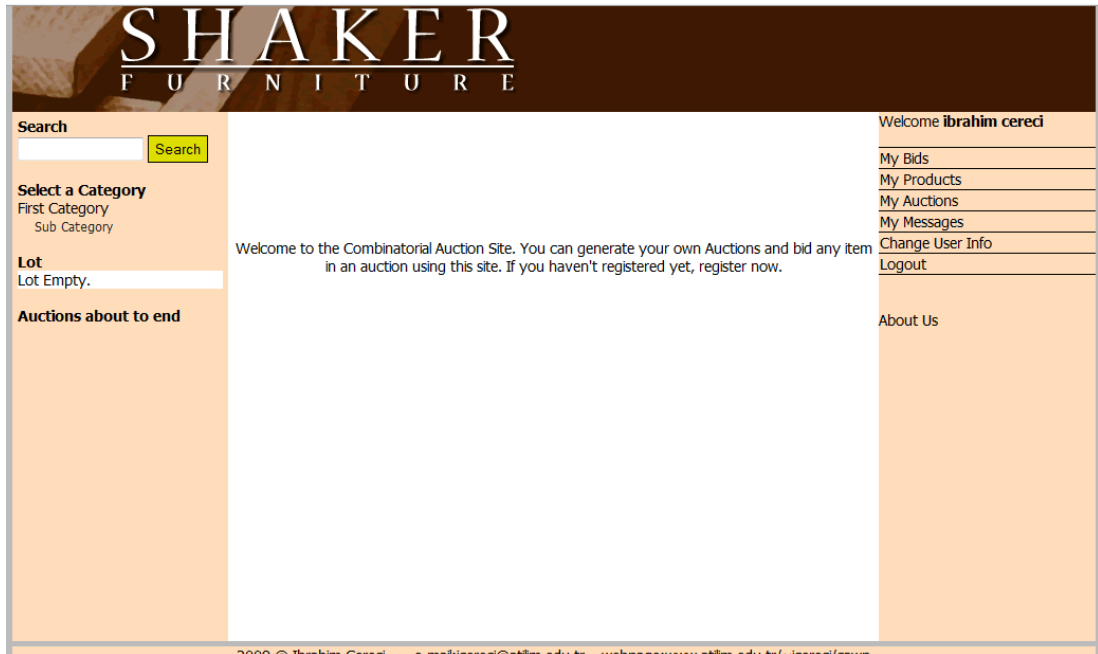


Figure 13. CAWP screen after changing title picture and CSS

Three accounts are created namely “shaker”, “jsmith”, and “murray”. Shaker is the account which sells the furniture in CAWP. Jsmith and murray are the customers which use CAWP system to buy furniture from shaker.

A scenario is generated over the first case study. In this scenario shaker adds 6 furniture to CAWP giving names, detailed information and pictures of the item.

Figure 14 shows addition of a product to the system.

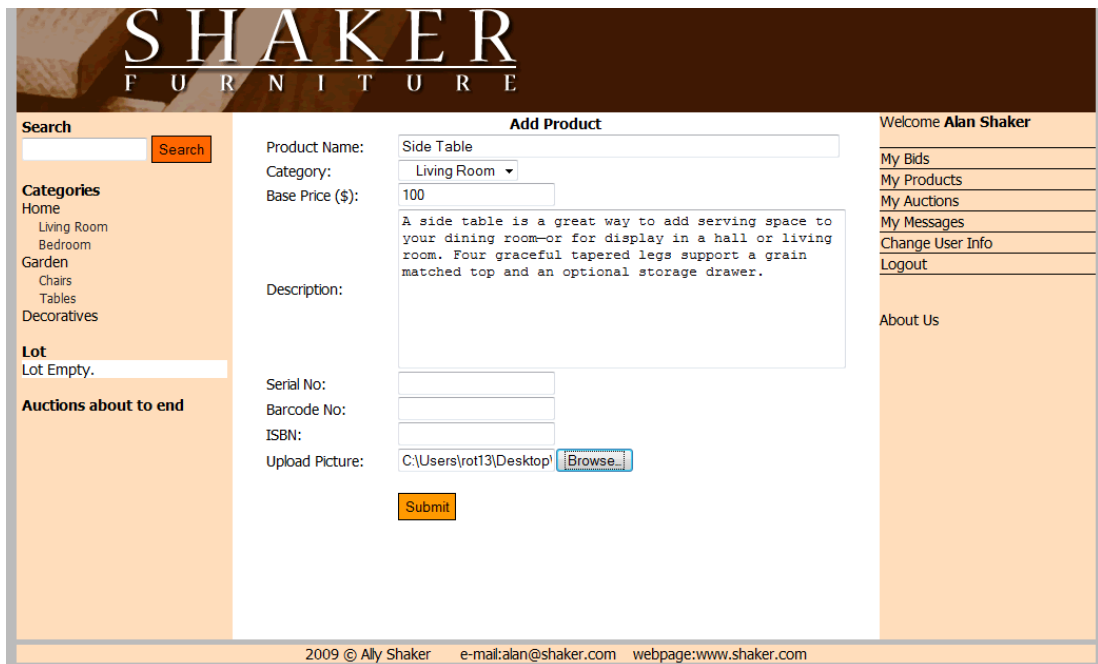


Figure 14. Adding a product to CAWP

After adding products, shaker creates an auction by giving auction a name, duration and adding previously created products to it. Figure 15 shows creation of an auction.

