(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
10 May 2001 (10.05.2001)

PCT

(10) International Publication Number
WO 01/33401 A2

(51) International Patent Classification[7]: G06F 17/00

(21) International Application Number: PCT/US00/30323

(22) International Filing Date:
3 November 2000 (03.11.2000)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
60/163,245 3 November 1999 (03.11.1999) US
09/703,712 2 November 2000 (02.11.2000) US

(71) Applicant *(for all designated States except US)*: **ADAPTIVE TRADE, INC.** [US/US]; 6 Montgomery Village Avenue, 5th Floor, Gaithersburg, MD 20879 (US).

(71) Applicants and
(72) Inventors: **BRODSKY, Alex** [US/US]; 222 Rollins avenue, Rockville, MD 20852 (US). **ZELIVINSKI, Stanislav** [US/US]; 11505 Brandy Hall Lane, Gaithersburg, MD 20850 (US).

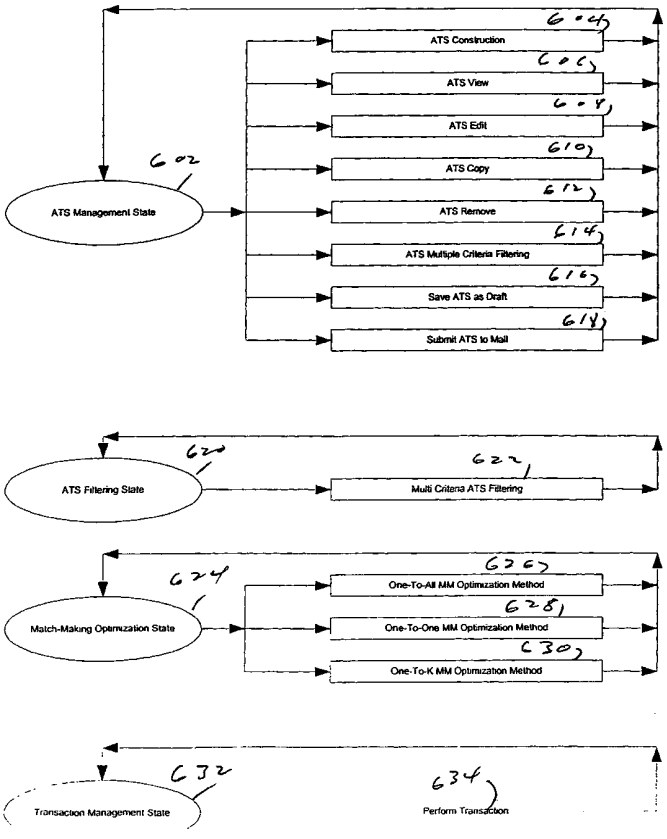
(72) Inventors; and
(75) Inventors/Applicants *(for US only)*: **KATZ, Marcel** [US/US]; 1710 Wilmart Street, Rockville, MD 20852 (US). **GOZHANSKY, Alan** [US/US]; 11505 Brandy Hall Lane, Gaithersburg, MD 20850 (US). **KARPISHPAN, Sonya** [US/US]; 259 Congressional Lane, #502, Rockville, MD 20852 (US).

(74) Agents: **COHEN, Herbert** et al.; Blank Rome Comisky & McCauley LLP, 900 17th Street, N.W., Suite 1000, Washington, DC 20006 (US).

(81) Designated States *(national)*: AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, BZ, CA, CH, CN, CR, CU, CZ, DE, DK, DM, DZ, EE, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, MZ,

[Continued on next page]

(54) Title: ELECTRONIC MALLS AND AUCTIONS BASED ON ADAPTIVE TRADE SPECIFICATIONS



(57) Abstract: An electronic marketplace based on adaptive trade specifications (ATS) is disclosed. The system allows users to perform optimized electronic transactions over the Web. The invention involves a method and system for providing match making among traders in an electronic marketplace according to item specifications provided by those traders via adaptive trade specifications. Given a trader's objective, the system will recommend the optimal transaction for this trader. This optimal transaction specifies items and quantities specified by other traders within the electronic marketplace. The invention describes methods for ATS-based trading in an electronic marketplace in general and in electronic malls and electronic auctions in particular.

WO 01/33401 A2

NO, NZ, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TR, TT, TZ, UA, UG, US, UZ, VN, YU, ZA, ZW.

(84) **Designated States** *(regional)*: ARIPO patent (GH, GM, KE, LS, MW, MZ, SD, SL, SZ, TZ, UG, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE, TR), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GW, ML, MR, NE, SN, TD, TG).

Published:

— *Without international search report and to be republished upon receipt of that report.*

For two-letter codes and other abbreviations, refer to the "Guidance Notes on Codes and Abbreviations" appearing at the beginning of each regular issue of the PCT Gazette.

Electronic Malls and Auctions based on Adaptive Trade Specifications

Background – Cross Reference to Related Applications

The present application claims the benefit of U.S. Provisional Application No. 60/163,245, filed November 3, 1999. Related subject matter is set forth in U.S. Provisional Application No. 60/161,355, filed October 26, 1999, and 60/163,243, filed November 3, 1999. The disclosures of all of the just-cited provisional applications are hereby incorporated by reference in their entireties into the present disclosure.

Background – Field of Invention

This invention relates to conducting trade activities within an Electronic Marketplace where the participants are represented through Adaptive Trade Specifications (ATS). An Electronic Marketplace is comprised of Electronic Malls and Electronic Auctions.

Background – Discussion of Prior Art

Current electronic marketplaces, specifically electronic malls and electronic auctions, will be reviewed.

Electronic Mall

Yahoo.com provides a list of more than one thousand electronic or virtual malls (see http://dir.yahoo.com/Business_and_Economy/Companies/Retailers/Virtual_Malls/). All these online malls provide only limited decision support tools. The existing Electronic Malls have mainly the following characteristics:

- Provide mainly business-to-consumer services.
- Target a single buyer at a time, for each atomic transaction.
- Narrow the buyer's selection process through a limited predefined set of filtering and navigation tools:
  - Generic string matching search engine.
  - Classification of items by categories and subcategories.
  - Ranking according to well-known predefined market criteria.
  - Ranking based on previous customers' feedback.
  - Predict customers' preferences based on past customers' buying patterns and profiles.

Electronic Auction

The existing Electronic Auctions offer regular *direct* and *reverse-auction* methods. They have mainly the following characteristics:

- The auctioned bundle consists of one or many items of a *single specific type*. There is no auction bundle that includes several types of items, which can be split among different bidders.
- Many bidders compete against a *single* fixed auctioned bundle.
- Bids consist of a *monetary* amount.
- The decision on what to bid, given a wide range of similar choices, is left entirely to the potential bidder who uses a selection process through a limited predefined set of filtering and navigation tools:
  - Generic string matching search engine.
  - Classification of items by categories and subcategories.
  - Ranking according to well-known predefined market criteria.
  - Ranking based on previous customers' feedback.
  - Predict customers' preferences based on past customers' buying patterns and profiles.

Electronic Marketplace

General problems in the existing models of Electronic Marketplace include also:

- Lack of massive support for complex business-to-business transactions. Most models are geared toward business-to-consumer and consumer-to-business transactions. There are no online, Web-based models for multiple-business to multiple-business transactions.
- Traders' dichotomy of scale. A small trader may be unable to satisfy the requirements of a large trader because the amount of assets (goods or money) offered or sought by a single small trader is incompatible with the conditions of a large trader. There is no real-time, automated, and opaque mechanism to create aggregations of small traders for a single transaction vis-à-vis a

large trader. The same is true when decomposition of a large trading basket is needed to satisfy the requirements of a single small trader.

- Lack of dynamic supply and demand feedback. There are no dynamic, real-time supply and demand gauging mechanisms that perform analysis across all the participants within the Electronic Marketplace - against specific traders' objectives.
- Lack of extensibility. When new types of commerce (e.g. auctions, reverse-auctions) need to be added to a predefined market paradigm (e.g. an electronic mall) extensive re-engineering of the current market paradigm is required. The same is true when new vertical markets are added to an existing set of vertical markets supported by a certain Electronic Marketplace.

Summary of the Invention

In light of the above, it will be apparent that a need exists in the art to overcome the above-noted deficiencies of the prior art. It is therefore an object of the invention to provide a system and process by which various traders that take part in the Electronic Marketplace perform optimal trade transactions.

It is another object of the invention to provide an Adaptive Trade Specification model for usage in the Electronic Marketplace.

It is a further object of invention to integrate into the Electronic Marketplace an ATS Based Match Making (MM) and Optimization automated method that can find optimal trade transaction for the traders in the Electronic Marketplace in scenarios that include but are not limited to, business-to-business and multiple-businesses to multiple-businesses.

To achieve the above and other objects, an Electronic Marketplace is comprised of Electronic Malls or Electronic Auctions, or a combination thereof. Traders will be represented in the Electronic Mall through Adaptive Trade Specifications (ATS). Given a trader's ATS, the invention provides an automated process that recommends specific transactions with other

traders' ATS's, that are mutually agreeable with, and optimize the objective of, the trader's ATS (e.g. minimal price, maximal profit, etc.). More specifically, the invention proposes the inclusion of the following elements in the infrastructure of the Electronic Marketplace: Adaptive Trade Specifications (ATS), ATS-based Match Making and Optimization, and Wizards for various types of traders. These are described bellow as a summary excerpt from one of the above-cited applications:

- **Adaptive Trade Specification (ATS) Model.** Adaptive Trade Specification (ATS), which is a formal mathematical description of trader's objective and constraints. ATS constraints include restrictions (on quantities, prices, totals, profits, revenues etc.) that must be satisfied to perform an optimal transaction, and the interconnection between various business parameters (such as profit, quantities, prices and costs). The core of each ATS is a specification of "items" the trader offers to GIVE as well as "items" to TAKE in return. For example, a procurement organization may offer to GIVE the "item" *money* and wants to TAKE items of *office supply*. An office equipment supplier may have an ATS, in which all its catalog appears as GIVE items, and money as the only TAKE item. Whereas, a manufacturer may have an ATS, in which all of its products appear as GIVE items, all raw materials and money (i.e., revenues for its products) as TAKE items. ATS is adaptive in that various numeric parameters such as quantities of items, prices, profit, revenue, totals etc. are not fixed, but could vary, provided that they satisfy the ATS constraints. Item specifications in an ATS are also constraint-based and not fixed. For example, an ATS of a trader may include, as one of the TAKE item specifications, a hard disk that has at least 12 GB capacity and is compatible with a G7305E mother board; no exact model and vendor is necessary.
- **ATS-based Match Making and Optimization (MMO) Automated Method.** Given a trader's ATS, the MMO method recommends specific transactions with other traders (i.e., against their ATS's) that are mutually agreeable and optimize the objective of the trader's ATS (e.g., minimal price, maximal profit etc.). The recommended set of transactions will

indicate exactly with whom the transaction should be made, the exact GIVE and TAKE items and their quantities, as well as other relevant parameters (e.g., price and profit). For example, for procurement ATS, the MMO method recommends a set of suppliers ATS's and the exact quantities of the items to be purchased from each, so that the procurement ATS objective, say the minimal total cost, is achieved. Or, for a manufacturer's ATS, the MMO method can recommend a set of buyers ATS's interested in the manufacturer's products, and a set of ATS's suppliers of raw materials necessary to manufacture the products, so that the manufacturer's objective, say maximal profit, is achieved. The ATS-based match making and optimization are generic and work uniformly regardless of a specific wizard (or trader type) that generated them.

- **Wizards for Various Types of Traders.** The invention also comprises a library of predefined wizards (i.e., specialized "smart" interface templates) for various types of traders that participate in the Electronic Marketplace (e.g., suppliers, procurement organizations, manufacturers etc). For each type of trader, the wizard automatically constructs an ATS from the user given set of trading parameters relevant to a trading scenario. The trader who uses a wizard does not need to understand the mathematical description of an ATS, but rather trading parameters and concepts that are familiar to the trader (e.g. availability, quantity, price, revenue, etc.). The list of Wizards and examples of types of traders that may participate in the Electronic Market are described in the above-cited applications.

The disclosure allows creating new paradigms for the electronic marketplace as follows:

- **ATS-based Electronic Mall**. This is a novel and powerful paradigm, which encompasses "the regular" electronic mall and shop model. The main difference is that an ATS-based shop trades one or more ATS's, rather than just selling items. Thus, an ATS-based trader can sell items as a "regular" shop (i.e., have a supplier ATS), or procure (i.e., have a procurement ATS), or offer manufacturing capability (i.e., have a manufacturing ATS) etc. While a "regular" shop may be selling a long list of items, an ATS-based trader shop is likely to have only a small number of,

and often just one, ATS. For example, an office supplies shop may include its entire catalog of items in a single ATS, in which constraints will be on the limitation in available quantities, and will also include the price function (e.g., based on price per item and volume discounts). The fact that totally unrelated items may appear in the same ATS is not a problem, because traders visiting the mall (e.g., to search, match-make etc.) will, by default, see only item sets that are relevant to their interests (ATS's). Visiting traders could perform match-making that optimizes their own ATS objective and recommends the best (combined) transaction. This can be done against ATS's in a particular shop or the entire mall. A visiting trader can immediately perform a transaction with any ATS in the mall, provided it is mutually agreeable with the visitor's ATS.

- **ATS-based Elecronic Auctions.** This is a novel and powerful auction paradigm, that is especially suited for business-to-business trade, and which encompasses "regular" direct and reverse auctions as a simple case. In a regular auction, the auctioned bundle (or item) is fixed and indivisible; each bid is a monetary amount; and the outcome is a single winner who offered the highest bid. In contrast, in an ATS-based auction, the auctioned bundle is an ATS (and thus dividable among different bidders); each (partial) bid is an ATS; and the outcome may be one or many "winners", with whom the auctioning trader will perform a multiple combined transaction.

For example, suppose that a surplus seller initiates an auction with a surplus ATS, that has the objective of maximizing the overall revenue (whether the entire surplus or only part of it will be sold). Bids could be ATS-based offers from buyers (procurement organizations). Each such ATS bid may indicate, as TAKE items, only some, not all, of the surplus items, and have constraints on the quantities and prices offered (as in any ATS, it may involve flexible, rather than fixed, quantities and cost functions). When the auction ends, the system collects the latest ATS bids from all auction participants, and performs a match-making that optimizes the objective of the auctioned ATS (the maximal overall revenue, in our example). The outcome of the auction is a set of "winning" ATS's and the exact recommended quantities (quantities of

items in the surplus to be sold to each "winner" so that the maximal overall revenue will be received).

In the special case when an auctioned ATS indicates fixed quantities for traded items, and requests that there be only one winner, the ATS auction degenerates to a regular auction. However, regular auctions are not flexible enough for corporate traders. For example, running a regular auction for the entire surplus would eliminate potential auction participants and possibly higher monetary awards. One could also split the surplus into small bundles, or even individual items, and run a regular auction for each bundle. However, this approach would require to run possibly hundreds of tiny auctions – a tedious process just for one surplus package. Also, many corporate auction traders may not at all be willing to participate in tiny-scale auctions, which again would eliminate potential participants. In contrast, ATS auctions provides the best of both worlds: a single auction, no restrictions on potential participants, and the optimal outcome for the auction initiator (ATS).

The ATS auction is generic for any auctioned ATS and bid ATS's, regardless of how (or using which wizard) they were generated. Thus, one could also auction a procurement ATS, in which case suppliers and manufacturers may be potential bidders. Or, one could auction a manufacturing ATS, in which case procurement organizations as well as raw material suppliers may be potential bidders. Thus, in addition to its flexibility, ATS auctions capture "direct", "reverse", and any "in-between" auctioning, depending on the ATS being auctioned.

It is an advantage of the invention in comparison with prior art, that it facilitates *collapse of scale* in the Electronic Marketplace. Parts of small ATS's or entire small ATS's will be automatically aggregated to accommodate large counterpart ATS's, and large ATS's will be decomposed into sets of specific items to satisfy the limitations of smaller traders. This process will eliminate the need for costly intermediary traders whose only role is to consolidate or to divide trading baskets in order to match the difference in scale between end-traders. This

elimination or collapse of an intermediate tier will increase profit margins for producers and reduce prices for consumers, thus benefiting the market as a whole.

It is a further advantage of the invention in comparison with prior art, that it provides *dynamic supply and demand feedback* to the Electronic Marketplace. Traders will be able to continuously observe how their objectives are met by seeing only the items in the ATS's that are relevant to their requirements. Since traders can specify time constraints that are both in the present and in the future, without committing to perform a specific trade, they will be provided with forecasts that will dynamically monitor their opportunities.

