Agent-Based Service Composition Through Simultaneous Negotiation in Forward and Reverse Auctions

Chris Preist
HP labs
Filton Rd, Stoke Gifford
Bristol, BS32 8QZ
United Kingdom
chris.preist@hp.com

Claudio Bartolini
HP labs
1501 Page Mill Rd
Mailstop 1U-1140
Palo Alto, CA 94304-1140
United States
claudio.bartolini@hp.com

Andrew Byde
HP labs
Filton Rd, Stoke Gifford
Bristol, BS32 8QZ
United Kingdom
andrew.byde@hp.com

ABSTRACT

Service composition is the act of taking several component products or services, and bundling them together to meet the needs of a given customer. In the future, service composition will play an increasingly important role in e-commerce, and automation will be desirable to improve speed and efficiency of customer response. In this paper, we consider a service composition agent that both buys components and sells services through auctions. It buys component services by participating in many English auctions. It sells composite services by participating in Request-for-Quotes reverse auctions. Because it does not hold a long-term inventory of component services, it must take risks; it must make offers in reverse auctions prior to purchasing all the components needed, and must bid in English auctions prior to having a guaranteed customer for the composite good. We present algorithms that is able to manage this risk, by appropriately bidding/offering in many auctions and reverse auctions simultaneously. The algorithms will withdraw from one set of possible auctions and move to another set if this will produce a better-expected outcome, but will effectively manage the risk of accidentally winning outstanding bids/offers during the withdrawal process. We illustrate the behavior of these algorithms through a set of worked examples.

Categories and Subject Descriptors

I.2.1 [**Artificial Intelligence**]: Applications and Expert Systems; I.2.8 [**Artificial Intelligence**]: Problem Solving, Control Methods, and Search

General Terms

Algorithms

Keywords

Service Composition, Auctions, Bidding, Agent

EC'03, June 9–12, 2003, San Diego, California, USA.

1. INTRODUCTION

Over the past few years, Electronic Commerce has become an increasingly central part of the economy. An Internet presence is considered an essential part of doing business, rather than an exotic add-on to a company. More and more transactions, both from business to consumer and between businesses, are taking place online. Simple fixed cost business transactions are often automated at one or both ends, and auctions are overwhelmingly conducted by automated auctioneer software. Agent technology has been proposed as a means of automating some of the more sophisticated negotiations which businesses are involved in (e.g. [11]). In this paper we look at a specific class of business process that will become increasingly important in the virtual economy - service composition. We focus specifically on algorithms that simultaneously purchase component services from a group of auctions, and sell composite services via reverse auctions.

Over the last decade, companies have focused more and more on their core competencies, by outsourcing non-core activities to other companies. This trend is beginning to have an impact on many E-businesses, as well as traditional bricks-and-mortar companies. Companies would like to be able to outsource some of their activities over the Internet. Initially, this has focused on semi-permanent arrangements, with the web acting as an intermediary. For example, career guidance information can be provided to employees of a large company via a web-based third party. However, as this trend is becoming increasingly important, much research and development effort has been focusing on a more dynamic, service-centric view of the Internet. Web Services are virtual entities that provide a service over the network through an open standard interface. The service may be information, such as the latest stock prices, or may be a virtual representation of some physical good or activity, such as a contract to transport a crate from one location to another. Because the service is offered through an open standard interface, any client familiar with this standard can use it. Furthermore, the output from one service can be fed directly into another service. This makes the creation of composite services and complex business processes which cross-organizational boundaries possible. Potentially, this can be done automatically and dynamically, and agent technology will play a key role in this.

This leads to the emergence of an important role in the virtual economy - the service composer. As companies focus on their core competencies, other companies can focus on creating composite packages. This is not new - travel agents, among others, have done exactly that for years - but what is new is that it will be able to take place dynamically, automatically, over the Internet.

In this paper, we consider a service composition agent that both buys components and sells services through auctions. It buys component services by participating in many English auctions. It sells composite services by participating in Request-for-Quotes reverse auctions. Because it does not hold a long-term inventory of component services, it must take risks; it must make offers in reverse auctions prior to purchasing all the components needed, and must bid in English auctions prior to having a guaranteed customer for the composite good. In section §2, we describe the problems facing such an agent, and in section §3, we present an example service composition scenario involving a virtual company, "FreightMixer". In §4, we present algorithms which is able to manage this risk, by appropriately bidding/offering in many auctions and reverse auctions simultaneously. The algorithms will withdraw from one set of possible auctions and move to another set if this will produce a better-expected outcome, but will effectively manage the risk of accidentally winning outstanding bids/offers during the withdrawal process. In section §4.5 we illustrate the behavior of these algorithms through a set of worked examples. We then discuss related work (Section §5) and present our conclusions and future work (Section §6).