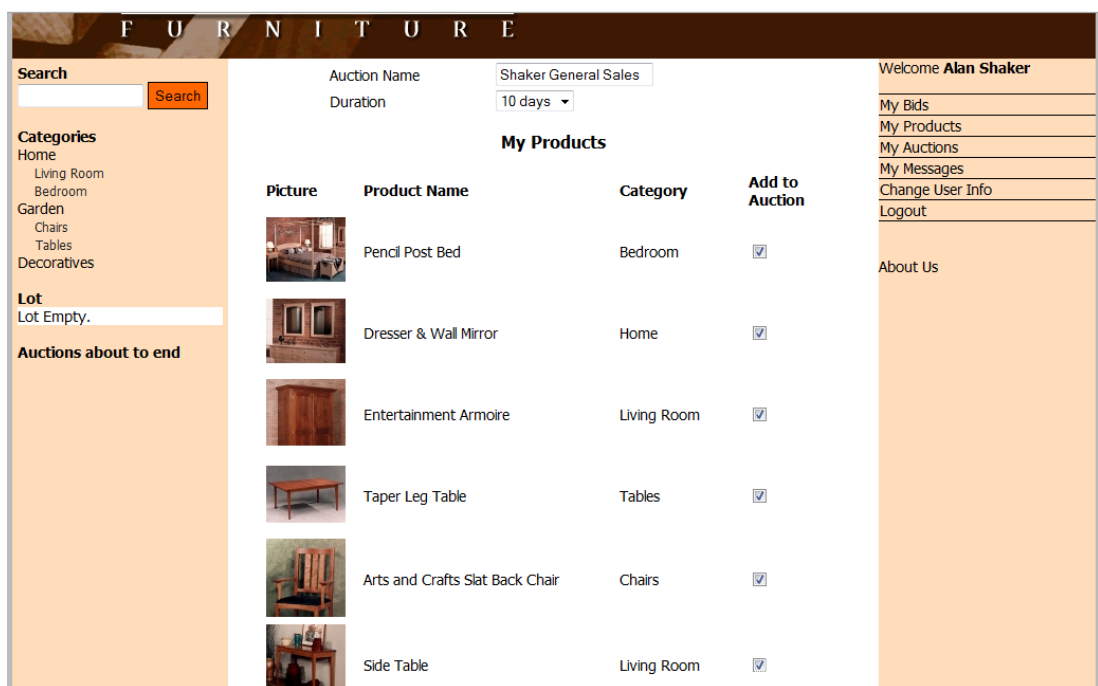


Figure 15. Creating an auction

Now there is an auction named Shaker General Sales and all the users of the system can bid bundles of its items. User jsmith bids 500\$ for Entertainment Armoire and Arts and Crafts Slat Back Chair. User murray places two different bids. First bid is a

480\$ bid for Entertainment Armoire and Taper Leg Table. Second bid is a 450\$ bid for Pencil Post Bed. All the bids should be higher than the base prices of the items, system does not avoid any invalid amount to be entered as value to the item.

The solver handler runs on the background always, it has a cyclic behaviour which checks for auctions that bidding period is passed. If solver handler finds such auction it automatically creates a input text file for solver to understand, run the solver, get the output file and import its results back to database. Figure 16 shows the solver solution on java console. Figure 17 and Figure 18 shows the CAWP pages which report users whether they win or lost a bid they placed.

```

public class Connect
{
    static Vector<String> v=new Vector<String>();
    static String winners="";
    static String winnersDB="";
    public static void main (String[] args)
    {
        boolean DEBUG=true;

        Connection conn = null;
    }
}

```

```

Connect (1) [Java Application] C:\Program Files\Java\jre6\bin\javaw.exe (11.Kas.2009 13:15:40)
-----New Cycle-----
Database connection established
Current Date Time : 2009-11-22 13:15:51
Auction 1 (Shaker General Sales) will be solved by me.
3 bids.
bid = 500 3 5
bid = 480 3 4
bid = 450 1
Winner Bids
0
2
1.txt deleted.
1.win deleted.
Winners inserted to the database
Current Date Time : 2009-11-22 13:15:52
Auction solved.
Database connection terminated
-----New Cycle-----
Database connection established
Current Date Time : 2009-11-22 13:16:02

```

Figure 16. Solver console report for solving WDP problem generated by CAWP

SHAKER FURNITURE

Search

Categories
Home
Living Room
Bedroom
Garden
Chairs
Tables
Decoratives
Lot
Lot Empty.
Auctions about to end

Auction Name	Auctioneer	Auction Ends in	Auction Status	Payment
Shaker General Sales	shaker	Auction Ended	Won	Pay

Welcome **John Smith**

- My Bids
- My Products
- My Auctions
- My Messages
- Change User Info
- Logout
- About Us

Figure 17. Result of the Auction of the given scenario for John Smith

SHAKER FURNITURE							Welcome Dave Murray
Auction Name	Auctioneer	Auction Ends in	Auction Status	Payment			
Shaker General Sales	shaker	Auction Ended	Lost				
Shaker General Sales	shaker	Auction Ended	Won	Pay			

Search:

Categories: Home, Living Room, Bedroom, Garden, Chairs, Tables, Decoratives

Lot: Lot Empty.