It is a further advantage of the invention in comparison with prior art, that it provides *enhanced extensibility* to the Electronic Marketplace. An Electronic Marketplace that specializes in a particular type of commerce (e.g. auctions) may plan to expand into additional types of commerce (e.g. electronic malls, reverse-auctions etc.). The same basic primitives (Adaptive Trade Specifications, Match-Making, Optimization) will be reused in the expanded system, without requiring a full re-engineering of the existing system. The only change will occur at the Wizard interface level. The same advantage is evident when an existing marketplace is expanding from particular vertical markets (e.g. pharmaceuticals, electronics) to additional new vertical markets (e.g. automotive, financial).

Brief Description of the Drawings

A preferred embodiment of the present invention will be set forth in detail with reference to the drawings, in which:

FIG. 1 *ATS-Based Trading Software System,* describes a high level graphical summary of the suite of software tools related to the ATS-Based Trading Software System.

FIG. 2 *ATS-Based Match-Making and Optimization Hardware Architecture Diagram,* describes a high level graphical summary of the hardware architecture of the system.

FIG. 3 *Item Specification and Adaptive Trade Specification (ATS) Class Diagram,* presents a high level graphical summary of the Item Specification and Adaptive Trade Specification classes.

FIGS. 4A-4E *Functional Diagram of Match-Making and Optimization Method,* present a high level graphical summary of five Mathematical Programming Optimization Methods used by the system.

FIGS. 5A-5E *Flow Charts of Specific Match-Making and Optimization Methods,* present in greater detail the methods of Figs. 4A-4E.

FIG. 6 *ATS Based Electronic Mall State Diagram,* presents a high level graphical summary of the ATS Based Electronic Mall.

FIG. 7 *ATS Based Electronic Auction State Diagram,* presents a high level graphical summary of the ATS Based Electronic Auction.

Detailed Description of the Preferred Embodiment

A preferred embodiment of the present invention will now be set forth in detail with reference to the drawings, in which like reference numerals refer to like elements throughout.

Fig. 1 shows an overview of the operations carried out by the preferred embodiment. An ATS-based electronic marketplace 101 can include one or more of an ATS-based electronic mall 103, an ATS-based electronic auction (forward or reverse) 105, and any other ATS-based commerce environment. As noted above, participants in the marketplace 101 form ATS's through various techniques. One such technique is the use of wizards 107, including one or more of a procurement wizard 109, a supplier wizard 111, a manufacturing wizard 113, a surplus seller wizard 115, a reseller wizard 117, a generic buy and sell wizard 119, a generic buy wizard 121, a generic sell wizard 123, a trade-in wizard 125, and other wizards adapted to specific purposes. These wizards, like those wizards that are known in the programming art, are utilities that guide a user through a specific task.

The ATS's formed through use of the wizards 107 are input to the ATS match-maker 127, which uses matchmaking optimization methods to be described below.

The processes performed by the matchmaker 127 are object-oriented and follow the specifications of the ODMG (Object Database Management Group). A Constraint Object Oriented Database (CSPACE) 129 uses an iterative query language (IQL) 131 and a constraint and optimization library 133 to perform the matchmaking and optimization. The CSPACE 129 communicates through an ODMG wrapper 135 with an ODMG-compliant database manager 137 and also communicates directly with a mixed integer programming (MIP) solver 139.

The above is implemented on a hardware architecture that will now be explained with reference to Fig. 2. The hardware architecture capable of running an ATS based match-making and optimization system includes several logical tiers, each one performing specific computational tasks. Each tier can be described in terms of specific tasks that it performs. From the hardware perspective, each tier can be built from computers having sufficient computational power.

Tier 1 includes a database server 201, which is a power server machine (preferably dual or quad Pentium III machine) running one of the following network operating systems: Windows NT 4.0, Novell 5.0, UNIX. The database server 201 performs all tasks related to data persistency, data integrity and querying. The database server 201 runs one of the commercially available object oriented databases such as Poet, Objectivity, Object Store, etc.

Tier 2 includes the application server 203, which is a power server machine (preferably dual or quad Pentium III machine) running one of the following network operating systems: Windows NT 4.0, Novell 5.0, UNIX etc. The application server 203 performs all tasks related to performing ATS-based match-making and optimization. The data are passed between layers via RMI, CORBA, DCOM or any other distributed computing protocol allowing remote method invocation and data transmission.

Tier 3 includes a Web server 205, which is a computer that responds to requests from Web browsers via HTTP. The Web server 205 transfers text files and corresponding graphics and data

via HTTP to remote computers that are running Web browsers. The Web server 205 should have the functionality commonly associated with e-commerce Web servers, such as CGI (Common Gateway Interface), JSP, ASP, etc., for performing searches and other dynamic HTML functions and SSL (Secure Socket Layer) for handling secure transactions.

The servers 201, 203, and 205 communicate with another through an internal network. However, in order to be useful to users, the Web server 205 communicates via the Internet 207 or another publicly accessible network with Tier 4, which includes computers 209 running on users' premises and used as Web clients. The Web clients 209 are computers or other devices (such as WAP-enabled wireless devices) capable of running any standard off-the-shelf browser. The clients 209 run Web-based applications that will use information provided by the application server 203 and the Web server 205. The servers can be, but are not necessarily, implemented on separate machines. Other possibilities are separate virtual machines or simply separate software processes or threads.

FIG. 3 provides a high level graphical description of the classes Item Specification and Adaptive Trade Specification. An ATS class 301 includes four components: give-item-entries 303, take-item-entries 305, constraints 307 and an objective 309. The give-item-entries 303 identify what the particular user is willing to give in the trade and include one or more item specifications 311. The take-item-entries 305 identify what the user wants in return and include one or more item specifications 313. The constraints 307 set forth restrictions that must be satisfied before a transaction can be carried out, e.g., constraints on quantity or on time of delivery. The objective 309 indicates what the particular user wants to optimize; for example, a seller may want to optimize (maximize) profit, while a buyer may want to optimize (minimize) total cost.

Before the electronic mall and the electronic auction are described in detail, the various optimization processes that can be used in either will be described with reference to Figs. 4A-4E

and 5A-5E. FIGS. 4A-4E provide a high level graphical description of the methods outlined below. Figs. 5A-5E provide corresponding low-level descriptions.

A. Method for Constructing ATS MM Constraints (Figs. 4A and 5A)

Method (403) Name: *Construct-ATS-MM-Constraints ({A1,A2,...,An})*

Input (401): A set *{A1,A2,...,An}* of ATS's.

Output (405): *Constraints* that express the fact that ATS's in *{A1,A2,...,An}* are mutually agreeable.

Algorithm Description:

Step 501. Construct *Original-ATS-Constraints* as

Constraints of *A1* AND

Constraints of *A2* AND

... AND

Constraints of *An*.

Step 503. Construct *Give-Quantity-Constraints* as follows:

a. Initially, set *Give-Quantity-Constraints* to the empty conjunction (logical AND) of constraints,

i.e. a constraint that is equivalent to TRUE.

b. For each ATS *A* from the set *{A1,...,An}* and

For each item specification *IS* from *Give-Item-Entries* of *A* do:

Set *Give-Quantity-Constraints* to

Give-Quantity-Constraints AND *quantity-range* of *IS*

(note, the latter is Lower-bound[IS] <= Quantity[IS] <= Upper-bound[IS])

Step 505. Construct *Take-Quantity-Constraints* as follows:

a. Initially, set *Take-Quantity-Constraints* to the empty conjunction (logical AND) of constraints,

i.e. a constraint that is equivalent to FALSE.

b. For each ATS *A* from the set *{A1,...,An}* and

For each item specification *IS* from *Take-Item-Entries* of *A* do:

Set *Take-Quantity-Constraints* to

Take-Quantity-Constraints AND *quantity-range* of *IS*

(Note: the latter is Lower-bound[IS] <= Quantity[IS] <= Upper-bound[IS])

Step 507. Construct the set *All-Give-Item-Specs* as follows:

a. Set *All-Give-Item-Specs* to the empty set

b. For each ATS *A* from the input set *{A1,...,An}* of ATS's do:

Set *All-Give-Item-Specs* to *All-Give-Item-Specs* union *Item-Specs,*

where *Item-Specs* is the set of all item specifications in *Give-Item-Entries* of the ATS *A.*

Step 509. Construct the set *All-Take-Item-Specs* as follows:

a. Set *All-Take-Item-Specs* to the empty set.

b. For each ATS *A* from the input set *{A1,...,An}* of ATS's do:

Set *All-Take-Item-Specs* to *All-Take-Item-Specs* union *Item-Specs,*

where *Item-Specs* is the set of all item specifications in *Take-Item-Entries* of the ATS *A.*

Step 511. For each item specification *tIS* from *All-Take-Item-Specs* and

For each item specification *gIS* from *All-Give-Item-Specs* such that

Give-Take-Item-Match(gIS, tIS) = TRUE (i.e., *gIS* satisfies the requirements of *tIS*) do:

Create a new quantity variable *Quantity[gIS, tIS]* .

(Note: *Quantity[gIS, tIS]* expresses the quantity of *gIS* given toward the required quantity of *tIS*)

Step 513. Construct *Take-Zero-Sum-Constraints* as follows:

For each item specification *tIS* from *All-Take-Item-Specs* do:

a. Set *Zero-Sum-Constraints[tIS]* to

Quantity[tIS] = Quantity[gIS-1, tIS] + ... + Quantity[gIS-n, tIS]

where *gIS-1, ...,gIS-n* are all item specification from *All-Give-Item-Specs* that are satisfied by the item specification *tIS* (i.e., *Give-Take-Item-Match(gIS-I, tIS] = TRUE* for every *I = 1,...,n*)

b. Set *Take-Zero-Sum-Constraints* to

Zero-Sum-Constraints[tIS-1] AND ... AND *Zero-Sum-Constraints[tIS-m]*

where *tIS-1, ..., tIS-m* are all item specifications from *All-Take-Item-Specs.*

Step 515. Construct *Give-Zero-Sum-Constraints* as follows:

For each item specification *tIS* from *All-Give-Item-Specs* do:

a. Set *Zero-Sum-Constraints[gIS]* to

Quantity[gIS] = Quantity[gIS, tIS-1] + ... + Quantity[gIS, tIS-m]

where *tIS-1, ...,tIS-m* are all item specification from *All-Take-Item-Specs* that satisfy the item specification *gIS* (i.e., *Give-Take-Item-Match(gIS, tIS-I] = TRUE* for every *I = 1,...,m*)

b. Set *Give-Zero-Sum-Constraints* to

Zero-Sum-Constraints[gIS-1] AND ... AND *Zero-Sum-Constraints[gIS-n]*

where *gIS-1, ..., gIS-n* are all item specifications from *All-Give-Item-Specs.*

Step 517. Construct *Constraints* as

Original-Constraints AND

Give-Quantity-Constraints AND

Take-Quantity-Constraints AND

Give-Zero-Sum-Constraints AND

Take-Zero-Sum-Constraints

Step 519. Return *Constraints* as output.

End of Method.

Generic Multiple MM Optimization Method (Figs. 4B and 5B)

Method (407) Name: *ATS-Multiple-MM-Optimization({A1,...,An}, Objective, Additional-Constraints)*

Input (409):

1. A set *{A1,...,An}* of ATS's (411)
2. *Objective* of the class *Objective-Class* (recall: it includes an *Objective-Function* and an indication whether minimum or maximum is sought. (413)
3. *Additional-Constraints,* which can be used to describe additional interrelationships among numeric variables in different ATS's in *{A1,...,An}*. (415)

Output (417):

1. An *Optimal-Variable-Instantiation* into all variables that appear in *MM-Constraints({A1,...,An})* (including quantities of all item specifications) that achieves the optimal objective of the *Optimizing-ATS.* (419)
2. The *Optimal-Value* for the *Objective-Function* for the *Optimal-Variable-Instantiation.* (421)
3. A set *Winning-ATS-set* of winning filtered *ATS's* from *Committed-ATS-Set* in which all items specifications *IS* with *Quantity[IS] = 0* are eliminated. Also eliminated from *Winning-ATS-Set* are all ATS's in which both *Give-Item-Entries* and *Take-Item-Entries* became empty, after item specification with zero quantities were eliminated. (423)

Algorithm Description:

Step 521. Construct *MM-Constraints* by applying the method *Construct-ATS-MM-Constraints({A1,...,An})* on the input set of ATS's *{A1,...,An}*.

Step 523. Construct *Combined-Constraints* as

MM MM-Constraints AND *Additional-Constraints*

Steps 525-529. If Objective indicates that minimum is sought (step 525), apply the method Minimize(Objective-Function, Combined-Constraints) (step 527) that returns the optimal Value-At-Point (Recall: it has the attributes Optimal-Value of the type Real and Optimal-Point of the

class Variable-Instantiation-Class). Otherwise, if Objective indicates that maximum is sought, apply the method Maximize(Objective-Function, Combined-Constraints) (step 529) that returns the optimal Value-At-Point. (Recall: it has the attributes Optimal-Value of the type Real and Optimal-Variable-Instantiation of the class Variable-Instantiation-Class).

Step 531. Initialize *Winning-ATS-Set* as *{A1,...,An}*.

Step 533. For every ATS *A* in *Winning-ATS-Set* do:

a. For every item specification *IS* in *Give-Item-Entries* of *A* do:

If *Quantity[IS]* is instantiated to *0* by the variable instantiation *Value-At-Point* then

Delete *IS* from *Give-Item-Entries* and the related mapping to *Quantity-Ranges*

b. For every item specification *IS* in *Take-Item-Entries* of *A* do:

If *Quantity[IS]* is instantiated to *0* by the variable instantiation *Value-At-Point* then

Delete *IS* from *Give-Item-Entries* and the related mapping to *Quantity-Ranges*

c. If both *Give-Item-Entries* and *Take-Item-Entries* of *A* become empty after deletion of item specifications in steps *a.* and *b.*, then delete *A* from *Winning-ATS-Set.*

Step 535. Return as output:

a. Optimal-Variable-Instantiation which is the *Variable-Instantiation* which was returned in *Value-At-Point.*

b. The *Optimal-Value* which was returned in *Value-At-Point.*

c. Winning-ATS-Set

End of method.

One-to-All MM Optimization Method (Figs. 4C and 5C)

Method (425) Name: *ATS-One-to-All-MM-Optimization({Optimizing-ATS, Committed-ATS-Set})*

Input (427):

1. *Optimizing-ATS,* which is an ATS whose *Objective* will be used for optimization. (429)

2. *Committed-ATS-Set,* which is a set of ATS's that are committed to perform a transaction if and only if their *Constraints* are satisfied. The *Objectives* of the committed ATS's are not used in optimization. (431)

Output (433):

1. An *Optimal-Variable-Instantiation* into all variables that appear in *MM-Constraints({Optimizing-ATS}* union *Committed-ATS-Set)* (including quantities of all item specifications) that achieves the optimal objective of the *Optimizing-ATS.* (435)
2. The *Optimal-Value* for the *Objective-Function* for the *Optimal-Variable-Instantiation.* (437)
3. A set *Winning-ATS-set* of winning filtered *ATS's* from *Committed-ATS-Set* in which all items specifications *IS* with *Quantity[IS] = 0* are eliminated. Also eliminated from *Winning-ATS-Set* are all ATS's in which both *Give-Item-Entries* and *Take-Item-Entries* became empty after item specifications with zero associated quantity were eliminated. (439)

Algorithm Description:

Step 541. Set *ATS-Set* to the union of *Committed-ATS-Set* and the singleton set *{Optimizing-ATS}*

Step 543. Set *Objective* to the objective of *Optimizing-ATS*

Step 545. Set *Additional-Constraints* to the empty conjunction of constraints, i.e., the constraint equivalent to TRUE.

Step 547. Apply the method *ATS-Multiple-MM-Optimization(ATS-Set, Objective, Additional-Constraints)* to compute *Optimal-Variable-Instantiation, Optimal-Value* and *Winning-ATS-Set.*

Step 549. Return *Optimal-Variable-Instantiation, Optimal-Value* and *Winning-ATS-Set* as output.