2. SERVICE COMPOSITION

Service composition is the act of purchasing several *component services*, combining them, and selling them as a single composite service. The *service composer* responsible for the generation of the composite service must purchase the component services from a group of *suppliers* and will sell the composite service to one or more *customers*. In §3, we will give a detailed example of a company responsible for shipping freight. The company, FreightMixer, is the service composer. It competes in Request-For-Quotes reverse auctions to supply customers with freight shipment services, for example from London to San Francisco. However, it does not own any cargo facilities of its own. Instead, it subcontracts, and arranges cargo space on a set of linked flights. The airlines running these flights are the suppliers, and the individual flights are the component services.

A service composer, therefore, may be simultaneously interacting with many potential customers and many potential suppliers. This affects its behavior throughout the business lifecycle (see [19] for more details). In this paper, we focus particularly on the decision problem it faces during negotiation. To operate effectively, the service composer will be involved in many interlinked negotiations, and must make tradeoffs between them as they progress. It will be negotiating with one or more potential purchasers in an effort to agree a price to supply a given service. While it is doing this, it will need to determine what bundle of component services are required to meet each customers needs, and negotiate with suppliers of these component services to determine what price to purchase them at. For any single bundle of base service types the composer will be involved in at least one, but more likely many, negotiations to acquire instances of each service type. Furthermore, it may be possible to meet the customer's needs with a variety of alternative bundles. The service composer may simultaneously negotiate to purchase alternative bundles, in an effort to find which bundle is best.

In an ideal world, from the service composer's perspective, it would be able to make provisional agreements in all these negotiations without making any binding commitment. This would allow it to negotiate a provisional price with a customer and then go to the suppliers to negotiate the cheapest bundle of components to provide that service. If the profit margin between the provisional sale price and the proposed bundle purchase price was adequate, it would make an agreement with the suppliers of that bundle and confirm the agreement with the customer. By doing this, the service composer is never at risk of purchasing components it cannot use, or of committing to a composite service it cannot provide.

However, intermediaries are rarely in a position to dictate the way in which negotiation will be carried out (i.e. the market mechanism). Nowadays, reverse auctions are becoming increasingly popular with large customers, to encourage visible competition within a group of potential suppliers of a service. In a reverse auction, potential suppliers must offer prices, undercutting the lowest current offer. An offer is binding if the auction closes and it is the lowest. If a composer places an offer prior to securing the required components, there is a risk that it may win and be forced to supply the service at a loss or default on the contract.

Component services may be available at fixed, guaranteed, prices. However, there is an increasingly large market selling surplus goods and services off cheaply via English auctions. If a composer does not use such a source, then a competitor may be able to undercut its prices. However, if it does participate in auctions, any bid is a potential commitment to purchase. Hence, placing such a bid is a risk if it does not have a guaranteed customer. A service composer therefore needs to place bids in auctions, offers in reverse auctions, and make purchases from fixed-price suppliers in such a way as to balance risk against potential gain. By making an offer in a reverse auction before securing the necessary component services or placing a bid for a component before having a guaranteed sale, the service composer takes a risk. However by doing so it may gain by having an increased chance of winning a deal or a reduced cost of securing a component. The algorithm presented in this paper is designed to make this tradeoff, and adjust appropriately as negotiation progresses.

3. FREIGHTMIXER: AN EXAMPLE SCENARIO

As a motivating example, we now present a scenario that demonstrates the need for sophisticated simultaneous negotiation in both forward and reverse auctions. FreightMixer is an imaginary transport company that ships goods around the world on behalf of customers. Unlike some other transport companies, it owns no transport infrastructure. Instead, it exploits cheap last-minute sales of excess hold space to meet the needs of its customers. Because it cannot guarantee that hold space will be available on a particular route, it may connect together several flight legs to get the package to its destination. While it may not be the quickest service, it aims to be the cheapest. Electronic marketplaces are both a source of resources (individual flight legs) and a channel for products (composite flights for a given customer).

FreightMixer acts in two distinct sets of markets:

In the markets for end-to-end cargo services, it acts as a potential seller. It observes the advertised requirements of potential customers. Usually, customers will conduct reverse auctions (or Request For Quotes, which are informal reverse auctions).