Auctions about to end

My Bids, My Products, My Auctions, My Messages, Change User Info, Logout, About Us

Figure 18. Result of the Auction of the given scenario for Dave Murray

5.2 Web Service Combinatorial Auction

CAWP can also be used when trading items are not goods but services. In this case-study, auctioned items are the usage of web services which run on grid computers. Web servers generally are public and can be usable if their description language is known. But those web services use system resources, if these web services are for heavy duty operations, they may need great amount of system resource and they can be restricted by other people using the same service. In this case study hourly usage rights of the web services are auctioned. Bidders can bid on combination of hours, they may value two consecutive hours more than two distinct hour. That is why this problem fits the CA area. In current version of CAWP these biddings are done by humans, but CAWP itself can be implemented as a web service and these bidding may be held between software agents automatically. In order to add CAWP web service capabilities, additional code writing is required.

Main difference between case study 1 and 2 is, auctions are daily in web service combinatorial auction. They start and finish at 0:00am. Auctioned services are related with the tomorrow's web service usage. After the trading period is over, auctioneer sends each winner web service password which is valid on hours won by the bidder.

CHAPTER 6

CONCLUSIONS

In this thesis, an online combinatorial auctioning platform CAWP is designed and implemented. The platform provides its users to create and to participate in combinatorial auctions without having to care about either the complexity of the WDP or its efficient solution. This system is more complex than a typical online auction web site, because there is an optimizer running and it deals with an NP-complete WDP problem, on the background. Different technologies combined together and created the platform, including server side web scripting, database technologies, solver handler, and the solver. All these technologies, except the solver have been created, originally. When we compare CAWP with another online CA system OICAS, CAWP is a more mature product than OICAS in terms of both its functionality and completeness. For example, CAWP works on a real web server using a real database management system. It is deployable and has e-commerce functionalities like user management and feedback mechanism through messaging. Furthermore, using the open source technologies enabled us to build the CAWP platform itself independent from any operating system.

Case studies are also generated to show that there are fields that really fit in combinatorial auctioning, and to the best of author's knowledge there is no other system like CAWP that enables a simple web user to sell his products in an online combinatorial auctioning environment.

CAWP lacks of some components an e-commerce platform should have. These can be implemented in future. Among them, CAWP does not work with real money. A

real third party payment can be included in order to make a sound e-commerce application. There should be strict rules which are embedded to system when a winner bidder does not wish to buy a good or real procurement of a product does not occur, users which cause these actions should be penalized automatically. A better WDP solver or a general purpose solver package can be adapted to the system instead of CASS, in order to increase the solver speed.

REFERENCES

- [1] S. Vries, R. Vohra: Combinatorial Auctions: A Survey. Can Be Found On: <http://www.cis.upenn.edu/~mkearns/teaching/cgt/combinatorial-auctions-survey.pdf> , In: December 1, 2000
- [2] R.H. Kwon, C. Lee, Z. Ma: An Integrated Combinatorial Auction Mechanism for Truckload Transportation Procurement. In: Technical Report, Mechanical and Industrial Engineering, the University of Toronto, December 2005.
- [3] M. O. Ball, L. Ausubel, F. Berardino, P.Cramton, G. Donohue, M. Hansen, K. Hoffman: Market-Based Alternatives for Managing Congestion at New York's LaGuardia Airport. In: Optimal Use of Scarce Airport Capacity, Proceedings of AirNeth Annual Conference, The Hague, April 2007
- [4] P. Milgrom: Putting Auction Theory to Work: The Simultaneous Ascending Auction. In: Journal of Political Economy, Vol. 108, No. 2, April 2000
- [5] C. Koboldt, D. Maldoom, R. Marsden: The First Combinatorial Spectrum Auction. In: DotEcon DP No. 03/01, May 2003
- [6] C. L. Jackson: Technology for Spectrum Markets. In: Ph.D. thesis. MIT, Department of Electrical Engineering and Computer Science. November 1976.
- [7] P. Cramton, Y. Shoham, R. Steinberg: An Overview of Combinatorial Auctions. In: ACM SIGecom Exchanges, Vol. 7, No. 1, December 2007.
- [8] G. Gottlob, G. Greco: On The Complexity of Combinatorial Auctions: Structured Item Graphs and Hypertree Decompositions. In: Electronic Commerce archive
Proceedings of the 8th ACM conference on Electronic commerce San Diego, California, USA Pages: 152 - 161 ,2007
- [9] Y. Bakos: The Emerging Role of Electronic Marketplaces on the web. In: Communications of the ACM 41(9), pg 35–42. , September 1998
- [10] Wolfram|Alpha: eBay as a financial entity. In Site: ["http://www02.wolframalpha.com/input/?i=eBay&a=*C.eBay-*Financial-](http://www02.wolframalpha.com/input/?i=eBay&a=*C.eBay-*Financial-)“, as of August 2009.
- [11] P. Klemperer: A Survey of Auction Theory. In Book: Auctions: Theory and Practice. Publisher: Princeton University Press, 2004