End of Method

One-to-One MM Optimization Method (Figs. 4D and 5D)

Method (441) Name: *One-to-One-MM-Optimization({Optimizing-ATS, Committed-ATS-Set})*

Input (443):

1. *Optimizing-ATS*, which is an ATS whose *Objective* will be used for optimization. (445)
2. *Committed-ATS-Set*, which is a set of ATS's that are committed to perform a transaction if and only if their *Constraints* are satisfied. The *Objectives* of the committed ATS's are not used in optimization. (447)

Output (449):

1. *Winning-ATS*, from *Committed-ATS-Set*, which is recommended for making a deal with. All item specifications *IS* with *Quantity[IS] = 0* (in *Optimal-Variable-Instantiation* below) are deleted. (451)
2. An *Optimal-Variable-Instantiation* into all variables that appear in *MM-Constraints({Optimizing-ATS, Winning-ATS})* (including quantities of all item specifications) that achieves the optimal objective of the *Optimizing-ATS*. (453)
3. The *Optimal-Value* of the *Objective-Function* for the *Optimal-Variable-Instantiation*. (455)

Algorithm Description:

A. If the *Objective* of the *Optimizing-ATS* requires minimum, do:

D Step 551. Set *Current-Minimum* to + *infinity*

E Step 553. Set *Current-Variable-Instantiation* to null (i.e., undefined).

F Step 555. Set *Winning-ATS* to null (i.e., undefined).

G For each ATS *A* in *Committed-ATS-Set* do:

Step 557. Apply *ATS-Multiple-MM-Optimization* on the set *{Optimizing-ATS, A}* of ATS's, the *Objective* of *Optimizing-ATS*, and the empty *Additional-Constraints*.

Steps 559-565. If the returned *Optimal-Value* < *Current-Minimum*, as determined in step 559, do:

Step 561. Set *Current-Minimum* to *Optimal-Value;*

Step 563. Set *Current-Variable-Instantiation* to the returned *Optimal-Variable-Instantiation.*

Step 565. Set *Winning-ATS* to the current ATS *A.*

Step 567. Return as output:

Winning-ATS

Current-Variable-Instantiation as *Optimal-Variable-Instantiation*

Current-minimum as *Optimal-Value.*

B. If the *Objective* of the *Optimizing-ATS* requires maximum, do:

H Step 551. Set *Current-Maximum* to - *infinity*

I Step 553. Set *Current-Variable-Instantiation* to null (i.e., undefined).

J Step 555. Set *Winning-ATS* to null (i.e., undefined).

K For each ATS *A* in *Committed-ATS-Set* do:

Step 557. Apply *ATS-Multiple-MM-Optimization* on the set *{Optimizing-ATS, A}* of ATS's, the *Objective* of *Optimizing-ATS,* and the empty *Additional-Constraints.*

Steps 559-565. If the returned *Optimal-Value* > *Current-Minimum,* as determined in step 559, do:

Step 561. Set *Current-Maximum* to *Optimal-Value;*

Step 563. Set *Current-Variable-Instantiation* to the returned *Optimal-Variable-Instantiation.*

Step 565. Set *Winning-ATS* to the current ATS *A.*

Step 567 Return as output:

Winning-ATS

Current-Variable-Instantiation as *Optimal-Variable-Instantiation*

Current-Maximum as *Optimal-Value.*

End of Method

One-to-K MM Optimization Method (Figs. 4E and 5E)

Method (457) Name: *ATS-One-to-K-MM-Optimization({Optimizing-ATS, Committed-ATS-Set})*

Input (459):

1. *Optimizing-ATS,* which is an ATS whose *Objective* will be used for optimization. (461)
2. *Committed-ATS-Set,* which is a set of ATS's that are committed to perform a transaction if and only if their *Constraints* are satisfied. The *Objectives* of the committed ATS's are not used in optimization. (463)

Output (465):

1. An *Optimal-Variable-Instantiation* into all variables that appear in *MM-Constraints({Optimizing-ATS}* union *Winning-ATS-Set)* (including quantities of all item specifications) that achieves the optimal objective of the *Optimizing-ATS.* (467)
2. The *Optimal-Value* for the *Objective-Function* for the *Optimal-Variable-Instantiation.* (469)
3. *Winning-ATS-set* of at most K winning filtered *ATS's* from *Committed-ATS-Set* in which all items specifications *IS* with *Quantity[IS] = 0* are eliminated. Also eliminated from *Winning-ATS-Set* are all ATS's in which both *Give-Item-Entries* and *Take-Item-Entries* became empty after item specifications with zero associated quantity were eliminated. (471)

Algorithm Description:

Step 571. For each K ATS's *{A1,...,Ak}* in *Committed-ATS-Set,* perform *ATS-One-to-All-MM-optimization(Optimizing-ATS, {A1,...,Ak}).*

Step 573. Among all sets *{A1,...,Ak},* choose the one that has minimal (or maximal, as required in *Optimizing-ATS*) *Optimal-Value.*

Step 575. Return as output the output of *ATS-One-to-All-MM-Optimization* for the selected set *{A1,...,Ak}* with the minimal (or maximal, as required in *Optimizing-ATS)* objective.

End of Method.

General Description of ATS-based Electronic Mall

The ATS Based Electronic Mall allows several kinds of traders to participate in the activities within the electronic mall. The traders are individuals and organizations that can be categorized as procurers, suppliers, manufacturers, collaborating bidders, surplus sellers etc. However, these categories of traders have an underlying common representation. They all represent their trade items and trade objectives via ATS. Each ATS is built using a specific Wizard. The ATS's and Wizards have been described above and are described in still further detail in the above-cited related patent applications. The present application describes how transactions between ATS's are performed in an Electronic Mall.

An ATS in the Electronic Mall can be exactly in one of the following two states:

- **Committed State.** In this state the trader that owns the ATS is committed to perform any transaction initiated against his ATS, if these transactions are mutually agreeable with his ATS. The Objective Function of an ATS in Committed state is dormant.
- **Non-Committed State.** In this state the ATS is not included in a transaction initiated by another trader. A trader that owns a non-committed ATS can launch an optimization function for this ATS and request a recommendation for optimum trade according to his objective function. This function will be performed against other committed ATS's in the electronic mall.

A trader in the electronic mall can own both committed and non-committed ATS's.

A collection of *committed* ATS's is analogous to a **shop** in the Electronic Mall, and a *non-committed* ATS is analogous to a **shopper**. Unlike a regular mall, a *shopper* can buy from one or more *shops,* sell to one or more *shops*, or perform a combination of both buying and selling. In this analogy the *shopper* (the non-committed ATS) is examining the *shops* (the committed ATS's) trying to find an optimal trade that will suit his needs. At the *shopper's*

request, the system will recommend which items and quantities from which *shops* should the *shopper* buy or sell or both.

There are several types of activities that are performed by a trader outside the scope of a transaction:

- General activities performed in any electronic mall, such as registration, credit clearance, secure transaction capabilities, removal of trader from the mall etc.
- Specific activities for an ATS based electronic mall, outside a transaction, such as:
  - Add new ATS to the database.
  - Remove an existing ATS from the database.
  - Retrieve ATS from the server for editing.
  - Save retrieved ATS as a draft.
  - Save edited ATS to the database.

These specific activities and additional activities will be described below.

All the interactions with the electronic mall will be performed by traders using Web based thin clients. All the computational activities triggered by those interactions will be performed against scaleable Web and database servers.

ATS-based Electronic Mall Database Description

The system comprises of a database that contains at least the following information:

- ***All-ATS-In-Mall.*** The set of all committed ATS's in the mall against which traders can perform transactions.
- For each ATS in *All-ATS-In-Mall:*

  ATS-Owner. The trader who submitted this ATS to the mall.
- ***All-ATS-Traders.*** The set of all traders in the mall.
- For each trader in *All-ATS-Traders:*

- ***Trader-Info***. All the necessary information about a trader, including identification, description, size, legal and financial details etc.
- ***All-Draft-ATS***. The set of all ATS's saved as draft for this trader.
- ***All-Committed-ATS***. The set of all ATS's saved as committed for his trader.
- ***Current-ATS.*** The current ATS selected by the trader to be used as input for various actions, e.g. optimization, search etc.

ATS-based Electronic Mall Diagram Description

The diagram in FIG. 6 describes the process of trading in the ATS-based Electronic Mall. The diagram is composed of states, user inputs, and methods. The diagram is composed by the following states:

ATS Management State 602

This state allows the user/trader to achieve the following main purposes:

- Submit one or more ATS's to the Mall in committed state.
- Select the current ATS for optimization.

This state presents the end user with a graphical window that allows him to perform a series of user inputs. Each user input triggers an Action/Method. These user inputs are performed by the user by selecting a button in the window.

These are the user inputs that originate from this state and their respective triggered methods:

- Selecting the *Construct ATS* option will trigger the **ATS Construction Method 604**.
- Selecting the *View ATS* option will trigger the **ATS View Method 606**.
- Selecting the *Edit ATS* option will trigger the **ATS Edit Method 608**.
- Selecting the *Copy ATS* option will trigger the **ATS Copy Method 610**.
- Selecting the *Remove ATS* option will trigger the **ATS Remove Method 612**.

- Selecting the *Multiple Criteria Filtering ATS* option will trigger the **ATS Multiple Criteria Filtering Method 614.**
- Selecting the *Save ATS as Draft* option will trigger the **Save ATS as Draft Method 616.**
- Selecting the *Submit ATS to Mall* option will trigger the **Submit ATS to Mall Method 618.**

The Methods triggered by the user input are described below.

ATS Construction Method 604

Method Name: *ATS Construction.*

Input: *User input in the Construction Wizard.*

Output: *An ATS.*

Description:

This method allows the user to create a new ATS using an appropriate Wizard. The Wizard is a graphical template that facilitates creation of ATS's and is described in detail in one of the above-referenced patent applications. The Wizard used for this method is taken from an extensible library of wizards (i.e., specialized "smart" interfaces) developed for various types of traders (e.g., suppliers, procurement organizations, manufacturers, etc.). In addition to the library of generic wizards for each trader type, libraries of customized wizards will be developed to accommodate specifics of various vertical markets (e.g., pharmaceuticals, electronics, office supplies etc.) Also, specialty wizards will be developed as solutions to individual corporate customers. All wizards will be implemented as Java-based plug-ins to Internet browsers, and thus will require no software installation on the client side. The main function of each wizard is to specify an Adaptive Trade Specification (ATS), which is a formal mathematical description of trader's objective and constraints.

ATS View Method 606

Method Name: *View ATS.*

Input: *User selects an ATS for view from an ATS list displayed in the window.*

Output: *A window displaying the selected ATS.*

Description:

This method allows the user to select an ATS for view only. No modifications or any other operations are allowed on this ATS. If the trader is not the owner of the ATS that was selected for view, the system can deny his request if the original creator of the ATS restricted the viewing permissions on the ATS. ATS permissions and ownership are not covered by this patent. It sufficient to say that the creator of an ATS has *read* and *write* permissions on the ATS. Other traders in the system have *read* only permission or *no* permission on the ATS.

ATS Edit Method 608

Method Name: *ATS Edit.*

Input: *User selects an ATS for edit from an ATS list displayed in the window.*

Output: *A window displaying the selected ATS.*

Description:

This method allows the user to select an ATS for edit. The trader has to be the owner of the ATS. He can change all editable values in the ATS such as items, quantities, price, objective function etc.

ATS Copy Method 610

Method Name: *Copy ATS.*

Input: *User selects an ATS for copy from an ATS list displayed in the window.*

Output: *An ATS.*

Description:

This method allows the user to select an ATS for view and create his own copy of the ATS. This allows the trader to use an existing ATS as a template for the creation of a new ATS. The user becomes the owner of newly created ATS.

ATS Remove Method 612

Method Name: *Remove ATS.*

Input: *User selects an ATS for removal from an ATS list displayed in the window.*

Output: *Updated Database.*

Description:

This method allows the user to remove an ATS from the database. All references to this ATS will be updated.

ATS Multiple Criteria Filtering Method 614

Method Name: *Multi Criteria ATS Filtering.*

Input: *User inputs search criteria (trader name, quantity, item name etc.).*

Output: *A set of ATS's that meet these criteria.*

Description:

This method allows the user to perform a search against the ATS's database using a variety of search criteria (e.g. trader name, quantity, item name, item property). One or more search criteria can be simultaneously entered and a Boolean combination of these criteria can be specified. The filtering can be performed using the Match-Making Optimization method or traditional search paradigms. For example a trader can issue a search for all ATS's in the database that are mutually agreeable with all items in the current ATS, or mutually agreeable with specific items in the trader's current ATS. In addition, the trader can specify traditional search criteria such as retrieving all ATS's that belong to a certain trader, or that belong to a trader from a specific geographical location. At the conclusion of this method the trader is presented with a list of ATS's that answer his search criteria. The trader may select the ATS's that will participate in the Match-Making Optimization process.

Save ATS as Draft Method 616

Method Name: *Save ATS As Draft.*

Input: *User selects Save ATS as Draft option.*

Output: *An ATS in a state of draft or non-committed in the database.*

Description:

This method allows the user to save an ATS to the database in a state of draft. An ATS in this state is non-committed, is visible only to the trader that created it, is not part of the Mall, and no trade is performed against it. Such an ATS is considered "work in progress" and can be retrieved later by its owner for editing or removal.

Submit ATS to Mall Method 618

Method Name: *Submit ATS to Mall.*

Input: *User selects Submit ATS to Mall option.*

Output: *An ATS in a committed state in the database.*

Description:

This method allows the user to submit an ATS to the mall. The ATS is saved in the database and is committed to participate in any transactions that satisfy its constraints.

ATS Filtering State 620

This state allows the user/trader to achieve the following main purpose:

- Search the database for ATS's according to certain criteria.
- Select the ATS's which are submitted as input to the Match-Making (MM) Optimization process.

This state presents the end user with a graphical window that allows him to perform a series of user inputs. Each user input triggers an Action/Method. These user inputs are performed by the user by selecting a button in the window.

These are the user inputs that originate from this state, and their respective triggered methods:

- Selecting the *Multi-Criteria-ATS-Filtering* option will trigger the **Multi Criteria ATS Filtering Method 622.**

The Methods triggered by the user input are described below.

Multi Criteria ATS Filtering Method

Method Name: *Multi Criteria ATS Filtering.*

Input: *User inputs search criteria (trader name, quantity, item name etc.).*

Output: *A set of ATS's that meet these criteria.*

Description:

This method allows the user to perform a search against the ATS's database using a variety of search criteria (e.g. trader name, quantity, item name, item property). One or more search criteria can be simultaneously entered and a Boolean combination of these criteria can be specified. The filtering can be performed using the Match-Making Optimization method or traditional search paradigms. For example a trader can issue a search for all ATS's in the database that are mutually agreeable with all items in the current ATS, or mutually agreeable with specific items in the trader's current ATS. In addition, the trader can specify traditional search criteria such as retrieving all ATS's that belong to a certain trader, or that belong to a trader from a specific geographical location. At the conclusion of this method the trader is presented with a list of ATS that answer his search criteria. The trader may select the ATS's that will participate in the Match-Making Optimization process.

ATS Match-Making Optimization State 624

This state allows to user/trader to achieve the following main purpose:

Receive a recommended transaction from the Match-Making (MM) Optimization process.

Given a trader's ATS, the Match-Making Optimization process recommends specific transactions with other traders (i.e., against their ATS's) that are mutually agreeable and optimize the objective of the trader's ATS (e.g., minimal price, maximal profit etc.). The recommended set

of transactions will indicate exactly with whom the transaction should be made, the exact GIVE and TAKE items and their quantities, as well as other relevant parameters (e.g., price and profit). For example, for procurement ATS, the MM optimization methods can recommend a set of suppliers ATS's and the exact quantities of the items to be purchased from each, so that the procurement ATS objective, say the minimal total cost, is achieved. Or, for a manufacturer's ATS, the MM optimization methods can recommend a set of buyers ATS's interested in the manufacturer's products, and a set of ATS's suppliers of raw materials necessary to manufacture the products, so that the manufacturer's objective, say maximal profit, is achieved. The ATS-based match making and optimization are generic and work uniformly regardless of how or for what type of trader the input ATS's were generated (e.g., what "wizard" interface generated them).

This state presents the end user with a graphical window that allows him to perform a series of user inputs. Each user input triggers an Action/Method. These user inputs are performed by the user by selecting a button in the window.

These are the user inputs that originate from this state and their respective triggered methods:

Selecting the *One-To-All-MM-Optimization* option will trigger the **One-To-All-MM-Optimization Method 626**.