In the markets for hold space on flights (and possibly ships), it acts as a potential buyer. It observes the availability and cost of different options in these markets. Some sellers will sell hold space at a fixed price, while others will conduct English auctions to dispose of excess space.

In its role as service composer, it must (a) understand requirements of the potential customers which are currently requesting services in the end-to-end cargo markets, and identify a service which could meet their needs (b) identify the alternative ways this service can be created from component services (i.e. hold space on specific flights) (c) identify potential sellers of these component services in the markets for hold space on flights.

By analyzing the various markets, FreightMixer can determine potential trading options. Each option will consist of the following:

- A potential buyer, or set of buyers, who are currently requesting a service in the end-to-end cargo marketplaces.
- A service specification which meets the needs of these buyers.
- One or more alternate decompositions of this service into component services.
- A list of sellers in the markets for hold space who are offering to sell individual component services appearing in these decompositions.

For example, suppose that FreightMixer observes a Request For Quotes reverse auction for sending a 1 tonne crate from London to San Francisco, with the best offer currently at 230, and a second reverse auction for sending a 1 tonne crate from London to Los Angeles, with the best offer currently at 260. Using its database of service models, Freightmixer identifies alternative combinations of flights that might potentially meet these needs. It identifies a direct route from London(LHR) to San Francisco(SFO), and also identifies alternative routes via Chicago(ORD), New York(JFK) and Boston(BOS). It finds no appropriate direct flight from LHR to Los Angeles(LAX), but finds flights from Chicago to Los Angeles and from San Francisco to Los Angeles. It then checks the auctions for excess hold space and finds that appropriate auctions exist for all legs except LHR to JFK. Each auction is for exactly 1 tonne of hold space. The alternatives it has are shown in Figure 1.

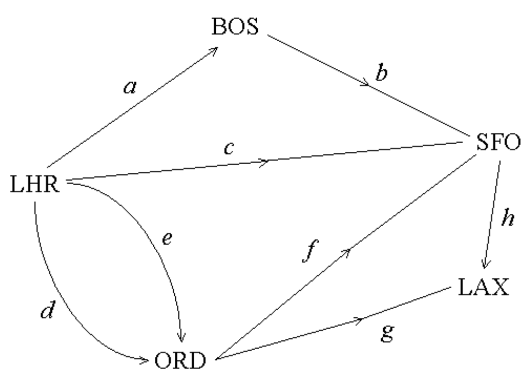

Figure 1: Graph of services

Hence, FreightMixer has an option consisting of the reverse auction for a shipment {LHR-SFO}, three alternative ways of generating the required service from component services ({LHR-SFO}, {LHR-ORD & ORD-SFO} or {LHR-BOS & BOS-SFO}) and potential sellers for each of the component services. It also has an option consisting of the reverse auction for a shipment {LHR-LAX}, and 4 alternative ways for generating the component services (e.g. {LHR-SFO & SFO-LAX} or {LHR-ORD & ORD-LAX}).

FreightMixer must now decide whether to enter into negotiation with the potential customers and potential suppliers, and if so, which offers and bids should it place. Should it attempt to win both contracts, or just one? How should it place its bids and offers to maximize the likely profit? In the following section, we present an algorithm that is able to make such decisions. Then, in section §4.5, we return to our example to illustrate the decisions the algorithm makes.

4. SPECIFICATION OF A NEGOTIATION ALGORITHM FOR PURCHASING SERVICE BUNDLES

We now present an algorithm to automate the decision problem faced by Freightmixer and other service composers. We make certain assumptions on the nature of the environment in which it operates:

- We assume that all transactions take place through English auctions, reverse auctions and fixed-price sales. We do not currently consider one-to-one negotiation or double auctions.
- We assume that there are a sufficiently large number of buyers and sellers, and sufficient stability in the environment, to allow statistical profiles of expected outcomes in auctions to be built.

We first present the decision problem the agent is faced with and then present the specification of an algorithm to perform service composition in this environment. We also present a simplified form which is more efficient, together a with pseudo-code design.

4.1 Specification of the Decision Problem

We assume our agent is participating in a set of auctions, $\mathcal{A}$. These auctions all start at roughly the same time, but may finish at different times. Some of these auctions are forward auctions, selling a single good or service, while others are reverse auctions, requiring a single good or service.

The forward auctions are English auctions with a fixed closing time. Participants can place bids at any time, provided the new bid is a minimum increment, ϵ, above the last bid. We choose units in which this minimum increment is 1[1]. At the closing time, the good or service for sale is sold to the highest bidder at the price they bid. We also assume that the bid increment of each auction is very small with respect to the value of the good or service for sale.