- [12] K. Omote: A Study on Electronic Auctions. In: Japan Advanced Institute of Science and Technology in partial fulfillment of the requirements for the degree of Doctor of Philosophy, March 2002
- [13] W. Vickrey: Counterspeculation, Auctions, and Competitive Sealed Tenders. In: The Journal of Finance, March 1961, pp. 9-37.
- [14] M. Wooldridge: Introduction to Multiagent Systems, John Wiley & Sons (2002)
- [15] P. Milgrom: Putting Auction Theory to Work, Cambridge University Press (2004)
- [16] J. Gallien, S. Gupta: Temporary and Permanent Buyout Prices in Online Auctions. In: Management Science, pp.814-833 Publication date: May 2007
- [17] S. Good, P. Lynn: The eBay Effect. In: Commercial Investment Real Estate, Publication date: Jan. 2007
- [18] T. Schoenherr, V. Mabert: Online Reverse Auctions: Common Myths versus Evolving reality. In: Business Horizons, pp.373-384, Publication date: Oct. 2007
- [19] C. Beam, A. Segev: Auctions on the Internet: A Field Study. In: 98-WP-1032, Fisher Center for Management and Information Technology, 1998
- [20] M. Kumar, S. Feldman: Internet Auctions. In: Proceedings of the 3rd conference on USENIX Workshop on Electronic Commerce - Volume 3 pp.5 ,1998
- [21] C. Boyd, W. Mao: Security Issues for Electronic Auctions. In: Seventh Annual Proceedings of Financial Cryptography, 2000
- [22] C. Koboldt: The First Combinatorial Spectrum Auction. In: DotEcon DP No. 03/01, May 2003
- [23] S. Lin, B. Chen, C. Liu, W. Soo: Web Service Allocations Based on Combinatorial Auctions and Market-based Mechanisms. In: Computer Supported Cooperative Work in Design, 2008. CSCWD 2008. 12th International Conference on Volume , Issue Page(s):452 - 458, 16-18 April 2008
- [24] W. E. Walsh, M. Wellman, F. Ygge: Combinatorial Auctions for Supply Chain Formation. In: ACM Conference on Electronic Commerce, 2000
- [25] K. L. Brown, Y. Fujishima, Y. Shoham: Taming the Computational Complexity of Combinatorial Auctions: Optimal and Approximate Approaches. In: Proceedings of the Sixteenth International Joint Conference on Artificial Intelligence, 1999

- [26] P. Cramton, Y. Shoham, and R. Steinberg: Combinatorial Auctions, The MIT press (2006)
- [27] K. L. Brown, M. Pearson, Y. Shoham: Towards a Universal Test Suite for Combinatorial Auction Algorithms. In: ACM Conference on Electronic Commerce, 2000
- [28] T. Sandholm, S. Suri, A. Gilpin, D. Levine: CABOB A Fast Optimal Algorithm for Combinatorial Auctions. In: Seventeenth International Joint Conference on Artificial Intelligence (IJCAI), 2001
- [29] A.H. Land, A.G. Doig: An automatic method of solving discrete programming problems. In: Econometrica 28 Publication date: July 1960
- [30] ILOG: ILOG CPLEX: High-performance software for mathematical programming and optimization. In Site: <http://www.ilog.com/products/cplex/>, as of August 2009
- [31] A. Holland, B. Sullivan: Super Solutions for Combinatorial Auctions. In Book: Recent Advances in Constraints, Publisher: Springer Berlin / Heidelberg, Volume 3419/2005, pp.187-200, 2005
- [32] K. L. Brown, Y. Shoham: Empirical Hardness Models: Methodology and a Case Study on Combinatorial Auctions. In: Journal of the ACM, Vol. 56, No. 4, Article 22, Publication date: June 2009
- [33] L. Fang, Y. Wang: OICAS: An Online Iterative Combinatorial Auction System. In: IEEE International Conference on Systems, Man and Cybernetics Publication date: Oct. 2005
- [34] eBay: eBay Privacy Policy. In Site: <http://pages.ebay.com/help/policies/privacy-policy.html>, as of August 2009
- [35] MySQL: MySQL Performance Benchmarks: Measuring MySQL's Scalability and Throughput. In: A MySQL® Technical White Paper March 2005
- [36] Apache: Apache Performance Notes: In Site: <http://httpd.apache.org/docs/1.3/misc/perf-tuning.html>, as of August 2009
- [37] eBay: eBay User Agreement. In Site: <http://pages.ebay.com/help/policies/user-agreement.html>, as of August 2009

APPENDIX A

APPENDIX

A. Privacy Policy

This privacy policy tells how we use and protect your personal information within the CAWP application.

Collection

You can browse CAWP without revealing your personal information to us. Once you login to the system and give us your information, you are not anonymous to us. If you give us your personal information that means you agree that we have to right to store that data.

We may collect and store following personal information:

- Email address, physical contact information, and some financial information, like credit card or bank account numbers.
- Transactional information based on your activities on the sites (such as bidding, buying, selling, item and other contents you generate in the system.
- Shipping, billing addresses and other information you provide to ship an item.
- Messages sent and received over CAWP system.
- IP addresses and standard web log information.