Selecting the *One-To-One-MM-Optimization* option will trigger the **One-To-One-MM-Optimization Method 628.**

Selecting the *One-To-K-MM-Optimization* option will trigger the **One-To-K-MM-Optimization Method 630.**

The Methods triggered by the user input are described below.

One-To-One-MM-Optimization Method 626

Method Name: *One-to-One-MM-Optimization({Optimizing-ATS, Committed-ATS-Set})*

Input:

- *Optimizing-ATS*, which is an ATS whose *Objective* will be used for optimization.
- *Committed-ATS-Set*, which is a set of ATS's that are committed to perform a transaction if and only if their *Constraints* are satisfied. The *Objectives* of the committed ATS's are not used in optimization.

Output:

- *Winning-ATS*, from *Committed-ATS-Set*, which is recommended for making a deal with. All item specifications *IS* with *Quantity[IS] = 0* (in *Optimal-Variable-Instantiation* below) are deleted.
- An *Optimal-Variable-Instantiation* into all variables that appear in *MM-Constraints({Optimizing-ATS, Winning-ATS})* (including quantities of all item specifications) that achieves the optimal objective of the *Optimizing-ATS*.

  The *Optimal-Value* of the *Objective-Function* for the *Optimal-Variable-Instantiation*.

One-To-All-MM-Optimization Method 628

Method Name: *One-to-All-MM-Optimization({Optimizing-ATS, Committed-ATS-Set})*

Input:

- *Optimizing-ATS*, which is an ATS whose *Objective* will be used for optimization.
- *Committed-ATS-Set*, which is a set of ATS's that are committed to perform a transaction if and only if their *Constraints* are satisfied. The *Objectives* of the committed ATS's are not used in optimization.

Output:

- An *Optimal-Variable-Instantiation* into all variables that appear in *MM-Constraints({Optimizing-ATS}* union *Committed-ATS-Set)* (including quantities of all item specifications) that achieves the optimal objective of the *Optimizing-ATS*.
- The *Optimal-Value* for the *Objective-Function* for the *Optimal-Variable-Instantiation*.

- A set *Winning-ATS-set* of winning filtered *ATS's* from *Committed-ATS-Set* in which all items specifications *IS* with *Quantity[IS] = 0* are eliminated. Also eliminated from *Winning-ATS-Set* are all ATS's in which both *Give-Item-Entries* and *Take-Item-Entries* became empty after item specifications with zero associated quantity were eliminated.

One-To-K-MM-Optimization Method 630

Method Name: *One-to-K-MM-Optimization({Optimizing-ATS, Committed-ATS-Set})*

Input:

- *Optimizing-ATS,* which is an ATS whose *Objective* will be used for optimization.
- *Committed-ATS-Set,* which is a set of ATS's that are committed to perform a transaction if and only if their *Constraints* are satisfied. The *Objectives* of the committed ATS's are not used in optimization.

Output:

- An *Optimal-Variable-Instantiation* into all variables that appear in *MM-Constraints({Optimizing-ATS}* union *Winning-ATS-Set)* (including quantities of all item specifications) that achieves the optimal objective of the *Optimizing-ATS.*
- The *Optimal-Value* for the *Objective-Function* for the *Optimal-Variable-Instantiation.*
- *Winning-ATS-set* of at most K winning filtered *ATS's* from *Committed-ATS-Set* in which all items specifications *IS* with *Quantity[IS] = 0* are eliminated. Also eliminated from *Winning-ATS-Set* are all ATS's in which both *Give-Item-Entries* and *Take-Item-Entries* became empty after item specifications with zero associated quantity were eliminated.

Transaction Management State 632

This state allows the user/trader to achieve the following main purpose:

- Execute the transaction that was recommended by the Match-Making (MM) Optimization process in the previous state.

This state presents the end user with a graphical window that allows him to perform a series of user inputs. Each user input triggers an Action/Method. These user inputs are performed by the user by selecting a button in the window.

These are the user inputs that originate from this state and their respective triggered methods:

- Selecting the *Perform Transaction* option will trigger the **Perform Transaction Method 634.**

Perform Transaction Method 634

Method Name: *Perform Transaction({Optimizing-ATS, Winning-ATS-Set})*

Input:

1. *Optimizing-ATS* of the transaction
2. *Winning-ATS-Set* of the transaction.

Output:

If the transaction succeeds then:

1. *Update affected ATS's in the database.*
2. *Notify the traders that own the affected ATS.*
3. *Perform all necessary financial, contractual, legal activities related to the transaction, the ATS's involved in the transaction, and their owners.*

Else

Notify the owner of the optimizing ATS that the transaction has failed.

End If.

Description:

If a transaction succeeds, the system will notify the owners of the affected ATS's that items from their ATS's are involved in a specific committing transaction. The system will update the affected items of all participating ATS's in the database.

If the transaction fails because of changes in the participating ATS's, or other software or hardware failures, the owner of the optimizing ATS will be notified accordingly.

The system will issue contractual, financial, and legal procedures to complete the transaction. The details of these procedures are beyond the scope of this patent.

General Description of ATS-based Electronic Auction

This is a novel and powerful auction paradigm, that is especially suited for business-to-business trade, and which encompasses *regular direct* and *reverse* auctions as a simple case. In a regular auction, the auctioned bundle (or item) is fixed and indivisible; each bid is a monetary amount; and the outcome is a single winner who offered the highest bid. In contrast, in an ATS-based auction, the auctioned bundle is an ATS (and thus dividable among different bidders); each bid is an ATS; and the outcome may be one or many *winners*, with whom the auctioning trader will perform a mutually beneficial multiple combined transaction.

Unlike a regular auction, the ATS Based Electronic Auction allows a certain participant (represented by one or many ATS's) to participate simultaneously in multiple concurrent auctions. The match-making engine will notify the participant when her ATS was affected by one or more concurrent auctions. Since the system treats all auctions as one match-making and optimization pool of auctions, it can affect ATS's that take part in different auctions. Each auction will notify the owner of an affected ATS at the auction's expiration time. If two or more auctions have an identical expiration time, the system will differentiate them based on a universal unique identifier assigned to the auction at its instantiation.

Participants

The ATS Based Electronic Auction will allow several kinds of traders to participate in the activities within the electronic auction. The traders will be individuals and organizations that can be categorized as procurers, suppliers, manufacturers, collaborating bidders, surplus sellers, etc. However, these categories of traders have an underlying common representation. They all

represent their trade items and trade objectives via ATS. Each ATS is built using a specific Wizard. In an ATS Based Electronic Auction all the ATS's involved are in a *committed* state.

Genericity

The ATS auction is generic for any auctioned ATS and bid ATS's, regardless of how (or using which wizard) they were generated. Thus, one could also auction a procurement ATS, in which case suppliers and manufacturers may be potential bidders. Or, one could auction a manufacturing ATS, in which case procurement organizations as well as raw material suppliers may be potential bidders. Thus, in addition to its flexibility, ATS auctions capture *direct*, *reverse*, and any *in-between* auctioning, depending on the ATS being auctioned.

Visibility

The participants in the auction can choose between two states of ATS visibility. Their ATS's can be either visible to other traders or hidden from other traders. An auction where the participant ATS's are hidden may benefit the auctioneer by boosting the highest bid he can obtain.

Single Round Auction

A *single round auction* is an ATS based electronic auction where only one round of bidding is allowed for each bidder. The auction is concluded when the time set a priori by the auction initiator has expired.

For example, suppose that a surplus seller initiates an auction with a surplus ATS that has the objective of maximizing the overall revenue (whether the entire surplus or only part of it will be sold). Bids could be ATS-based offers from buyers (procurement organizations). Each such ATS bid may indicate, as TAKE items, only some, not all, of the surplus items, and have constraints on the quantities and prices offered (as in any ATS, it may involve flexible, rather than fixed, quantities and cost functions). When the auction ends, the systems collects the latest ATS bids from all auction participants, and performs a match-making that optimizes the objective of the auctioned ATS (the

maximal overall revenue, in our example). The outcome of the auction is a set of "winning" ATS's and the exact recommended quantities (quantities of items in the surplus to be sold to each "winner" so that the maximal overall revenue will be received).

In the special case when an auctioned ATS indicates fixed quantities for traded items, and requests that there be only one winner, the ATS auction degenerates to a regular auction. However, regular auctions are not flexible enough for corporate traders. For example, running a regular auction for the entire surplus would eliminate potential auction participants and possibly higher monetary awards. One could also split the surplus into small bundles, or even individual items, and run a regular auction for each bundle. However, this approach would require to run possibly hundreds of tiny auctions – a tedious process just for one surplus package. Also, many corporate auction traders may not at all be willing to participate in tiny-scale auctions, which again would eliminate potential participants. In contrast, ATS auctions provide the best of both worlds: a single auction, no restrictions on potential participants, and the optimal outcome for the auction initiator (ATS).

Multiple Round Auction

A *multiple round auction* is an ATS based electronic auction where more than one round of bidding is allowed for each bidder. The auction is concluded when the time set a priori by the auctioneer has expired.

An ***Auction State*** is defined as set of affected items in each bidding ATS that is among the projected winning bids, at a specific time within the life span of the auction. The Auction State describes the instantiated ATS's as a result of applying the Match-Making Optimization mechanism. The system updates the *Auction State* when a new ATS is entered as an auction bid, when a bidder changes an ATS, and when a bidder removes and ATS from the auction. At any time during the auction the bidder can view the Auction State, evaluate which items in her ATS are being engaged into a potential trade, and retrieve his ATS in order to perform a change. If the bidder retrieves an ATS for change, a copy of the original ATS remains active in the auction,

until he commits the new changes. After the new changes are committed, the system will trigger the Match-Making Optimization mechanism in order to update the *Auction State.*

If an ATS participates concurrently in more than one Multiple Round Auction, the system will update the *Auction State* of the first time-wise relevant auction in which the bidder participates.

The termination of the auction, specified in advance as the expiration time, will trigger a final Match-Making and Optimization cycle, and will establish a final *Auction State.* This state will reflect the distribution of the auctioned items among the participants.

For example, suppose that a surplus seller initiates an auction with a surplus ATS that has the objective of maximizing the overall revenue (whether the entire surplus or only part of it will be sold). Bids could be ATS-based offers from buyers (procurement organizations). Each such ATS bid may indicate, as TAKE items, only some, not all, of the surplus items, and have constraints on the quantities and prices offered (as in any ATS, it may involve flexible, rather than fixed, quantities and cost functions). Let's suppose that a new trader submits his ATS as a bid. The system will perform an Match-Making and Optimization cycle taking into account the new ATS and will present the new *Auction State* to all the new trader. Based on the information contained in the *Auction State* the trader can change specific constraints on items within her ATS (e.g. quantity, unit price, availability etc.) in order to improve his chances of "winning" the auction. This process can be repeated by the same trader or by other traders until the auction expires.

When the auction ends, the systems collects the latest ATS bids from all auction participants, and performs a match-making that optimizes the objective of the auctioned ATS (the maximal overall revenue, in our example). The outcome of the auction is a set of "winning" ATS's and the exact recommended quantities (quantities of items in the surplus to be sold to each "winner" so that the maximal overall revenue will be received).

If a bidder participates in several concurrent auctions the items of his winning ATS are updated at the end of the first auction, and the remaining quantities in his ATS are moved to the next

auction that is about to finish. This process continues until the last auction in which an ATS participates expires, or the quantities of the ATS are exhausted.

ATS-based Electronic Auction Database Description

The system comprises of a database that contains at least the following information:

- ***All-Active-Auctions***. The set of all active auctions in the system.
- For each auction in ***All-Active-Auctions***:
  - ***Auction-Initiator***. The trader that initiated the auction on an ATS.
  - ***Auction-Initiator-ATS.*** The ATS auctioned by the auction initiator.
  - ***Auction-Type.*** The type of the auction: single-round or multiple-round.
  - ***Auction-Optimization-Type.*** The type of optimization that the auction initiator will request:
    - One-To-One
    - One-To-All
    - One-To-K
  - ***All-Current-Committed-Bids***. The set of all ATS's that are bids in the auction.
  - ***Auction-Start-Time.*** The time at which the auction starts.
  - ***Auction-End-Time***. The time at which the auction ends.
- ***All-Active-Bids.*** The set of all bids that are performed in the system.
- For each active bid in ***All-Active-Bids:***
  - ***Bidding-ATS***. The ATS that corresponds to that bid.
  - ***Bid-Auctions-Set***. An ordered set of auctions against which the bid will be submitted (*Note*: This bid is committed to the first auction in the ordered set.).
  - ***First-Active-Auction.*** The first auction to which the bid is submitted.

- ***Auction-State.*** The state of the *First-Active-Auction.* This includes the list of potential winners if the auction ended now.

- ***All-ATS-Traders***. The set of all traders in the auction.
- For each trader in ***All-ATS-Traders:***
    - ***All-ATS-Bids***. The set of ATS's that belong to the trader and participate in active auctions.
    - ***All-Auctioned-ATS***. The set of ATS's auctioned, against the trader has active bids.

ATS-based Electronic Auction Diagram Description

The diagram in FIG. 7 describes the process of trading in the ATS-based Electronic Auction. The diagram is composed of states, user inputs, and methods. The diagram is composed of the following states:

ATS Auction Management State 702

This state allows the user/trader to achieve the following main purposes

- Submit one or more committed ATS's to one or more Auctions.

This state presents the end user with a graphical window that allows him to perform a series of user inputs. Each user input triggers an Action/Method. These user inputs are performed by the user by selecting a button in the window.

These are the user inputs that originate from this state and their respective triggered methods:

- Selecting the *Construct ATS* option will trigger the **ATS Construction Method 704.**
- Selecting the *View ATS* option will trigger the **ATS View Method 706.**
- Selecting the *Edit ATS* option will trigger the **ATS Edit Method 708.**
- Selecting the *Copy ATS* option will trigger the **ATS Copy Method 710.**
- Selecting the *Remove ATS* option will trigger the **ATS Remove Method 712.**

- Selecting the *Multiple Criteria Filtering ATS* option will trigger the **ATS Multiple Criteria Filtering Method 714.**
- Selecting the *Initiate Auction* option will trigger the **Initiate Auction Method 716.**
- Selecting the *Put a Bid* option will trigger the **Put a Bid Method 718.**
- Selecting the *View Own Bids* option will trigger the **View Own Bids Method 720.**
- Selecting the *View Auction State* option will trigger the **View Auction State Method 722.**

The Methods triggered by the user input are described below.

ATS Construction Method 704

Method Name: *ATS Construction.*

Input: *User input in the Construction Wizard.*

Output: *An ATS.*

Description:

This method allows he user to create a new ATS using an appropriate Wizard. The Wizard is a graphical template that facilitates creation of ATS's and is described in detail in a co-pending patent application referenced above ("Automated Methods for Creation of Adaptive Trade Specifications"). The Wizard used for this method is taken from an extensible library of wizards (i.e., specialized "smart" interfaces) developed for various types of traders (e.g., suppliers, procurement organizations, manufacturers etc. In addition to the library of generic wizards for each trader type, libraries of customized wizards will be developed to accommodate specifics of various vertical markets (e.g., pharmaceuticals, electronics, office supplies etc.) Also, specialty wizards will be developed as solutions to individual corporate customers. All wizards will be implemented as Java-based plug-ins to Internet browsers, and thus will require no software installation on the client side. The main function of each wizard is to specify an Adaptive Trade Specification (ATS), which is a formal mathematical description of trader's objective and constraints, such as in the examples above.

ATS View Method 706

Method Name: *View ATS.*

Input: *User selects an ATS for view from an ATS list displayed in the window.*

Output: *A window displaying the selected ATS.*

Description:

This method allows the user to select an ATS for view only. No modifications or any other operations are allowed on this ATS. If the trader is not the owner of the ATS that was selected for view, the system can deny his request if the original creator of the ATS restricted the viewing permissions on the ATS. ATS permissions and ownerships are not covered by this patent. It is sufficient to say that the creator of an ATS has read and write permissions on the ATS. Other traders in the system have read only permission or no permission on the ATS.

ATS Edit Method 708

Method Name: *ATS Edit.*

Input: *User selects an ATS for edit from an ATS list displayed in the window.*

Output: *A window displaying the selected ATS.*

Description:

This method allows the user to select an ATS for edit. The trader has to be the owner of the ATS. He can change all editable values in the ATS such as items, quantities, price, objective function etc.