The reverse auctions are reverse English auctions with a fixed closing time. Participants can place offers at any time, provided the new offer is a minimum decrement, chosen to be 1 as above, below the last bid. At the closing time, the lowest offerer is given a contract to provide the required good or service at the price they offered.

To simplify the mathematical notation used later in this section, we model reverse auctions as 'negative' English auctions. A contract to provide a good g is represented as the sale of a negative good, $\overline{g}$. Participants place negative bids, which must increase (i.e. move towards zero) by at least the minimum bid increment. We

[1]We make this choice for ease of presentation. ϵ might be different in different auctions. Handling different values of ϵ would complicate the presentation, but not fundamentally change the algorithm

also model fixed-price sellers as auctions with a known and certain closing price.

The *technology*, T, of the agent consists of a set of rewrite rules, each with an associated cost. The rewrite rules are of the form $(\{a\} \Leftarrow \{b_1, ..., b_n\}, c)$, where $a, b_1, ..., b_n$ are goods. The rule is read as stating that the agent has the technology to generate good a from the bag of goods $\{b_1, ..., b_n\}$ at a cost of c. (We consider this as a bag, not a set, of goods as a composite service may use several identical components.) We also add to T all rules of the form $(\emptyset \Leftarrow \{g, \overline{g}\}, 0)$ for all goods g. These rules represent the fact that the agent can satisfy a contract $\overline{g}$ if they are able to generate a good g.

Given a bag of goods G, we say that G' is a one-step rewrite of G at cost c if:

$$(A \Leftarrow B, c) \in T,\ B \subseteq G,\ and\ G' = A \cup G \setminus B$$

We say that G^n is a rewrite of G at cost $c(G^n)$ if it is a one-step rewrite of G at cost c, or there is some rewrite G^{n-1} of G at cost $c(G^{n-1})$ such that G^n is a one-step rewrite of G^{n-1} at cost $c(G^n) - c(G^{n-1})$. The definition holds for $n \geq 1$, with $G^0 := G$.

Note that the rewrite relationship is not symmetric. Moreover, we make the assumption that applying successive rewrites to a bag of goods will never result in the original bag. That is $\nexists G$ s.t. $G^n = G, \forall n > 0$.

Let $W(G)$ be the set of all rewrites of G.

To each bag of goods G in the domain of interest we associate a number $v(G)$, the exogenous valuation to the agent of these goods. This defines the value the agent receives from making use of these goods (and contracts, represented by negative goods) outside the auction environment explicitly represented. The exogenous valuation of a good may be positive, if it is easy to sell on and has an expected market price. It may be negative, representing the dumping cost, or zero, if the good has no value but can be freely disposed of. The valuation of a supply contract, $\overline{g}$ will be negative, representing the cost of fulfilling the contract on the open market, or the de-commitment penalty associated with dropping the contract.

In the service composition environment, we expect the exogenous valuation of goods and contracts to be poor and most of the agent's profit to be made through sales in the reverse auctions. It is in the interests of the agent, therefore, to buy bundles in which the agent's technology can pair contracts won in reverse auctions with compositions of goods won in forward auctions.

The valuation $V(G)$ of a given bag of goods, G, is given by:

$$V(G) = \max_{G' \in W(G)} (v(G') - c(G'))$$

In other words, it is the valuation produced by optimum application of the agent's technology on G.

We define a *bid set*[2] to be a pair $(A, \mathbf{p})$, where $A \subset \mathcal{A}$ and $\mathbf{p} : A \to \mathbb{Z}$ is a price function, representing the price the agent is bidding in each auction in A.

The "utility" to the agent of winning a bid set $(A, \mathbf{p})$ is

$$u(A, \mathbf{p}) = V(A) - \sum_{a \in A} \mathbf{p}(a).$$

Our agent maintains a probabilistic model of the expected outcomes of each auction, based on past performance of similar auctions.

[2] Technically, it is a 'bid and offer set', but for ease of presentation we refer to it as the bid set and refer to all bids/offers as bids.