Marketing

We don't sell or rent your personal information to third parties without your permission. We may combine your information from third parties and use it to improve and personalize your services.

Use

Our primary purpose in collecting data is to provide you with a safe, smooth, efficient, and customized experience. You agree that we may use your personal information to:

- Provide the services and customer support you request
- Resolve disputes, collect fees, and troubleshoot problems
- Prevent potentially prohibited or illegal activities, and enforce our User Agreement
- Customize, measure and improve our services, content and advertising
- Tell you about our services and those of our corporate family, targeted marketing, service updates, and promotional offers based on your communication preferences
- Compare information for accuracy, and verify it with third parties

Our Disclosure of Your Personal Information

We may disclose personal information to respond to legal requirements, enforce our policies, respond to claims that a listing or other content violates the rights of others, or protect anyone's rights, property, or safety. Such information will be disclosed in accordance with applicable laws and regulations.

We may share your information with:

- Service providers under contract who help with our business operations, like web servers, payment authorities.
- Other third parties to whom you explicitly ask us to send your information.
- Law enforcement or other governmental officials, in response to a verified request relating to a criminal investigation or alleged illegal activity. In such events, we will disclose information relevant to the investigation, such as name, city, state, zip code, telephone number, email address, User ID history, IP address, fraud complaints, and bidding and listing history.

- Other business entities, should we plan to merge with or be acquired by that business entity.

Without limiting the above, in an effort to respect your privacy and our ability to keep the community free from bad actors, we will not otherwise disclose your personal information to law enforcement, other government officials, or other third parties without a subpoena, court order or substantially similar legal procedure, except when we believe in good faith that the disclosure of information is necessary to prevent imminent physical harm or financial loss or to report suspected illegal activity.

Information You Share on CAWP

Your user ID is displayed in the CAWP and it is connected to all your CAWP activity. Other people can see your bids, auctions, comments. So if you associate your name with your User ID, the people to whom you have revealed your name will be able to personally identify your CAWP activities.

If you access our sites from a shared computer or a computer in an internet café, certain information about you, such as your User ID, activity or reminders from CAWP, may also be visible to other individuals who use the computer after you.

Using Information from CAWP

CAWP enables you to share your information with other auctioneers and bidders in order to complete your trades. We cannot guarantee the privacy and security of your information; we encourage you to evaluate the privacy and security policies of your trading partner before sharing information with them.

No Spam, Spyware or Spoofing

We and our users do not tolerate spam. You are not licensed to add other CAWP users, even a user who has purchased an item from you, to your mailing list (email or physical mail) without their express consent. You may not use our communication tools to send spam or otherwise send content that would violate our User Agreement

Account Protection

Your password is the key to your account. Try to generate hard to guess and long password to increase the security level.

Accessing, Changing Your Personal Information

You can see, review and change most of your personal information by signing on to CAWP.

Security

Your information is stored in our database. That database is only accessible with a user name and password. We cannot promise, and you should not expect that your personal information or private communications will always remain private.

Third Parties

This document only states the data CAWP collected from you and usage of it. If you share your data with a bidder or auctioneer, they may have different policies of using your data; this document does not include them.

General

These privacy policies are subject to a change. You can always check the site for the latest privacy policy. The new privacy policies will take effect after a week from the announcement of the new privacy policies.

B. User Agreement

By using CAWP, you agree to the following terms described in this user agreement document. This agreement is taken and adapted from the privacy policy [37] used in eBay. This Agreement is effective on August 09, 2009, for current users, and upon acceptance for new users.

Using CAWP

While using CAWP sites, services you will not:

- post content or items in an inappropriate category or areas on our sites and services;
- violate any laws, third party rights or our policies.
- use our sites services or tools if you are not able to form legally binding contracts, are under the age of 18, or are temporarily or indefinitely suspended from using our sites, services or tools;
- fail to deliver payment for items purchased by you, unless the seller has materially changed the item's description after you bid, a clear typographical error is made, or you cannot authenticate the seller's identity;
- fail to deliver items purchased from you, unless the buyer fails to meet the posted terms, or you cannot authenticate the buyer's identity;
- manipulate the price of any item or interfere with other users' listings;
- circumvent or manipulate our fee structure, the billing process, or fees owed to CAWP;
- post false, inaccurate, misleading, defamatory, or libelous content (including personal information);
- take any action that may undermine our comment system.

- transfer your CAWP account (including feedback) and User ID to another party without our consent;
- distribute or post spam, unsolicited or bulk electronic communications, chain letters, or pyramid schemes
- distribute viruses or any other technologies that may harm CAWP, or the interests or property of CAWP users;
- copy, modify or distribute rights or content from the CAWP sites; or
- harvest or otherwise collect information about users, including email addresses, without their consent.

Abusing CAWP

Without limiting other remedies, we may limit, suspend or terminate our service and user accounts, prohibit access to our sites and its content, services and tools, delay or remove hosted content, and take technical and legal steps to keep users off the sites if we think that they are creating problems, possible legal liabilities, or acting inconsistently with the letter or spirit of our policies. We also reserve the right to cancel unconfirmed accounts or accounts that have been inactive for a long time, or to modify or discontinue CAWP sites, services or tools. CAWP does not guarantee that the integrity of the system. At anytime system can crash and valuable data may be lost, you create and participate auctions by knowing that they may never be concluded. Calculation of the winners in a Combinatorial Auction is a hard task, our chosen optimizer may not find the optimal solution, or any solution, that auctions may be cancelled and never pass to the trading phase.