ATS Copy Method 710

Method Name: *Copy ATS..*

Input: *User selects an ATS for copy from an ATS list displayed in the window.*

Output: *An ATS.*

Description:

This method allows the user to select an ATS for view and create his own copy of the ATS. This allows the trader to use an existing ATS as a template for the creation of a new ATS. The user becomes the owner of newly created ATS.

ATS Remove Method 712

Method Name: *Remove ATS.*

Input: *User selects an ATS for removal from an ATS list displayed in the window.*

Output:

Description:

This method allows the user to remove an ATS from the database. All references to this ATS will be updated.

ATS Multiple Criteria Filtering Method 714

Method Name: *Multi Criteria ATS Filtering.*

Input: *User inputs search criteria (trader name, quantity, item name, auction etc.).*

Output: *A set of ATS's that meet these criteria.*

Description:

This method allows the user to perform a search against the ATS's database using a variety of search criteria (e.g. trader name, quantity, item name, item property). One or more search criteria can be simultaneously entered and a Boolean combination of these criteria can be specified. The filtering can be performed using the Match-Making Optimization method or traditional search paradigms. For example a trader can issue a search for all ATS's in the database that are mutually agreeable with all items in the current ATS, or mutually agreeable with specific items in the trader's current ATS. In addition, the trader can specify traditional search criteria such as retrieving all ATS's that belong to a certain trader, or that belong to a trader from a specific geographical location. At the conclusion of this method the trader is presented with a list of ATS that answer his search criteria.

The trader can specify if he is searching against ATS's that are auctioned, bidding ATS's in an auction, or ATS's in the entire system.

Initiate Auction Method 716

Method Name: *Initiate Auction.*

Input: *One ATS that will be auctioned.*

Output: *A new Auction with a start time and end time.*

Description:

This method allows the user to create a new auction in the database. The trader that performs this auction is the auction initiator. The ATS selected by the auction initiator is the ATS that is auctioned.

Put a Bid Method 718

Method Name: *Put a Bid.*

Input: *One ATS that will be submitted as a bid in one auction.*

Output: *A new bid in the auction.*

Description:

This method allows the user to submit an ATS as a bid to an auction. The ATS can participate in multiple auctions. The ATS is committed to the *first auction* to which it was submitted. The *first auction* is described as follows:

Given a set of auctions {A1, A2, A3,...An} each starting at a point in time {t(A1), t(A2), t(A3)...t(An)} where $t(A1) < t(A2) < t(A3) < \ldots < t(An)$, the *first auction* is A1. When the *first auction* expires, A2 becomes the *first auction*, and so on. If an ATS bid is among the winners in the first auction, the items affected are updated by the system, and the remaining items in the ATS are moved to the next auction. This process is repeated until the last auction in which the ATS participates expires, or all items in the ATS are exhausted.

View Own Bids Method 720

Method Name: View Own Bids.

Input: *The trader's unique id.*

Output: *The set of ATS's that belong to the trader that participate as bids in all active auctions in the system.*

Description:

This method allows the user to view all the ATS's that are bids in active auctions. In a multiple round auctions the trader is allowed to edit the ATS's and to resubmit them as bids in order to improve his chances to be among the projected winners.

View Auction State Method 722

Method Name: View Auction State.

Input: *The Auction's unique id.*

Output: *The current state of the auction*

Description:

An *Auction State* is defined as set of affected items in each bidding ATS that is among the projected winning bids, at a specific time within the life span of the auction. A bidder in the auction can ask the system to display the Auction State at any time in the lifetime of the auction.

ATS Auction Filtering State 724

This state allows the user/trader to achieve the following main purpose:

- Search the database for *auctioned* ATS's according to certain criteria.

This state presents the end user with a graphical window that allows him to perform a series of user inputs. Each user input triggers an Action/Method. These user inputs are performed by the user by selecting a button in the window.

There is only one user input originating from this state, and thus only one triggered method:

- Selecting the *Multi Criteria Auction Filtering* option will trigger the **Multi Criteria Auction Filtering Method 726.**

The Method triggered by the user input is described below.

Multi Criteria Auction Filtering Method 726

Method Name: *Multi Criteria ATS Filtering.*

Input: *User inputs search criteria (trader name, quantity, item name etc.).*

Output: *A set of active auctioned ATS's that meet these criteria.*

Description:

This method allows the user to perform a search against the active auctioned ATS's in the database using a variety of search criteria (e.g. trader name, quantity, item name, item property). One or more search criteria can be simultaneously entered and a Boolean combination of these criteria can be specified. The filtering can be performed using the Match-Making Optimization method or traditional search paradigms. For example a trader can issue a search for all active auctioned ATS's in the database that are mutually agreeable with all items in the trader's ATS, or mutually agreeable with specific items in the trader's ATS. In addition, the trader can specify traditional search criteria such as retrieving all ATS's that belong to a certain trader, or that belong to a trader from a specific geographical location. At the conclusion of this method the trader is presented with a list of auctioned ATS's that answer his search criteria.

ATS Auction Management State 728

This state allows to user/trader to achieve the following main purpose:

- Perform an optimal transaction with the set of winning bidding ATS's at the conclusion of the auction.

This state presents the end user with a graphical window that allows him to perform a series of user inputs. Each user input triggers an Action/Method. These user inputs are performed by the user by selecting a button in the window.

Like the ATS Auction Filtering State 724, the ATS Auction Management State 728 has only one user input and thus only one triggered method. On the other hand, that one triggered method provides multiple options that will be explained below.

- Selecting the *Manage Auction Closure* option will trigger the **Manage Auction Closure Method 730**.

The Method triggered by the user input is described below.

Manage Auction Closure Method 730

Method Name: *Manage Auction Closure.*

Input: *Current auction.*

Output: *Updated database, performed transaction.*

Description:

Given an auctioned ATS, the Match-Making Optimization method recommends specific transactions with the winning bidders. The winning bidders are the ATS's that are mutually agreeable and optimize the objective of the auctioned ATS (e.g., minimal price, maximal profit etc.). The initiating auction trader can decide which type of auction optimization he wants to perform:

- ***One-To-One optimization 732***. The system will present one winning ATS bidder at the end of the auction.
- ***One-To-All optimization 734.*** The system will present all winning ATS's at the end of the auction.
- ***One-To-K optimization 736***. The system will present at most K winning ATS's at the end of the auction.

If the trader who owns the initiating auction selects:

One-To-One option 732, then perform *One-To-One-MM-Optimization Method*;

One-To-K option 734, then perform *One-To-One-MM-Optimization Method*;

One-To-All option 736, then perform *One-To-One-MM-Optimization Method*;

One-To-One-MM-Optimization Method 732

Method Name: *One-to-One-MM-Optimization({Optimizing-ATS, Committed-ATS-Set})*

Input:

- *Optimizing-ATS*, which is an ATS whose *Objective* will be used for optimization.
- *Committed-ATS-Set*, which is a set of ATS's that are committed to perform a transaction if and only if their *Constraints* are satisfied. The *Objectives* of the committed ATS's are not used in optimization.

Output:

- *Winning-ATS*, from *Committed-ATS-Set*, which is recommended for making a deal with. All item specifications *IS* with *Quantity[IS] = 0* (in *Optimal-Variable-Instantiation* below) are deleted.
- An *Optimal-Variable-Instantiation* into all variables that appear in *MM-Constraints({Optimizing-ATS, Winning-ATS})* (including quantities of all item specifications) that achieves the optimal objective of the *Optimizing-ATS*.
- The *Optimal-Value* of the *Objective-Function* for the *Optimal-Variable-Instantiation*.

One-To-All-MM-Optimization Method 734

Method Name: *One-to-All-MM-Optimization({Optimizing-ATS, Committed-ATS-Set})*

Input:

- *Optimizing-ATS*, which is an ATS whose *Objective* will be used for optimization.
- *Committed-ATS-Set*, which is a set of ATS's that are committed to perform a transaction if and only if their *Constraints* are satisfied. The *Objectives* of the committed ATS's are not used in optimization.

Output:

- An *Optimal-Variable-Instantiation* into all variables that appear in *MM-Constraints({Optimizing-ATS}* union *Committed-ATS-Set)* (including quantities of all item specifications) that achieves the optimal objective of the *Optimizing-ATS.*
- The *Optimal-Value* for the *Objective-Function* for the *Optimal-Variable-Instantiation.*
- A set *Winning-ATS-set* of winning filtered *ATS's* from *Committed-ATS-Set* in which all items specifications *IS* with *Quantity[IS] = 0* are eliminated. Also eliminated from *Winning-ATS-Set* are all ATS's in which both *Give-Item-Entries* and *Take-Item-Entries* became empty after item specifications with zero associated quantity were eliminated.

One-To-K-MM-Optimization Method 736

Method Name: *One-to-K-MM-Optimization({Optimizing-ATS, Committed-ATS-Set})*

Input:

- *Optimizing-ATS,* which is an ATS whose *Objective* will be used for optimization.
- *Committed-ATS-Set,* which is a set of ATS's that are committed to perform a transaction if and only if their *Constraints* are satisfied. The *Objectives* of the committed ATS's are not used in optimization.

Output:

- An *Optimal-Variable-Instantiation* into all variables that appear in *MM-Constraints({Optimizing-ATS}* union *Winning-ATS-Set)* (including quantities of all item specifications) that achieves the optimal objective of the *Optimizing-ATS.*
- The *Optimal-Value* for the *Objective-Function* for the *Optimal-Variable-Instantiation.*
- *Winning-ATS-set* of at most K winning filtered *ATS's* from *Committed-ATS-Set* in which all items specifications *IS* with *Quantity[IS] = 0* are eliminated. Also eliminated from *Winning-ATS-Set* are all ATS's in which both *Give-Item-Entries* and *Take-Item-Entries* became empty after item specifications with zero associated quantity were eliminated.

If a transaction succeeds, the system will notify the owners of the affected ATS's that items from their ATS's are involved in a specific committing transaction. The system will update the affected items all of participating ATS's in the database.

If the transaction fails because of changes in the participating ATS's, or other software or hardware failures, the owner of the optimizing ATS will be notified accordingly.

The system will issue contractual, financial, and legal procedures to complete the transaction. The details of these procedures are beyond the scope of this patent, but will be apparent to those skilled in the art of electronic commerce in general and electronic business-to-business commerce in particular.

While a preferred embodiment of the present invention has been set forth in detail above, those skilled in the art who have reviewed the present disclosure will readily appreciate that other embodiments can be realized within the scope of the present invention. For example, disclosures of certain hardware, operating systems, and other software are illustrative rather than limiting, as are specific numerical values. Therefore, the present invention should be construed as limited only by the appended claims.

We claim:

1. A system for implementing an electronic mall for a plurality of traders, the system comprising:

(a) a database server; and

(b) a database stored on the database server, the database comprising a plurality of adaptive trade specifications, each of the plurality of adaptive trade specifications comprising, for one of the traders:

(i) at least one give-item entry identifying at least one item that said one of the traders is willing to give in an exchange;

(ii) at least one take-item entry identifying at least one item that said one of the traders wants in return for the at least one item identified in the give-item entry; and

(iii) an indication of whether the adaptive trade specification is in (A) a committed state in which the trader corresponding to the adaptive trade specification is committed to perform any transaction matching the adaptive trade specification or (B) a non-committed state in which the trader is free to initiate transactions.

2. The system of claim 1, wherein each of the plurality of adaptive trade specifications further comprises:

(iv) at least one constraint entry identifying at least one constraint that said one of the traders has placed on the exchange; and

(v) an objective entry identifying an objective sought by said one of the traders in the exchange.

6. The system of claim 1, wherein the database further comprises an indication of all committed adaptive trade specifications in the database.

7. The system of claim 3, wherein the database further comprises:

an indication of all of the traders in the mall; and

for each of the adaptive trade specifications, an indication of one of the traders who submitted that adaptive trade specification.

8. The system of claim 4, wherein the database further comprises, for each of the traders in the mall, descriptive information about that trader.

9. The system of claim 4, wherein the database further comprises, for each of the traders in the mall, a set of all adaptive trade specifications that the traders has saved as drafts.

10. The system of claim 4, wherein the database further comprises, for each of the traders in the mall, a set of all adaptive trade specifications which are associated with the trader and which are in the committed state.

11. The system of claim 4, wherein the database further comprises, for each of the traders in the mall, an adaptive trade specification that is currently selected by the trader.

12. The system of claim 1, further comprising means for searching the database to locate matches among the adaptive trade specifications.

13. The system of claim 9, wherein the means for searching locates the matches between a single adaptive trade specification that is in the non-committed specification and the adaptive trade specifications that are in the committed state.

14. The system of claim 10, wherein the means for searching operates in four states:

(a) an adaptive trade specification management state in which a trader manages that trader's adaptive trade specifications;

(b) an adaptive trade specification filtering state in which the trader searches other traders' adaptive trade specifications;

(c) an adaptive trade specification matchmaking optimization state in which the means for searching locates at least one optimal match from among the other traders' adaptive trade specifications; and

(d) a transaction management state in which the trader makes a transaction with at least one of the other traders corresponding to the at least one optimal match.

12. A system for implementing an electronic auction house for a plurality of traders, the system comprising:

(a) a database server; and

(b) a database stored on the database server, the database comprising a plurality of adaptive trade specifications, each of the plurality of adaptive trade specifications comprising, for one of the traders:

(i) at least one give-item entry identifying at least one item that said one of the traders is willing to give in an auction; and

(ii) at least one take-item entry identifying at least one item that said one of the traders wants in return for the at least one item identified in the give-item entry;

the database further storing an identification, for each said auction, of an adaptive trade specification which is being auctioned and identifications of all of the adaptive trade specifications which have been placed as bids on the adaptive trade specification being auctioned.

13. The system of claim 12, wherein each of the plurality of adaptive trade specifications further comprises:

(iii) at least one constraint entry identifying at least one constraint that said one of the traders has placed on the auction; and

(iv) an objective entry identifying an objective sought by said one of the traders in the auction.

14. The system of claim 12, wherein the database further comprises a list of all active auctions.

15. The system of claim 14, wherein the database further comprises, for each auction in the list of all active auctions, an identification of a trader who initiated the auction.

16. The system of claim 15, wherein the database further comprises, for each auction in the list of all active auctions, an indication of whether the auction is a single-round or multiple-round auction.

17. The system of claim 15, wherein the database further comprises, for each auction in the list of all active auctions, an indication of whether the trader who initiated the auction wishes to be matched with only one winning bidder, all winning bidders, or at most a number K of winning bidders.

18. The system of claim 14, wherein the database further comprises a list of all active bids.

19. The system of claim 18, wherein the database further comprises, for each bid in the list of all active bids, an indication of an active trade specification corresponding to the bid.

20. The system of claim 18, wherein the database further comprises, for each bid in the list of all active bids, a set of all auctions against which the bid is submitted.

21. The system of claim 20, wherein the set of all auctions is ordered by time.

22. The system of claim 14, wherein the database further comprises a list of all active traders.

23. The system of claim 22, wherein the database further comprises, for each trader in the list of active traders, a list of active bids by that trader.

24. The system of claim 22, wherein the database further comprises, for each trader in the list of active traders, a list of active auctions by that trader.

25. The system of claim 12, further comprising means for searching the database to locate matches between the adaptive trade specifications.

26. The system of claim 25, wherein the means for searching locates the matches between a single adaptive trade specification which is not being auctioned and the adaptive trade specifications which are being auctioned.

27. The system of claim 26, wherein the means for searching operates in three states:

(a) an adaptive trade specification management state in which a trader manages that trader's adaptive trade specifications, initiates an auction and places a bid;

(b) an adaptive trade specification auction filtering state in which the trader searches other traders' adaptive trade specifications; and

(c) an adaptive trade specification auction management state in which the trader closes auctions.

28. A method for implementing an electronic mall for a plurality of traders, the method comprising:

(a) receiving a plurality of adaptive trade specifications, each of the plurality of adaptive trade specifications comprising, for one of the traders:

(i) at least one give-item entry identifying at least one item that said one of the traders is willing to give in an exchange;

(ii) at least one take-item entry identifying at least one item that said one of the traders wants in return for the at least one item identified in the give-item entry; and

(iii) an indication of whether the adaptive trade specification is in (A) a committed state in which the trader corresponding to the adaptive trade specification is committed to perform any transaction matching the adaptive trade specification or (B) a non-committed state in which the trader is free to initiate transactions; and

(b) storing the plurality of adaptive trade specifications in a database.