To each auction $a \in \mathcal{A}$ is associated a price distribution $P_a : \mathbb{Z} \to [0, 1]$ representing the belief that, with probability $P_a(p)$, auction a will close at price p. We set $F_a(p) = \sum_{p' \geq p} P_a(p')$: the agent's believed probability that auction a will close at or above price p. For subsets $A \subset \mathcal{A}$ we define $P_A(\mathbf{p})$ to be the believed probability that the auctions in A will close at the prices specified by a *price function* $\mathbf{p} : \mathcal{A} \to \mathbb{Z}$:

$$P_A(\mathbf{p}) = \prod_{a \in A} P_a(\mathbf{p}(a)), \quad (1)$$

and likewise F_A, the probability that that the auctions in A will close at or above the prices specified by $\mathbf{p}$:

$$F_A(\mathbf{p}) = \prod_{a \in A} F_a(\mathbf{p}(a)). \quad (2)$$

If the price in auction a is q, then the agent believes that the probability of a bid at price $p \geq q$ winning is

$$P_{win}(a, p, q) := \frac{P_a(p)}{F_a(q)}. \quad (3)$$

Similarly, for a collection of auctions A with current prices $\mathbf{q} : A \to \mathbb{R}$, the probability of the auctions closing at prices $\mathbf{p}$ is

$$P_{win}(A, \mathbf{p}, \mathbf{q}) = \frac{P_A(\mathbf{p})}{F_A(\mathbf{q})}. \quad (4)$$

We model a fixed-price seller, which will guarantee a sale of a given good at a price p, as an auction with 100% certainty of closing at p.

4.2 Specification of the algorithm

We now consider how the agent can use these beliefs to calculate information about expected future utility of deals it may win. Firstly, we define the notion of the expected utility $E(B, A, \mathbf{q})$ of a set of auctions B, given a set of observed prices $\mathbf{q}$, and given that the agent holds active bids in auctions A.

$$E(B, A, \mathbf{q}) = V(B) - C(B \cap A, \mathbf{q}) - C(B \setminus A, \mathbf{q} + 1) \quad (5)$$

where the function $C(S, \mathbf{q}')$ is the expected cost of winning the auctions S at prices greater than or equal to $\mathbf{q}'$:

$$C(S, \mathbf{q}') = \sum_{\mathbf{p}' \geq \mathbf{q}'} \sum_{a \in S} P_{win}(a, \mathbf{p}'(a), \mathbf{q}'(a)) \mathbf{p}'(a) \quad (6)$$

The expected utility of a set of auctions is thus the value of the bundle, minus the expected cost of winning each of the auctions. The latter is calculated by using the believed probability that the auction will finish at each given price, if our agent places a bid at that price. We restrict $p' > q$ for auctions $B \setminus A$, in (5) because we know that the agent does not hold bids in these auctions at prices $\mathbf{q}$, and so has no probability of winning at these prices.

The expression (5) gives us some idea of the intrinsic value of a bundle of goods B, but is not the expected return for placing a single bid in the auctions in B. In general such a bid does not *have* an expected return: we must reason over complete strategies.

Consider the expected value (given that prices are currently $\mathbf{q}$, and the agent holds the active bids A) of the following strategy, which we call "commitment to B": The agent chooses a set of auctions B, and for all future time steps, will always bid on any elements of B in which it does not hold active bids. If the agent sticks to this commitment, then we know its future choices, and so precise formulae for expected return can be calculated.

Let S be a possible set of auctions that the agent may win using this strategy: $B \subset S \subset A \cup B$. The probability that the auctions $S \setminus B$ will not be outbid, while the auctions $A \setminus S$ are, is

$$P_{ret}(S, A, \mathbf{q}) = \frac{F_{A \setminus S}(\mathbf{q}+1) P_{S \setminus B}(\mathbf{q})}{F_{A \setminus B}(\mathbf{q})}$$

Given this eventuality, the expected utility is evaluated in the same way as (5), except that instead of $V(B)$, the value we obtain is $V(S)$, and we occur additional costs for each auction in $S \setminus B$ that we win.

It follows that the expected value for following the commitment to B is:

$$E_c(B, A, \mathbf{q}) = E(B, A, \mathbf{q}) + \\ + \sum_{B \subset S \subset A \cup B} P_{ret}(S, A, \mathbf{q}) \Big(V(S) - V(B) - \sum_{a \in S \setminus B} \mathbf{q}(a) \Big) \quad (7)$$

The terms in this expression for which $S = B$ are the desired outcomes. The other terms correspond to obtaining some non-empty collection $S \setminus B$ of goods that do not contribute to our desired bundle B. Although they could still provide positive value, it is anticipated that in the service composition arena, where goods tend to complement one another, the slight increase of $V(S)$ with respect to $V(B)$ will not be large enough to compensate for the increase in costs $\sum_{a \in S \setminus B} \mathbf{q}(a)$, and each of these terms would have a negative impact on the expected value of the commitment. This will certainly be the case for negative goods appearing in $S \setminus B$, representing the possible winning of a reverse auction not appearing in our desired bundle (and therefore the need to satisfy it on the open market or to pay the de-commitment penalty).