CAWP Purchase Protection

In CAWP, sellers and buyers are responsible of realizing a smooth trading environment. We encourage buyers to work with sellers before placing a complaint about them.

Buyer Obligations:

- Buyers agree to file claims if sellers are not behaving due to their user agreements.
- For covered claims that meet the conditions and are not excluded, CAWP will refund the payment of the buyer.
- Buyers who CAWP believes are not acting in good faith, abusing the program or a seller, attempting to commit or committing fraud, or trying to unjustly benefit from the program may become ineligible for CAWP Purchase Protection. Buyers who become ineligible will be notified by CAWP prior to ineligibility. CAWP reserves the right to temporarily, indefinitely or permanently suspend the CAWP Purchase Protection program immediately if we suspect abuse, excessive claims, tampering, or interference with the proper working of the program.
- CAWP Purchase Protection doesn't cover certain types and categories of goods or services, false, abusive or illegitimate claims, or transactions where you have been or may be compensated from another third party.

Seller Obligations:

- Sellers agree to follow the program requirements as explained in the CAWP user agreement.
- All sellers with an open claim must work in good faith to resolve the claim, refund money and/or provide a replacement item, and accept a return of the item as explained by the CAWP Purchase Protection Policy.
- If a seller doesn't resolve a claim, the seller agrees to pay CAWP the amount paid by CAWP to the buyer in accordance with the CAWP Purchase Protection policy. CAWP will invoice the seller for these amounts and seller agrees to pay CAWP in accordance with the invoice terms.

Obligations when a buyer opens a claim alleging an item is counterfeit:

- Sellers shall work with the buyer in good faith during the resolution process to provide buyer with appropriate documentation or other assurances to

satisfy the buyer that the item is not counterfeit, if such information is available.

- If buyer and seller cannot agree that the item is not counterfeit, for covered claims that meet the conditions and are not excluded, buyers are required to send the item back to the seller. Cost of return shipping will be paid by the buyer, unless otherwise agreed upon by the buyer and seller.
- Covered claims that meet the conditions and are not excluded will count as a violation by the seller of our prohibited and infringing items policy.
- While buyers are not obligated to provide third party confirmation that an item is counterfeit in order to open a claim, in cases where there is written confirmation from the manufacturer that the item is counterfeit, or in additional circumstances where CAWP elects to exercise its discretion, CAWP may request the destruction of the item by an authorized third party and at CAWP's expense.

Fees and Services

Joining CAWP and bidding on listed items is free. We charge users for services such as creating an auction. When you create an auction, regardless the size or estimated value of the goods in the auction you pay a fixed amount. That amount may change at any time.

On the sale of the goods you are selling on an auction a final 10% commission is taken by the CAWP system over the winning bid prices. Each bid is paid separately and commission of that bid is taken during that transaction. This means if a bid is not realized, you don't have to give commission over it.

Content

For the convenience of sellers, we may offer catalogs of stock images, descriptions and product specifications, which are provided by third parties. While we try to offer reliable data, we cannot promise that the catalogs will always be accurate and up-to-

date. If you choose to include catalog content in your listings, you are responsible for ensuring that your listings are accurate and do not include misleading information.

Limitation and Liability

You will not hold CAWP responsible for other users' content, actions or inactions, items they list or their destruction of allegedly fake items. You acknowledge that we are not a traditional auctioneer. Instead, the sites are a venue to allow anyone to offer, sell, and buy just about anything, at anytime, from anywhere, in a variety of pricing formats and locations, such as stores, fixed price formats and auction-style formats. We are not involved in the actual transaction between buyers and sellers.

While we may help facilitate the resolution of disputes through various programs, we have no control over and do not guarantee the quality, safety or legality of items advertised, the truth or accuracy of users' content or listings, the ability of sellers to sell items, the ability of buyers to pay for items, or that a buyer or seller will actually complete a transaction or return an item. We do not transfer legal ownership of items from the seller to the buyer.

Further, we cannot guarantee continuous or secure access to our sites, services or tools, and operation of our sites, services or tools may be interfered with by numerous factors outside of our control. Accordingly, to the extent legally permitted, we exclude all implied warranties, terms and conditions. We are not liable for any loss of money, goodwill or reputation, or any special, indirect or consequential damages arising, directly or indirectly, out of your use of or your inability to use our sites, services and tools.