29. The method of claim 28, wherein each of the plurality of adaptive trade specifications further comprises:

(iv) at least one constraint entry identifying at least one constraint that said one of the traders has placed on the exchange; and

(v) an objective entry identifying an objective sought by said one of the traders in the exchange.

30. The method of claim 28, wherein the database further comprises an indication of all committed adaptive trade specifications in the database.

31. The method of claim 30, wherein the database further comprises:

an indication of all of the traders in the mall; and

for each of the adaptive trade specifications, an indication of one of the traders who submitted that adaptive trade specification.

32. The method of claim 31, wherein the database further comprises, for each of the traders in the mall, descriptive information about that trader.

33. The method of claim 31, wherein the database further comprises, for each of the traders in the mall, a set of all adaptive trade specifications that the traders has saved as drafts.

34. The method of claim 31, wherein the database further comprises, for each of the traders in the mall, a set of all adaptive trade specifications which are associated with the trader and which are in the committed state.

35. The method of claim 31, wherein the database further comprises, for each of the traders in the mall, an adaptive trade specification that is currently selected by the trader.

36. The method of claim 28, further comprising (c) searching the database to locate matches among the adaptive trade specifications.

37. The method of claim 36, wherein step (c) comprises locating the matches between a single adaptive trade specification that is in the non-committed specification and the adaptive trade specifications that are in the committed state.

38. The method of claim 36, wherein step (c) is performed in four states:

(i) an adaptive trade specification management state in which a trader manages that trader's adaptive trade specifications;

(ii) an adaptive trade specification filtering state in which the trader searches other traders' adaptive trade specifications;

(iii) an adaptive trade specification matchmaking optimization state in which the means for searching locates at least one optimal match from among the other traders' adaptive trade specifications; and

(iv) a transaction management state in which the trader makes a transaction with at least one of the other traders corresponding to the at least one optimal match.

39. A method for implementing an electronic auction house for a plurality of traders, the method comprising:

(a) receiving a plurality of adaptive trade specifications, each of the plurality of adaptive trade specifications comprising, for one of the traders:

(i) at least one give-item entry identifying at least one item that said one of the traders is willing to give in an auction; and

(ii) at least one take-item entry identifying at least one item that said one of the traders wants in return for the at least one item identified in the give-item entry;

(b) storing the plurality of adaptive trade specifications in the database; and

(c) storing in the database an identification, for each said auction, of an adaptive trade specification which is being auctioned and identifications of all of the adaptive trade specifications which have been placed as bids on the adaptive trade specification being auctioned.

40. The method of claim 39, wherein each of the plurality of adaptive trade specifications further comprises:

(iii) at least one constraint entry identifying at least one constraint that said one of the traders has placed on the auction; and

(iv) an objective entry identifying an objective sought by said one of the traders in the auction.

41. The method of claim 39, wherein the database further comprises a list of all active auctions.

42. The method of claim 41, wherein the database further comprises, for each auction in the list of all active auctions, an identification of a trader who initiated the auction.

43. The method of claim 42, wherein the database further comprises, for each auction in the list of all active auctions, an indication of whether the auction is a single-round or multiple-round auction.

44. The method of claim 42, wherein the database further comprises, for each auction in the list of all active auctions, an indication of whether the trader who initiated the auction wishes to be matched with only one winning bidder, all winning bidders, or at most a number K of winning bidders.

45. The method of claim 41, wherein the database further comprises a list of all active bids.

46. The method of claim 45, wherein the database further comprises, for each bid in the list of all active bids, an indication of an active trade specification corresponding to the bid.

47. The method of claim 45, wherein the database further comprises, for each bid in the list of all active bids, a set of all auctions against which the bid is submitted.

48. The method of claim 47, wherein the set of all auctions is ordered by time.

49. The method of claim 41, wherein the database further comprises a list of all active traders.

50. The method of claim 49, wherein the database further comprises, for each trader in the list of active traders, a list of active bids by that trader.

51. The method of claim 49, wherein the database further comprises, for each trader in the list of active traders, a list of active auctions by that trader.

52. The method of claim 39, further comprising (d) searching the database to locate matches between the adaptive trade specifications.

53. The method of claim 52, wherein step (d) comprises locating the matches between a single adaptive trade specification which is not being auctioned and the adaptive trade specifications which are being auctioned.

54. The method of claim 53, wherein step (d) is performed in three states:

(i) an adaptive trade specification management state in which a trader manages that trader's adaptive trade specifications, initiates an auction and places a bid;

(ii) an adaptive trade specification auction filtering state in which the trader searches other traders' adaptive trade specifications; and

(iii) an adaptive trade specification auction management state in which the trader closes auctions.

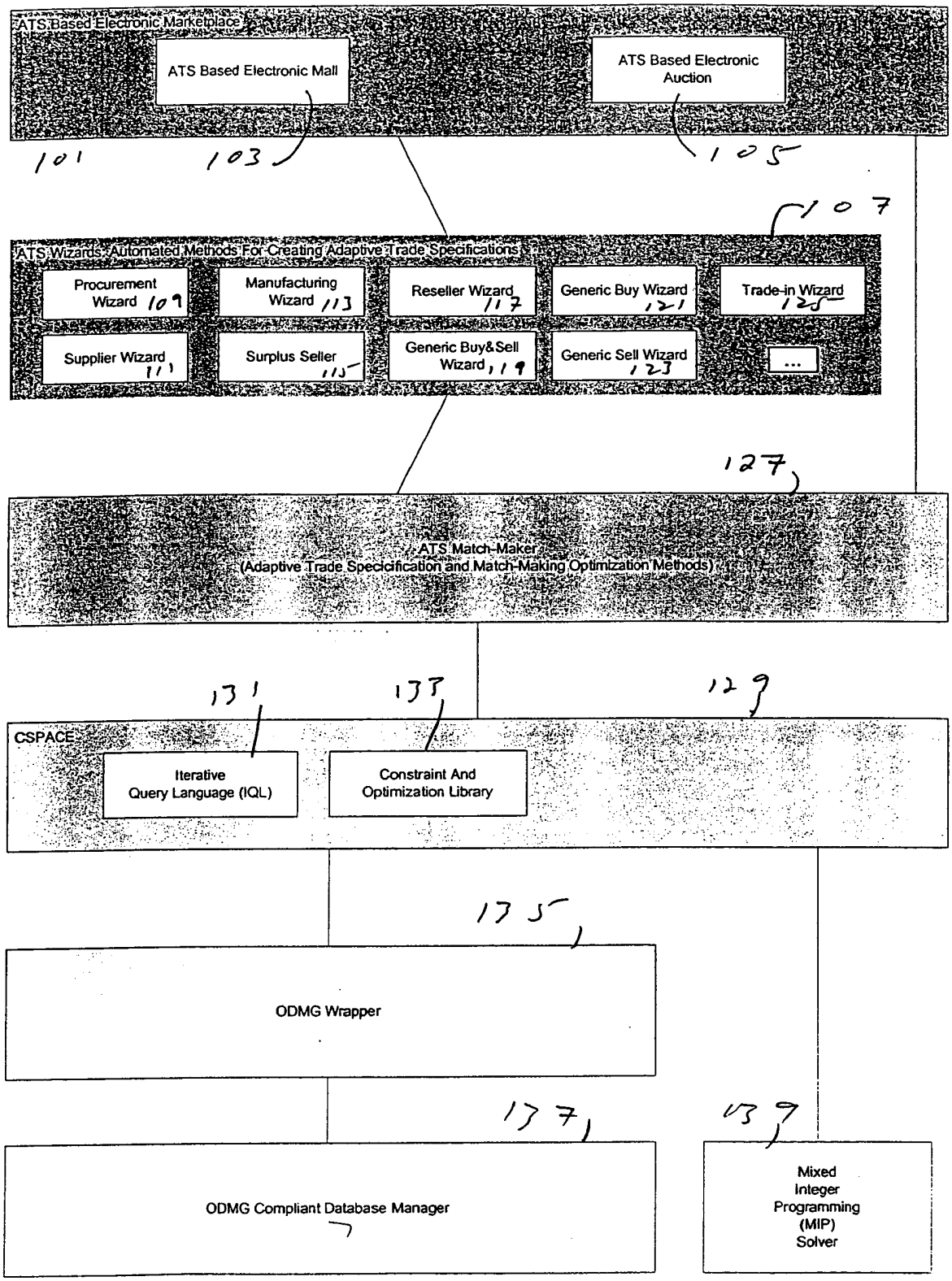

ATS Based Electronic Marketplace
ATS Based Electronic Mall
ATS Based Electronic Auction
101
103
105
107
ATS Wizards - Automated Methods For Creating Adaptive Trade Specifications
Procurement Wizard 109
Manufacturing Wizard 113
Reseller Wizard 117
Generic Buy Wizard 121
Trade-in Wizard 125
Supplier Wizard 111
Surplus Seller 115
Generic Buy&Sell Wizard 119
Generic Sell Wizard 123
...
127
ATS Match-Maker
(Adaptive Trade Specification and Match-Making Optimization Methods)
131
133
129
CSPACE
Iterative Query Language (IQL)
Constraint And Optimization Library
135
ODMG Wrapper
137
139
ODMG Compliant Database Manager
Mixed Integer Programming (MIP) Solver

Fig. 1

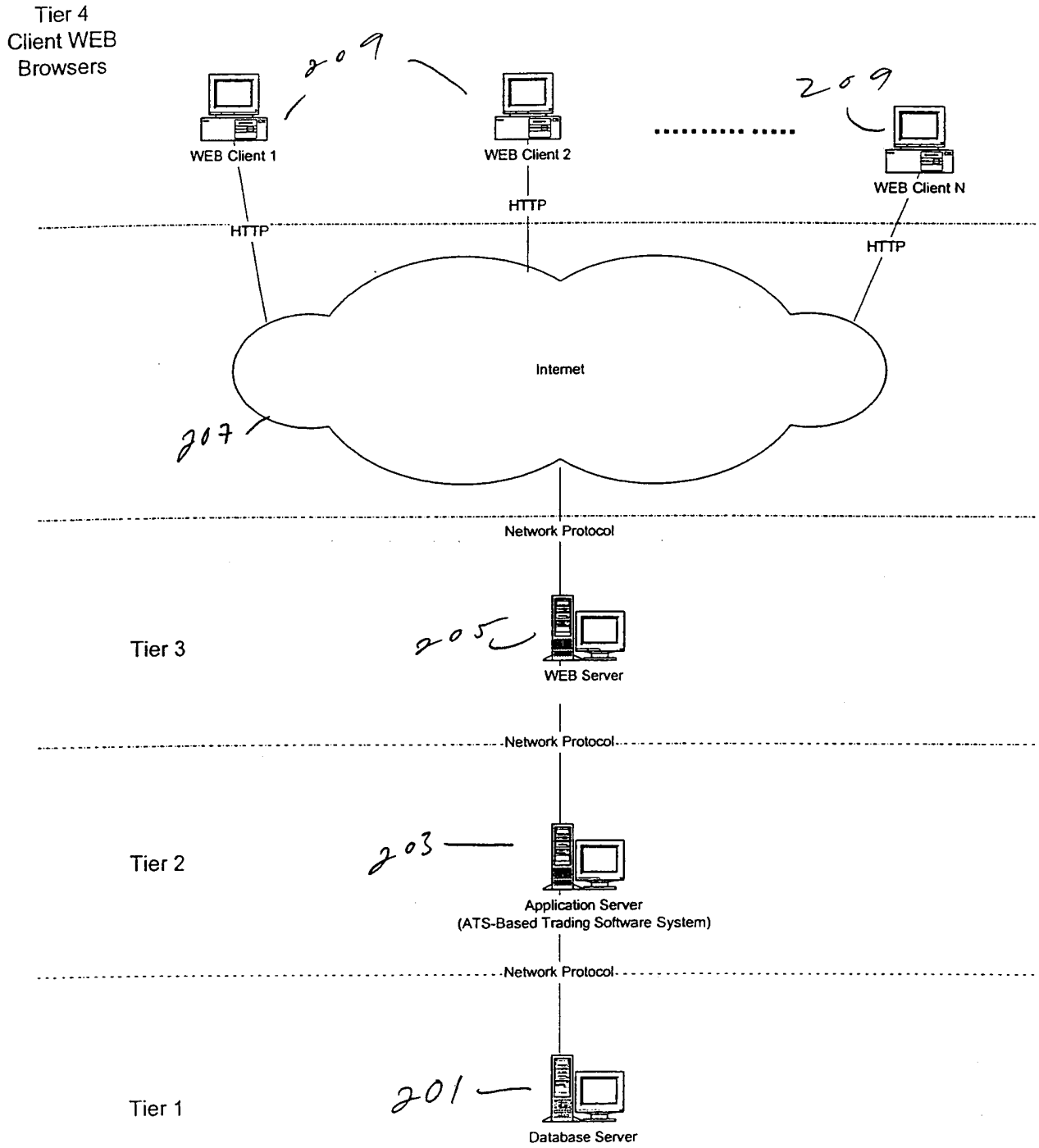

Tier 4
Client WEB
Browsers
209
209
WEB Client 1
WEB Client 2
WEB Client N
HTTP
HTTP
HTTP
Internet
207
Network Protocol
Tier 3
205
WEB Server
Network Protocol
Tier 2
203
Application Server
(ATS-Based Trading Software System)
Network Protocol
Tier 1
201
Database Server

FIG. 2

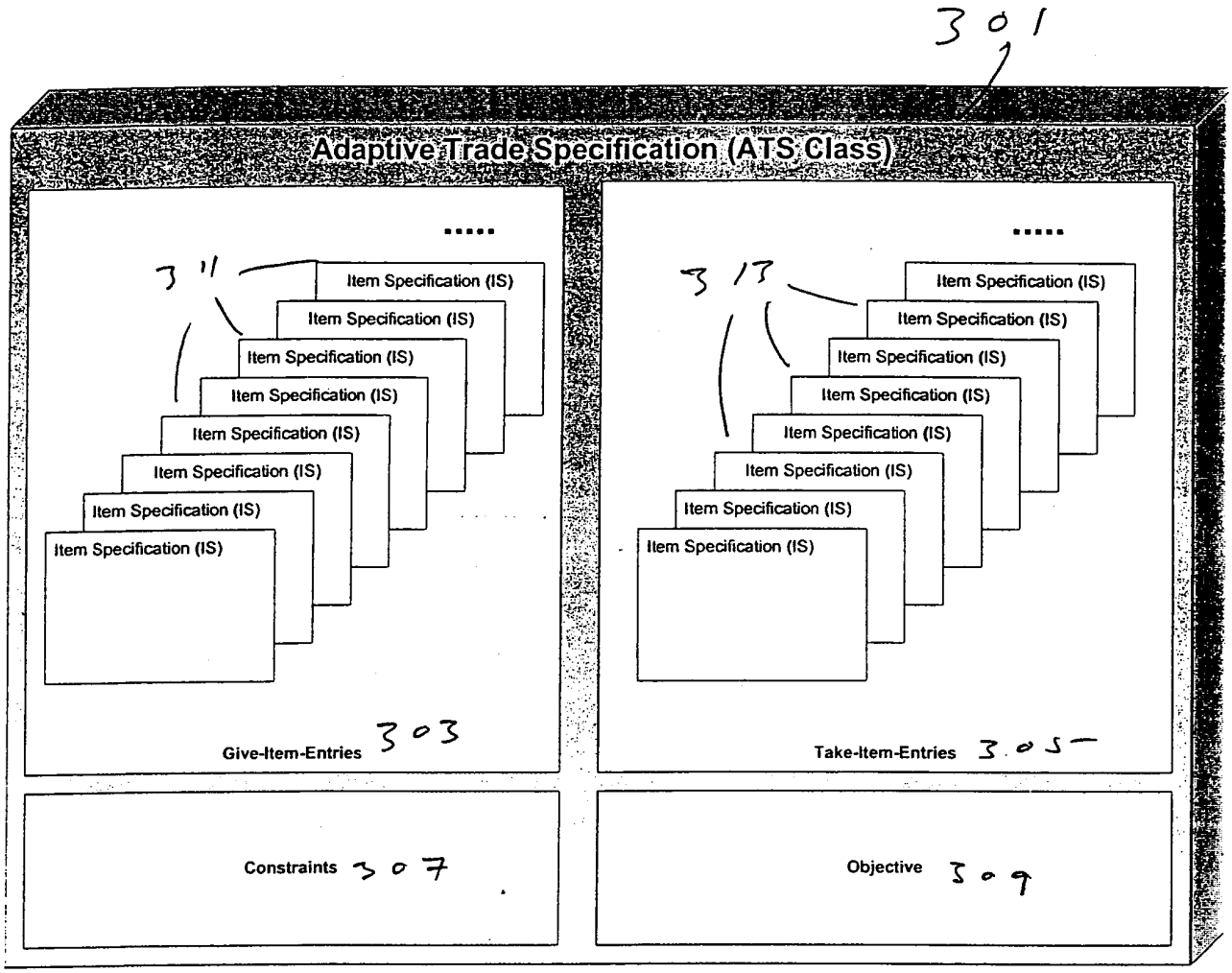

301
Adaptive Trade Specification (ATS Class)
311
Item Specification (IS)
Item Specification (IS)
Item Specification (IS)
Item Specification (IS)
Item Specification (IS)
Item Specification (IS)
Item Specification (IS)
Item Specification (IS)
Give-Item-Entries 303
317
Item Specification (IS)
Item Specification (IS)
Item Specification (IS)
Item Specification (IS)
Item Specification (IS)
Item Specification (IS)
Item Specification (IS)
Item Specification (IS)
Take-Item-Entries 305
Constraints 307
Objective 309