The algorithm (`COMPOSER`) we propose is that at each time step the agent calculates the commitment B which has largest expected utility $E_c(B, A, \mathbf{q})$ given the currently held bids A and prices $\mathbf{q}$, and places the minimal bids required to take the lead in $B \setminus A$.

In practice this means it will initially identify the set of options which maximize its a-priori expected utility. These options will consist of a reverse auction for a given composite service, together with a set of English auctions for the required components. It will place bids in these forward/reverse auctions and will continue to compete in these auctions, placing more bids when outbid. However, if sufficient competing bids are placed to reduce the expected utility of this set of auctions, then it may change to another set of auctions which can generate the same composite service. It will do this if the expected gain from changing to this new bundle outweighs the expected cost of currently held bids which appear in the old bundle but not in the new bundle. If competing bids are placed in one of the reverse auctions it is participating in, and the expected value of that auction decreases sufficiently it may withdraw from that reverse auction. It may use the associated forward auctions in another option, or may withdraw from them as well.

There are two obvious problems with this algorithm:

- By its very nature, our algorithm does not in fact commit, since it re-evaluates its options at each opportunity. However, the value $E_c(B, A, \mathbf{q})$, which is truly the expected value of committing to bid on B, and hence is *not* the expected value according to the specified algorithm, is none-the-less (we claim) a good indication of the optimal choice to make. The estimate we use is conservative, in that the agent chooses a single bundle that will give the best overall expected utility. Choosing a different bundle for each possible outcome can only improve on this. We have adopted this approach initially, as we believe that it will provide good performance in the majority of situations. Experimentation and further analysis will be necessary to test this hypothesis.

- In practice, if the number of auctions is large, it will be difficult to evaluate equation (7) for every alternative bundle given realistic computational resource bounds. Ideally, if we had perfect information and unlimited computation time, we would calculate this accurately. However, if the algorithm is to be used in realistic circumstances, it is necessary to develop a simplified version which is more tractable. We now turn our attention to this problem.

4.3 Specification of a simplified algorithm

We can improve the efficiency of the algorithm by making two simplifying assumptions;

1. We do not consider every possible bundle of auctions, but restrict our attention to a promising subset of these bundles. We focus on only those bundles which can be transformed using the production rules into the empty set, (the *candidate* class).

2. When considering the expected utility of switching to a possible bundle, we ignore any utility gain of unplanned purchase of items in $S \setminus B$, beyond their immediate exogenous utility. (In other words, we ignore any utility gain from the unplanned purchase of goods and contracts which satisfy each other).

The first assumption is valid if we assume that the exogenous valuation of any good or set of goods will always be less than or equal to their purchase price in the auctions we represent.[3] In this situation, no utility can be gained by purchasing a good or set of goods for the purpose of disposing them outside the system represented. Because of this, any set of auctions will always contain a subset with equal or higher expected utility which can be transformed via the rewrite rules into the empty set.

Note that this assumption can be made without loss of generality. If a particular good b has an exogenous valuation $v(b)$ which can at times be greater than its purchase price in an auction, we can add an endogenous fixed-price buyer willing to purchase b for $v(b)$ (this is represented as an English auction for $\overline{b}$ with a 100% probability of closing at $-v(b)$). If a set of goods $\{b_1, ..., b_n\}$ is super-additive (i.e. $v(\{b_1, ..., b_n\}) > v(b_1) + ... + v(b_n)$ we represent this as a production rule $(\{q\} \Leftarrow \{b_1, ..., b_n\}, 0)$ together with a fixed-price buyer for q at $v(\{b_1, ..., b_n\})$.

The second assumption allows us to simplify equation (7) to become:

$$E_c(B, A, \mathbf{q}) = E(B, A, \mathbf{q}) + \\ + \sum_{a \in A \setminus B} P_{win}(a, \mathbf{q}(a), \mathbf{q}(a)).(v(a) - \mathbf{q}(a)) \quad (8)$$

We believe that this is, in almost all circumstances, a good approximation for equation (7). It is very unlikely that an agent will decide to withdraw from a reverse auction and set of forward auctions able to satisfy it, and then win all the relevant auctions because no-one else bids in them. Hence the positive contribution of

[3]This also applies to negative goods - the cost of satisfying a contract exogenously should be greater than or equal to the contract price in a reverse auction.

this serendipity to the expected utility will be small, so is safe to ignore. Note that because of this, equation (8) will yield a value of E_c slightly less than or equal to that given by equation (7).