C. Optimizer Handler written in Java

Optimizer handler is a java program which consists of two classes

```
//-----Connect.java class-----
import java.sql.*;
import java.text.DateFormat;
import java.text.SimpleDateFormat;
import java.util.Vector;
import java.io.*;

public class Connect
{
    static Vector<String> v=new Vector<String>();
    static String winners="";
    static String winnersDB="";
    public static void main (String[] args)
    {
        boolean DEBUG=true;

        Connection conn = null;
        FileOutputStream fout;
        while(true)
        {
            try{
                Thread.currentThread();
                Thread.sleep(10000);//sleep for 10sec
                System.out.println("-----New Cycle-----");
            }
            catch(Exception ie){
            }
        }
        try
        {
            String userName = "root";
            String password = "mysql";
            String url = "jdbc:mysql://localhost/thesis";
            Class.forName ("com.mysql.jdbc.Driver").newInstance ();
            conn = DriverManager.getConnection (url, userName, password);
            if(DEBUG)
            {
                System.err.println ("Database connection established");
            }
        }
    }
}
```

```

    }

    Statement stmt = conn.createStatement();
    ResultSet rs;
    ResultSet rs3;

    int inserted, solveUpdate,insertWinners,updateToSolved;
    int prodAuctClear;

    DateFormat dateFormat = new SimpleDateFormat("yyyy-MM-dd");
    java.util.Date date = new java.util.Date();
    //System.out.println("Current Date Time : " + dateFormat.format(date));

    DateFormat dateFormat2 = new SimpleDateFormat("yyyy-MM-dd
HH:mm:ss");
    java.util.Date date2 = new java.util.Date();
    System.out.println("Current Date Time : " + dateFormat2.format(date2));

    Statement stmt2 = conn.createStatement();
    rs = stmt.executeQuery("SELECT * from auction where status=0 and
end_time <= '"+dateFormat.format(date)+"'");
    boolean flag=true;
    while ( rs.next()&& flag==true )
    {
        inserted = stmt2.executeUpdate("update auction set `status` = '1'
where auction_id="+rs.getString("auction_id"));
        if(inserted>0)
        {
            flag=false;
            System.out.println("Auction
"+rs.getString("auction_id")+ "("+rs.getString("auction_name")+") will be solved by
me.");
        }

        Statement stmt3 = conn.createStatement();
        rs3 = stmt3.executeQuery("SELECT * from auction_bid where
auction_id="+rs.getString("auction_id"));

        rs3.last();
        int bidnumber = rs3.getRow();
        if(bidnumber>0)
        {
            if(DEBUG)
            {
                System.out.println(bidnumber+" bids.");
            }
        }
        rs3.beforeFirst();
        String filename=rs.getString("auction_id")+".txt";
        fout = new FileOutputStream (filename);
        new PrintStream(fout).println ("goods 999");
    }

```



```

        new PrintStream(fout).println ("bids "+bidnumber);
        new PrintStream(fout).println ("dummy 0");

int bidid=0;

while ( rs3.next() )
{
    new PrintStream(fout).println (bidid+" "+rs3.getString("bid")+" #");
    if(DEBUG)
    {
        System.out.println("bid = "+rs3.getString("bid"));
    }

    bidid++;
}
fout.close();
try {

    Process p =Runtime.getRuntime().exec("cmd /c start cass.exe -z -i
"+rs.getString("auction_id")+ ".txt");
    p.waitFor();
}
catch (Exception err) {
    err.printStackTrace();
}

System.err.println("Winner Bids");
Thread.sleep(1000);
FileInput.readFile(rs.getString("auction_id")+ ".win");

Thread.sleep(100);
File f= new File(rs.getString("auction_id")+ ".txt");
while(f.exists())
{
    boolean success = f.delete();
    if(success)
    {
        if(DEBUG)
            System.out.println(rs.getString("auction_id")+ ".txt
deleted.");
    }
}

File f2 = new File(rs.getString("auction_id")+ ".win");
while(f2.exists())
{
    boolean success = f2.delete();
    if(success)

```

```

        {
            if(DEBUG)
                System.out.println(rs.getString("auction_id")+".win
deleted.");
        }
    }

    for(int i=0;i<v.size();i++)
    {
        rs3.absolute(Integer.valueOf(((String)v.get(i)))+1);
        winnersDB=winnersDB+" "+rs3.getString("bid_id");
    }

    Statement stmt5 = conn.createStatement();
        //insertWinners = stmt5.executeUpdate("INSERT INTO
auction_result ( `auction_id` , `winners` ) VALUES ( '"+rs.getString("auction_id")+"
','"+winnersDB+" )" );
        insertWinners = stmt5.executeUpdate("update auction
set `winners` = '"+winnersDB+" where auction_id='"+rs.getString("auction_id"));

        if(insertWinners>0)
        {
            if(DEBUG)
                System.out.println("Winners inserted to the database");
            Statement stmt6 = conn.createStatement();
            updateToSolved = stmt6.executeUpdate("update auction set `status`
= '2' where auction_id='"+rs.getString("auction_id"));
            if(updateToSolved>0)
            {
                DateFormat dateFormat3 = new SimpleDateFormat("yyyy-
MM-dd HH:mm:ss");
                java.util.Date date3 = new java.util.Date();
                System.out.println("Current Date Time : " +
dateFormat3.format(date3));
                System.out.println("Auction solved.");
            }
        }
    }

    winners="";
    winnersDB="";
    v=new Vector<String>();

}
else
{