FIG. 3

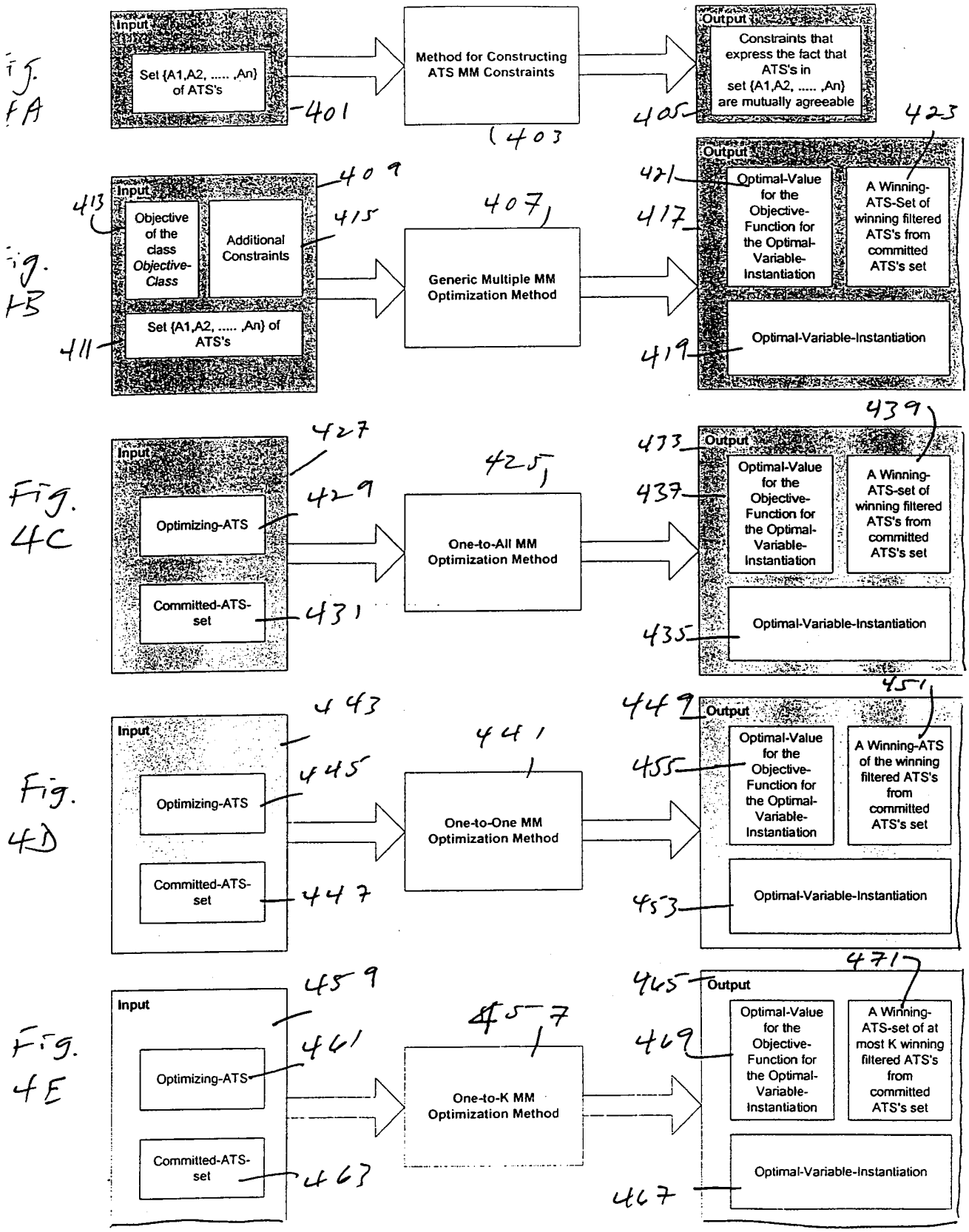

Input
Set {A1,A2, ... ,An} of ATS's
401
Method for Constructing ATS MM Constraints
403
Output
Constraints that express the fact that ATS's in set {A1,A2, ... ,An} are mutually agreeable
405
Fig. 4A
Input
Objective of the class Objective-Class
413
Additional Constraints
415
Set {A1,A2, ... ,An} of ATS's
411
409
Generic Multiple MM Optimization Method
407
Output
Optimal-Value for the Objective-Function for the Optimal-Variable-Instantiation
421
A Winning-ATS-Set of winning filtered ATS's from committed ATS's set
423
417
Optimal-Variable-Instantiation
419
Fig. 4B
Input
Optimizing-ATS
427
429
Committed-ATS-set
431
One-to-All MM Optimization Method
425
Output
Optimal-Value for the Objective-Function for the Optimal-Variable-Instantiation
433
A Winning-ATS-set of winning filtered ATS's from committed ATS's set
439
437
Optimal-Variable-Instantiation
435
Fig. 4C
Input
Optimizing-ATS
443
445
Committed-ATS-set
447
One-to-One MM Optimization Method
441
Output
Optimal-Value for the Objective-Function for the Optimal-Variable-Instantiation
449
A Winning-ATS of the winning filtered ATS's from committed ATS's set
451
455
Optimal-Variable-Instantiation
453
Fig. 4D
Input
Optimizing-ATS
459
461
Committed-ATS-set
463
One-to-K MM Optimization Method
457
Output
Optimal-Value for the Objective-Function for the Optimal-Variable-Instantiation
465
A Winning-ATS-set of at most K winning filtered ATS's from committed ATS's set
471
469
Optimal-Variable-Instantiation
467
Fig. 4E

Fig. 5A

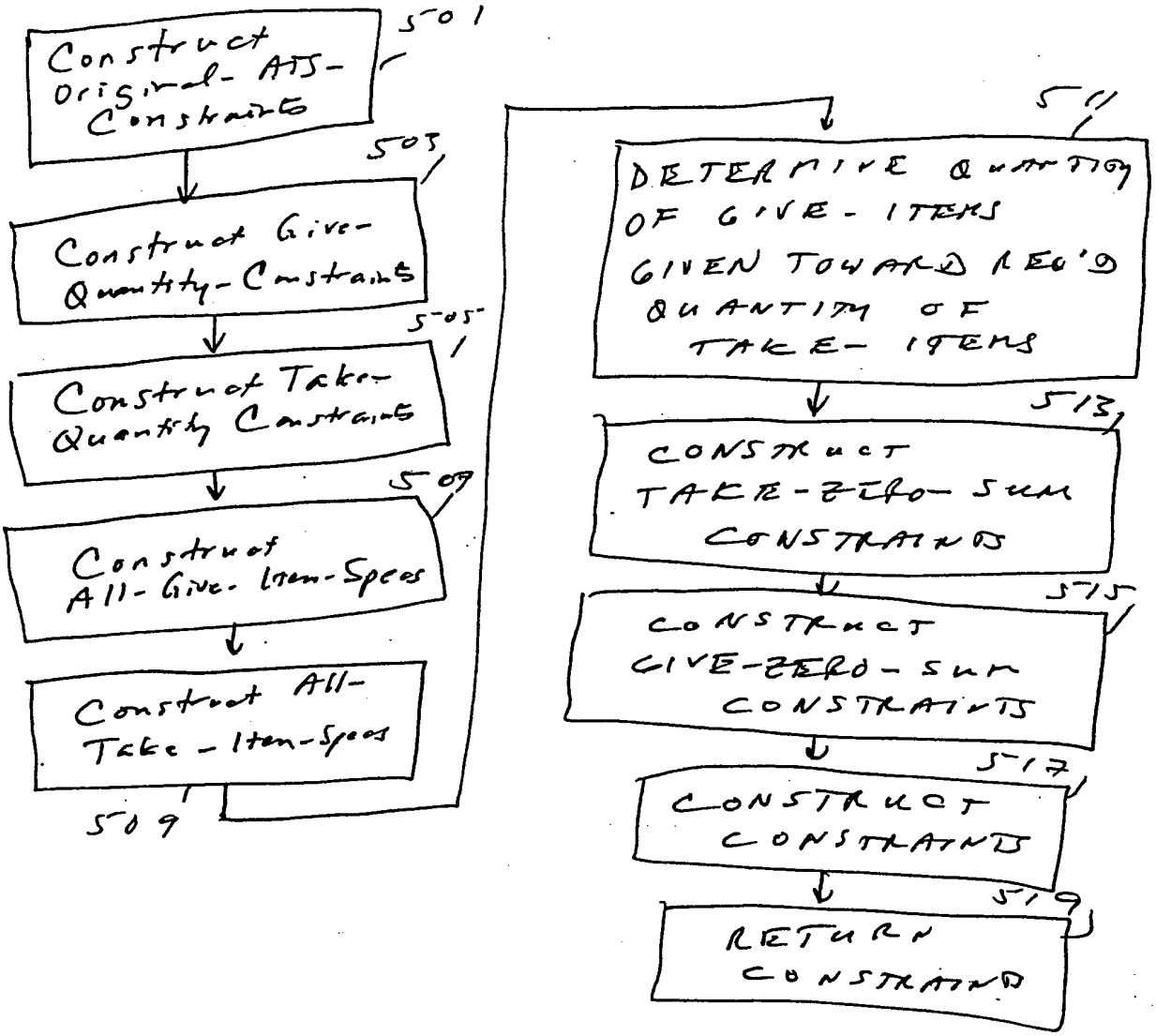

Construct Original-ATS-Constraints
501
Construct Give-Quantity-Constraints
503
Construct Take-Quantity Constraints
505
Construct All-Give-Item-Specs
507
Construct All-Take-Item-Specs
509
DETERMINE QUANTITY OF GIVE-ITEMS GIVEN TOWARD REQ'D QUANTITY OF TAKE-ITEMS
511
CONSTRUCT TAKE-ZERO-SUM CONSTRAINTS
513
CONSTRUCT GIVE-ZERO-SUM CONSTRAINTS
515
CONSTRUCT CONSTRAINTS
517
RETURN CONSTRAINTS
519

Fig. 5B

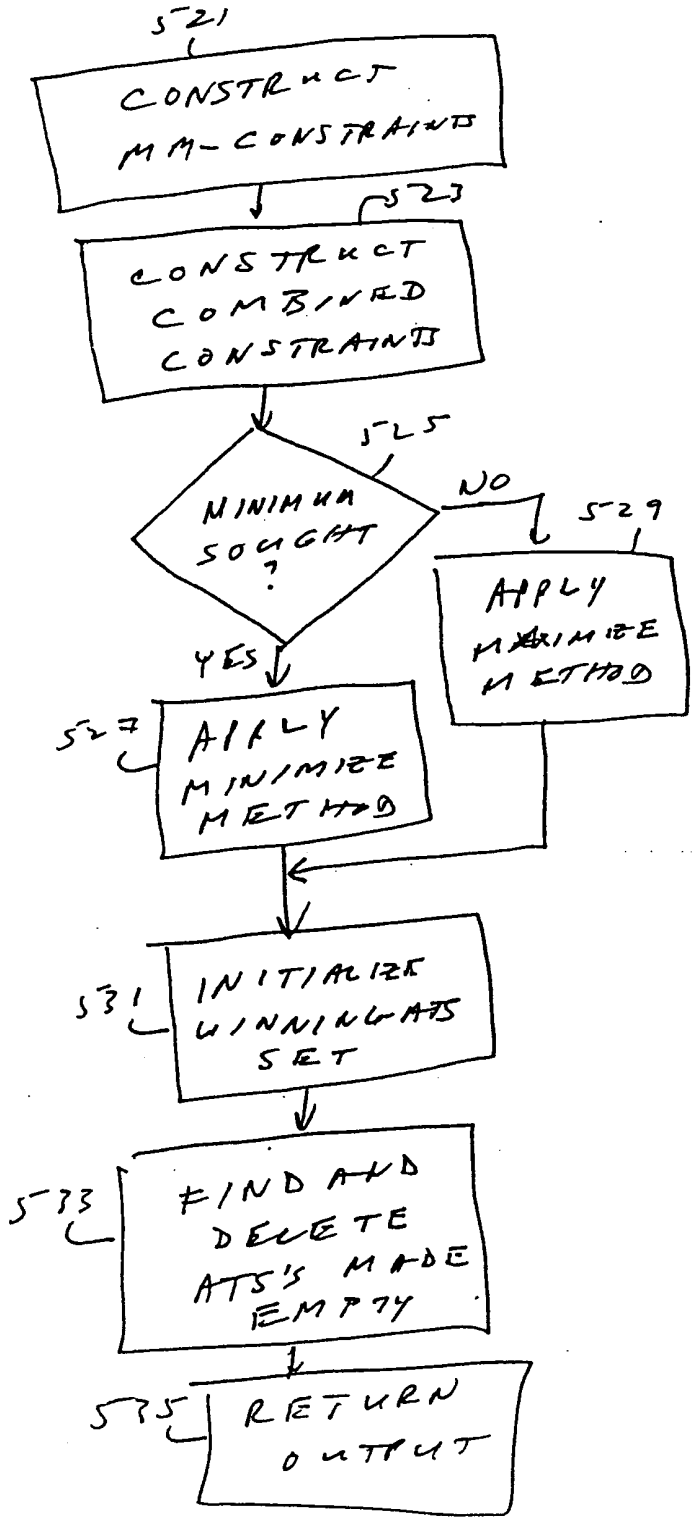

521
CONSTRUCT MM-CONSTRAINTS
523
CONSTRUCT COMBINED CONSTRAINTS
525
MINIMUM SOUGHT ?
NO
529
APPLY MAXIMIZE METHOD
YES
527
APPLY MINIMIZE METHOD
531
INITIALIZE WINNING ATS SET
533
FIND AND DELETE ATS's MADE EMPTY
535
RETURN OUTPUT