These two assumptions greatly simplify the computation of the optimal bid set. We no longer need to calculate the endogenous valuation of all bundles, only those in the candidate class. Calculation of (8) for a given bundle only needs one endogenous valuation, as opposed to the $2^{|A\setminus B|}$ valuations required by (7).

Furthermore the candidate class of bundles, together with their endogenous valuations, can be generated prior to participation in the auctions (provided the set of auctions is known). It is sufficient to exhaustively backward-chain the rewrite rules from the empty set, while simultaneously keeping track of the incremental cost of transformation. This will generate all bundles which can be transformed into the empty set, together with their cost of transformation. The (negative) endogenous valuation of a given bundle will simply be the minimal cost of transformation to the empty set. We now present an algorithm to carry this out.

4.4 Algorithm for the computation of the candidate bundle set and bundles valuation

The algorithm presented here computes all the candidate bundles, while associating a valuation to the bundles. Let C be the set of the candidate bundles and W and X be the working sets to contain bundles that will end up in C after having generated other candidate bundles. O is the bag of all offerings that are available to the agent. Notice that the same basic component might appear more than once in O. T is the technology: the set of the production rules. Given a bag of goods, or *bundle* B, $V(B)$ is the valuation of the bundle. $J(G)$ is the set of the justifications of for G. A *justification* for a bundle G is a couple (G', r) such that where G is a rewrite of G' through r (see 4.1). Given a rule $r \in T$, $c(r)$ is the production cost associated to the rule.

$C = \emptyset$
$V(\emptyset) = 0$
$W = (\emptyset)$
For each offering $o \in O$
  For each bundle $G \in W$
    Let $G' = G \cup \{o, \bar{o}\}$
    Let $J(G') = \emptyset$
    Let $W = W \cup \{G\}$

W now contains all achievable bundles of the kind $\{o, \bar{o}\}$

Repeat
  Let G be a bundle in W
  For each element $a \in G$
    Let $R \subseteq T$ be the set of rules having a as *head*
    For each $r \in R$
      Let D be the *body* of r
      Let $G' = G \bigcup D \setminus \{a\}$
      If $G' \setminus O \neq \emptyset$ then break; G' is an invalid bundle
      Let $J(G') = J(G') \bigcup \{(G, r)\}$
      Let $W = W \bigcup G' \setminus G$
      Let $X = X \bigcup G$
Until $W = \emptyset$

X now contains all the candidate bundles with their justifications. This section of the algorithm is guaranteed to terminate because of the assumption that continuing to apply production rules to a bundle will never result in the original bundle, made in 4.1.

We now compute the valuations for the bundles from their justifications

Repeat
  Let $G \in X$ s.t. $\forall (G', r) \in J(G), G' not \in X$[4]
    If $J(G) = \emptyset$ then Let $V(G) = 0$
    Else
      $V(G) = \max_{(G',r)\in J(G)}(V(G') - c(r))$
      ; where c(r) is the cost associated to the rule r
      ; notice that $V(G')$ has been assigned, since $G' not \in X$
    Let $X = X \setminus \{G\}$
    Let $C = C \bigcup \{G\}$
Until $X = \emptyset$

At this point C is the *candidate* set and for each candidate bundle a valuation has been assigned.

4.5 Worked Example

To illustrate how this analysis operates, we return to the FreightMixer scenario described in §3. Based on past histories of similar auctions to the ones that were selected during the matchmaking phase, Freightmixer creates beliefs about the expected distribution of closing prices of these auctions. In addition, Freightmixer creates beliefs about the expected distribution of the two reverse auctions it is considering participating in. These are represented identically to the forward auctions, but as having negative prices. This represents the fact that Freightmixer will receive money if they win the auction.

We assume that the closing prices are uniformly distributed over the following sets:

$$\begin{array}{lcl} a & : & \{40, 45, \ldots, 135, 140\} \\ b & : & \{20, 25, \ldots, 95, 100\} \\ c & : & \{130, 135, 140, 145, 150\} \\ d & : & \{50, 55, \ldots, 105, 110\} \\ e & : & \{80, 85, \ldots, 115, 120\} \\ f & : & \{30, 35, \ldots, 65, 70\} \\ g & : & \{20, 25, \ldots, 135, 140\} \\ h & : & \{70, 75, 80, 85, 90\} \\ & & \\ \overline{la} & : & \{-250, -240, \ldots, -200, -190\} \\ \overline{sfo} & : & \{-220, -210, -200, -190, -180\} \end{array} \tag{9}$$

Before bidding begins, the agent holds no bids. We assume that the current price function $\mathbf{q}_0$ lies just below all of the above prices.