```

```

        System.out.println("No bid was made to the auction.");
        Statement stmt4 = conn.createStatement();
        solveUpdate = stmt4.executeUpdate("update auction set `status` =
'3' where auction_id="+rs.getString("auction_id"));
        if(solveUpdate>0)
        {
            System.out.println("Auction solved.");
            Statement stmtx = conn.createStatement();
            prodAuctClear = stmtx.executeUpdate("update product set
`auction_id` = '0' where auction_id="+rs.getString("auction_id"));
            if(prodAuctClear>0)
            {
                System.out.println(prodAuctClear+" items deleted from
auction.");
            }
        }
    }
}

if(flag) //if there are no auction to be solved at the moment
{
    System.out.println("No auction to solve");
}

}
catch (Exception e)
{
    System.err.println ("Cannot connect to database server");
    System.err.println (e);
}
finally
{
    if (conn != null)
    {
        try
        {
            conn.close ();
            if(DEBUG){
                System.err.println ("Database connection terminated");
            }
        }
        catch (Exception e)
        {

```



```
    }  
    return 0;  
  }  
}
```

D. Web Site Code Samples

These code samples are just a minor part of the web site, We put the index page which all pages are directed from, product listing page, style of the web page base css file, product categories page, and a form page which is change user information page.

There are 49 web pages in the CAWP. All of them are implemented by Ibrahim Cereci.

```
-----index.php-----
<?php
session_start();
include_once("config.php");
?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-9" />
<title>Combinatorial Auction Site</title>
<link rel="stylesheet" type="text/css" href="base.css" />
</head>
<body>
<div align="center">
<table class="main" width="1024px">
<tr>
<td class="top" colspan="3" height="100px"></td>
</tr>
<!--
<tr>
<td class="thin" colspan="2">Breadcrumbs</td>
<td class="thin">Search</td>
</tr>
-->
<tr>
<td class="left" width="20%" valign="top">
<?php
include_once("search.php");
echo "<br />";
include_once("categories.php");
echo "<br />";
include_once("lot.php");
echo "<br />";
include_once("auction.php")
?>
```

```

</td> <!--Center is the part where product categories are displayed-->
<td class="center" width="60%" height="500px" valign="top"> <!--Center is the
main page things happen-->
<?php
if($_GET["pid"]=="main")
{
    include_once("main.php");
}
else if($_GET["pid"]=="login")
{
    include_once("login.php");
}
else if($_GET["pid"]=="logout")
{
    include_once("logout.php");
}
else if($_GET["pid"]=="register")
{
    include_once("register.php");
}
else if($_GET["pid"]=="doregister")
{
    include_once("doregister.php");
}
else if($_GET["pid"]=="displayproduct")
{
    include_once("display_product.php");
}
else if($_GET["pid"]=="addtolot")
{
    include_once("add_to_lot.php");
}
else if($_GET["pid"]=="listproducts")
{
    include_once("list_products.php");
}
else if($_GET["pid"]=="addpicture")
{
    include_once("add_picture.php");
}
else if($_GET["pid"]=="doaddpicture")
{
    include_once("do_add_picture.php");
}
else if($_GET["pid"]=="myproducts")
{
    include_once("my_products.php");
}
else if($_GET["pid"]=="addproduct")

```

```

{
    include_once("add_product.php");
}
else if($_GET["pid"]=="doaddproduct")
{
    include_once("do_add_product.php");
}
else if($_GET["pid"]=="changeuserinfo")
{
    include_once("change_user_info.php");
}
else if($_GET["pid"]=="dochangeuserinfo")
{
    include_once("do_change_user_info.php");
}
else if($_GET["pid"]=="dochangeuseraddress")
{
    include_once("do_change_user_address.php");
}
else if($_GET["pid"]=="dodeleteproduct")
{
    include_once("do_delete_product.php");
}
else if($_GET["pid"]=="editproduct")
{
    include_once("edit_product.php");
}
else if($_GET["pid"]=="doeditproduct")
{
    include_once("do_edit_product.php");
}
else if($_GET["pid"]=="myauctions")
{
    include_once("my_auctions.php");
}
else if($_GET["pid"]=="mybids")
{
    include_once("my_bids.php");
}
else if($_GET["pid"]=="displaybid")
{
    include_once("display_bid.php");
}
else if($_GET["pid"]=="deletebid")
{
    include_once("delete_bid.php");
}
else if($_GET["pid"]=="displayseller")
{

```



```

        include_once("display_seller.php");
    }
else if($_GET["pid"]=="listauctions")
{
    include_once("list_auctions.php");
}
else if($_GET["pid"]=="displayauction")
{
    include_once("display_auction.php");
}
else if($_GET["pid"]=="lotdetail")
{
    include_once("lot_detail.php");
}
else if($_GET["pid"]=="emptylot")
{
    include_once("empty_lot.php");
}
else if($_GET["pid"]=="makebid")
{
    include_once("make_bid.php");
}
else if($_GET["pid"]=="newauction")
{
    include_once("new_auction.php");
}
else if($_GET["pid"]=="createauction")
{
    include_once("create_auction.php");
}
else if($_GET["pid"]=="makepayment")
{
    include_once("make_payment.php");
}
else if($_GET["pid"]=="domakepayment")
{
    include_once("do_make_payment.php");
}
else if($_GET["pid"]=="displayabid")
{
    include_once("display_a_bid.php");
}
else if($_GET["pid"]=="deleteauction")
{
    include_once("delete_auction.php");
}
else if($_GET["pid"]=="dosearch")
{
    include_once("do_search.php");
}

```