Fig. 5C

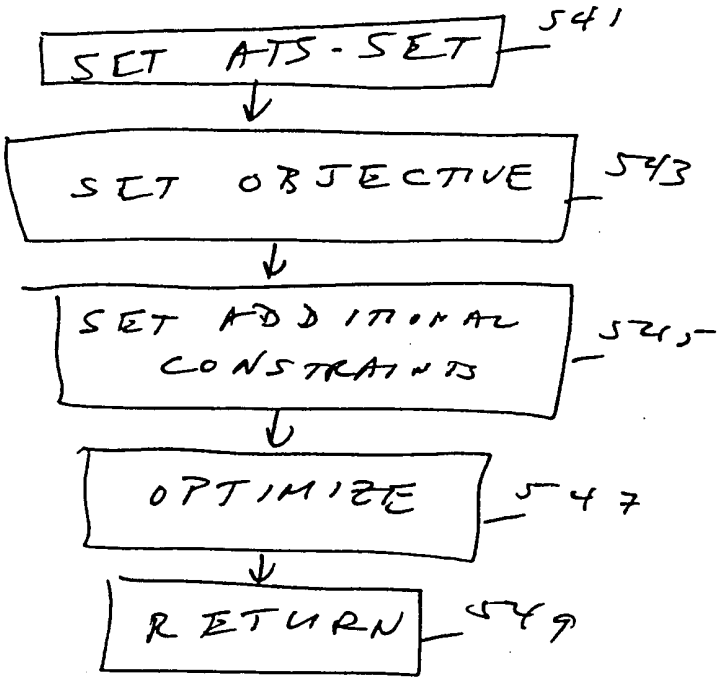

SET ATS-SET
541
SET OBJECTIVE
543
SET ADDITIONAL CONSTRAINTS
545
OPTIMIZE
547
RETURN
549

Fig. 5D

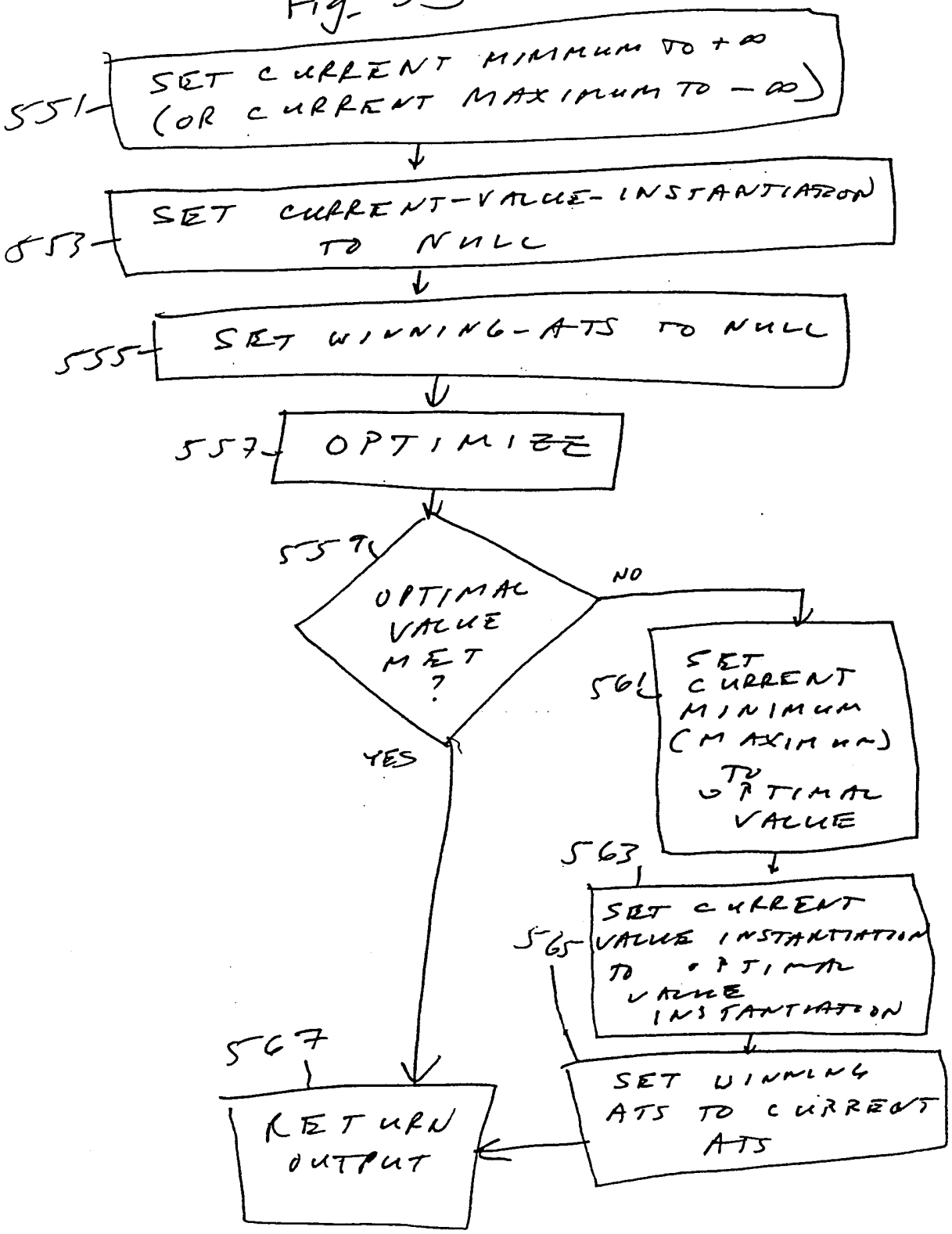

551
SET CURRENT MINIMUM TO +∞ (OR CURRENT MAXIMUM TO −∞)
553
SET CURRENT-VALUE-INSTANTIATION TO NULL
555
SET WINNING-ATS TO NULL
557
OPTIMIZE
559
OPTIMAL VALUE MET ?
NO
YES
561
SET CURRENT MINIMUM (MAXIMUM) TO OPTIMAL VALUE
563
565
SET CURRENT VALUE INSTANTIATION TO OPTIMAL VALUE INSTANTIATION
SET WINNING ATS TO CURRENT ATS
567
RETURN OUTPUT

Fig. 5E

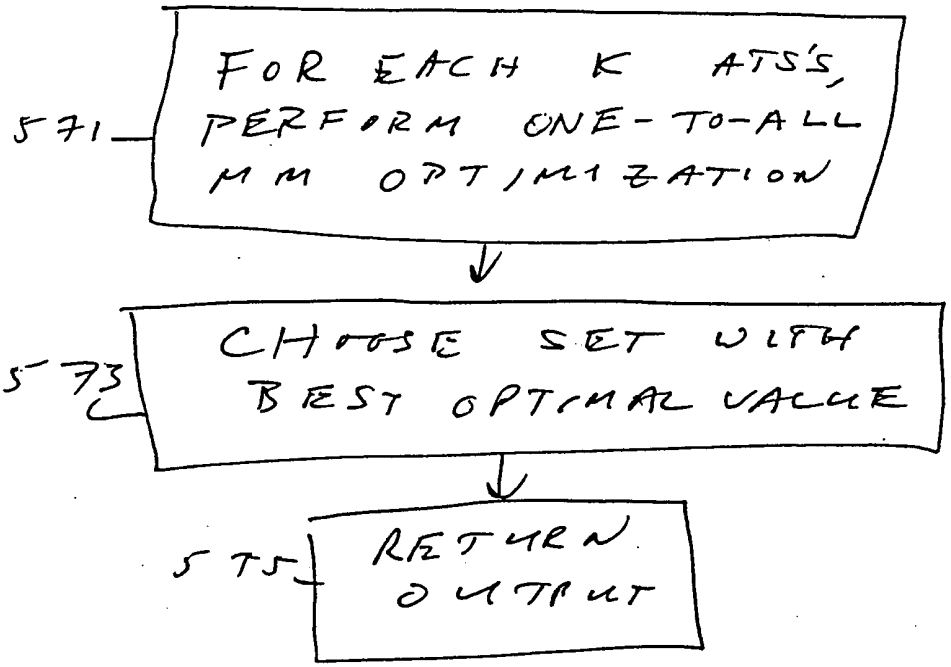

FOR EACH K ATS'S, PERFORM ONE-TO-ALL MM OPTIMIZATION
571
CHOOSE SET WITH BEST OPTIMAL VALUE
573
RETURN OUTPUT
575

Fig. 6

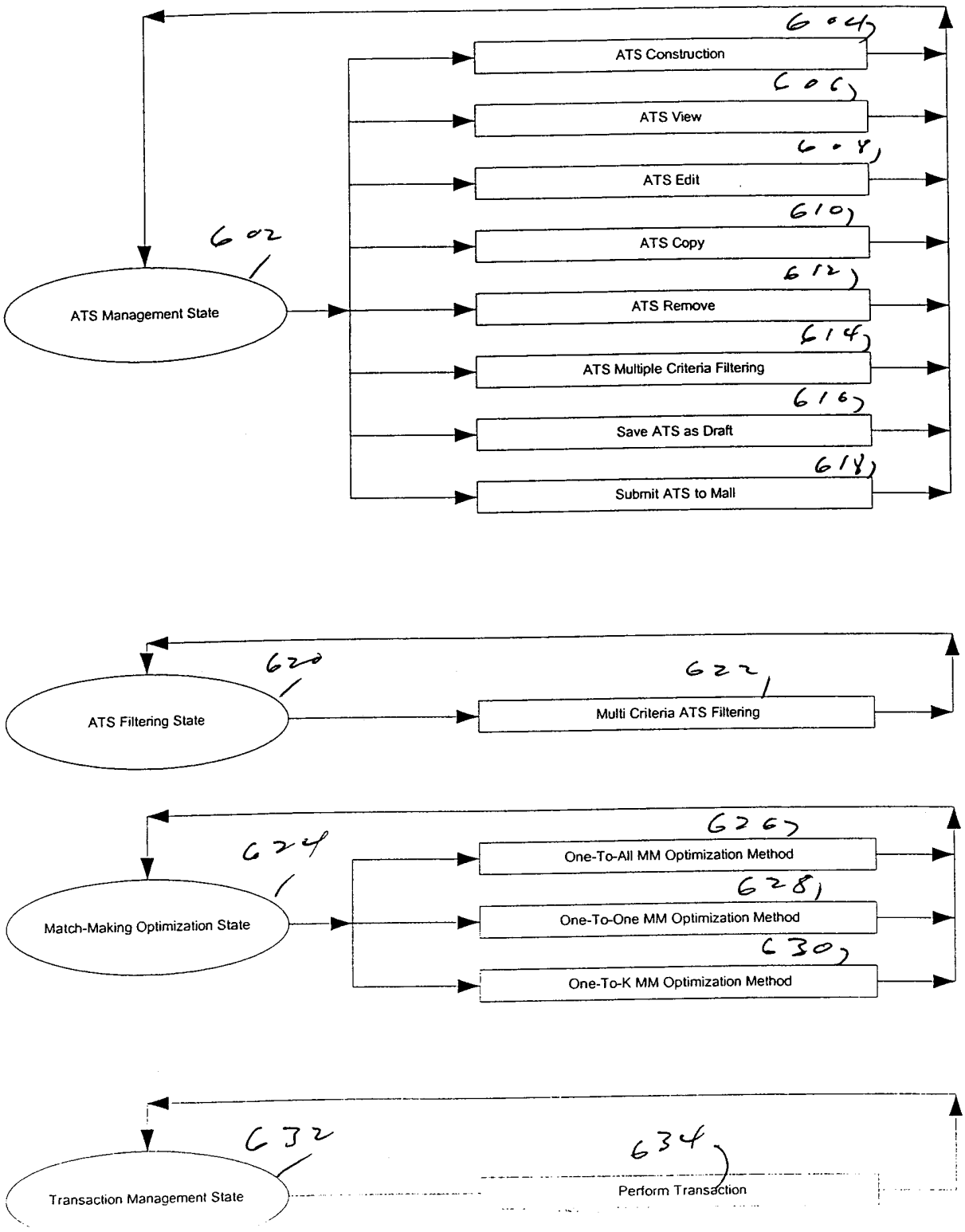
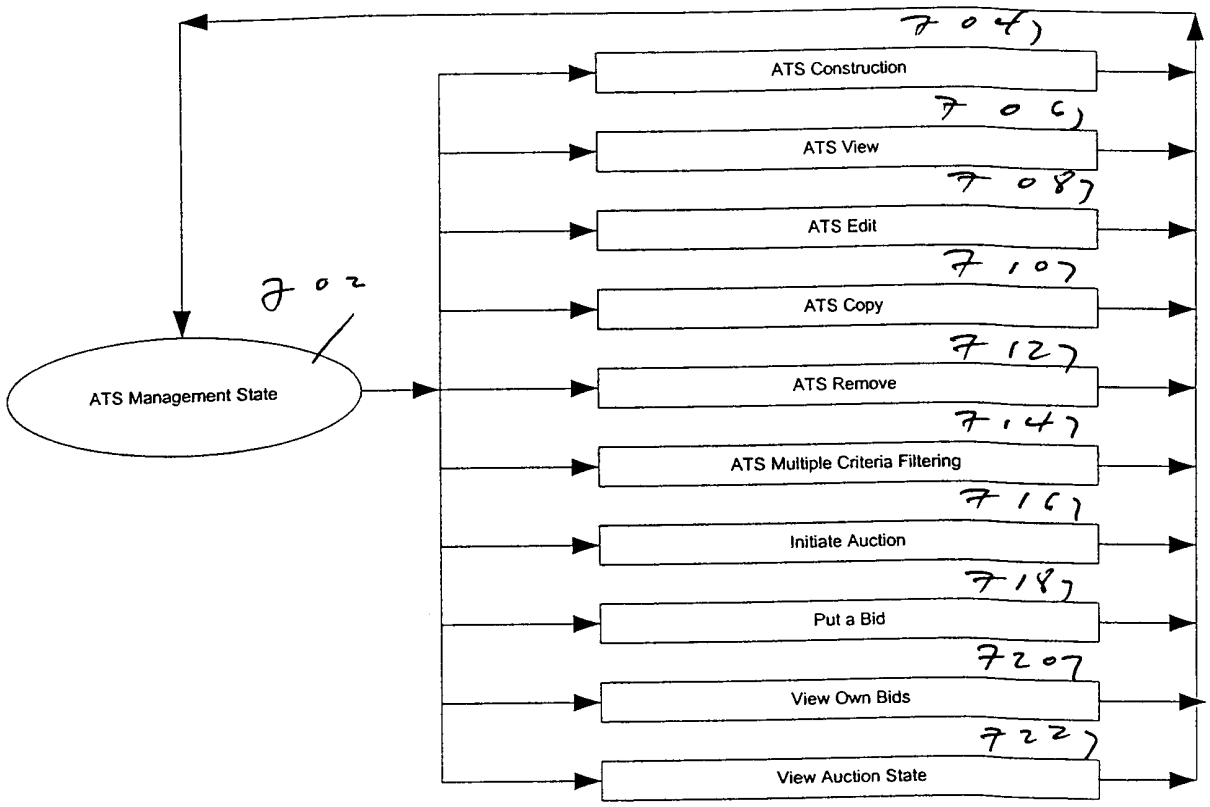
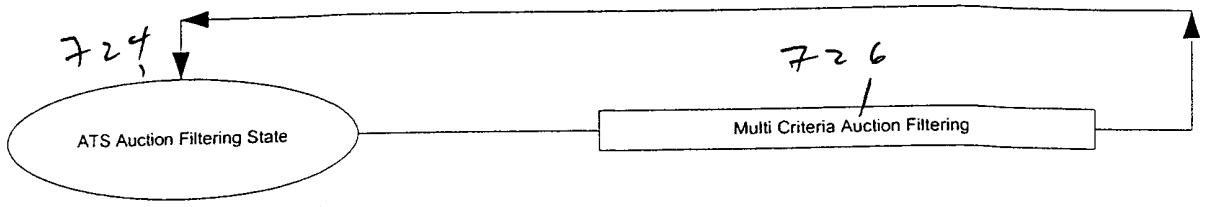
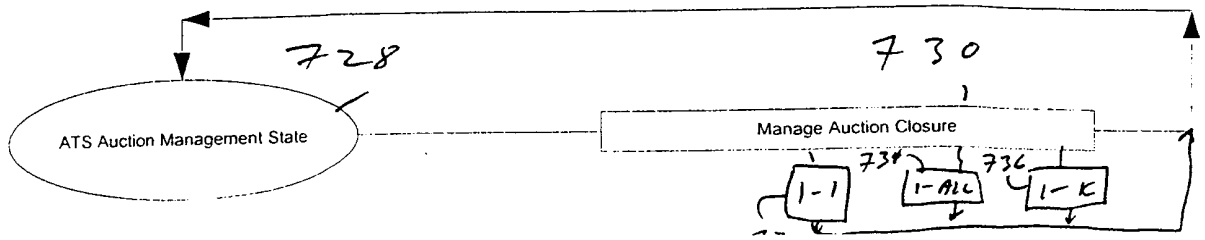

602
ATS Management State
604
ATS Construction
606
ATS View
608
ATS Edit
610
ATS Copy
612
ATS Remove
614
ATS Multiple Criteria Filtering
616
Save ATS as Draft
618
Submit ATS to Mail
620
ATS Filtering State
622
Multi Criteria ATS Filtering
624
Match-Making Optimization State
626
One-To-All MM Optimization Method
628
One-To-One MM Optimization Method
630
One-To-K MM Optimization Method
632
Transaction Management State
634
Perform Transaction

Fig. 7


704
ATS Construction
706
ATS View
708
ATS Edit
710
ATS Copy
712
ATS Remove
714
ATS Multiple Criteria Filtering
716
Initiate Auction
718
Put a Bid
720
View Own Bids
722
View Auction State
702
ATS Management State


724
ATS Auction Filtering State
726
Multi Criteria Auction Filtering


728
ATS Auction Management State
730
Manage Auction Closure
734
736
1-1
1-AUC
1-K