The agent has a set of production rules, representing the graph in figure 1. For the purposes of this example, we assume that the cost of composing is zero, and hence do not annotate the rules with costs:

$$\begin{array}{rcl} sfo & \Leftarrow & \{a, b\} \\ sfo & \Leftarrow & \{c\} \\ ord & \Leftarrow & \{e\} \\ ord & \Leftarrow & \{d\} \\ sfo & \Leftarrow & \{ord, f\} \\ la & \Leftarrow & \{ord, g\} \\ la & \Leftarrow & \{sfo, h\} \end{array} \tag{10}$$

Using the algorithm described above, we can apply equation (5) to evaluate the expected cost/benefit of participating in any set of forward and reverse auctions, which allows us to determine the set that has the optimal expected payoff. In this case the set is $\{d, g, c, \overline{sfo}, \overline{la}\}$. It has an expected purchase cost of $80 + 80 +$

[4]Again, G is guaranteed to exist if $X \neq \emptyset$ by the assumption in 4.1.

$140 - 200 - 220 = -120$. Hence, participating in these auctions and committing to purchase the complete bundle gives an expected payoff of 120. The agent therefore will place initial bids in these five auctions.

Let us now assume that the auctions have progressed, so that auction g now has a leading bid of 105 held by another party (giving an expected purchase price of 125) and we hold auction c with a bid of 130 (still giving an expected purchase price of 140). Let us explore the algorithm's behavior in two different cases.

Firstly, assume reverse auction $\overline{la}$ has a leading offer of 210 held by another party. This gives an expected closing price of 195 if we continue to participate in it. Hence, the expected profit of the bundle we are currently pursuing is now 50. Can we do better than this? Clearly we can, as the profit from the sub-bundle $\{c, \overline{sfo}\}$ is 60, and we only hold a bid in auction c, so there is no risk in withdrawing from the others. Applying equation (8) to the various alternatives shows that this is indeed the optimal bundle to commit to now: There is no way of profitably satisfying $\overline{la}$, hence we no longer pursue this aim. If we held no bids in any auction, equation (5) shows that we would choose to bid for the bundle $\{f, d, \overline{sfo}\}$, giving an expected profit of 70. However, equation (8) gives an expected profit of 44. The additional cost of de-committing from auction c and hence risking accidental purchase is not worth the potential gain of swapping to a different bundle.

Now if we assume that we hold the leading offer of 210 in $\overline{la}$, giving an expected closing price of 200 if we continue to participate in it. The expected profit of the current bundle is therefore 55, which is still less than the expected profit (according to (5)) of the sub-bundle $c, \overline{sfo}$. However, applying equation (8) tells us that we shouldn't switch to this bundle. The probability of us accidentally winning $\overline{la}$ is $1/3$, resulting in us receiving a payment of 210 but needing to purchase la from a fixed-price competitor at market value, that for the sake of the example we supposed to be 260. The expected cost of this risk (16.7) is significantly higher than the benefit of de-committing (5), so we should not do so.

However, applying the production rules and equation (8) to alternative options shows that there is a better alternative. Rather than using c to generate sfo, it can be used together with h to generate la at an expected cost of 200. f and d can then be used to generate sfo in a more cost-effective way. Hence, the optimal bundle to commit to now is $\{c, h, \overline{la}, f, d, \overline{sfo}\}$. This gives an expected profit of 70.

These examples demonstrate the different ways in which the algorithm can react to situations. It will often remain bidding for a certain bundle of goods and contracts. However, if this is no longer expected to be optimal according to equation (8), it will adopt an alternative approach. This may involve withdrawing from part of the bundle by de-committing from a reverse auction together with associated forward auctions. Alternatively, it may involve remaining committed to all reverse auctions but adjusting the set of forward auctions being used to purchase the required components. Finally, it may involve withdrawing completely from all auctions. At each stage, its decision is determined by its estimate of the expected utility of pursuing different options.

5. RELATED WORK

Research into automated negotiation has long been an important part of distributed AI and multi-agent systems. Initially it focused primarily on negotiation in collaborative problem solving, as a means towards improving coordination of multiple agents working together on a common task. [12] provide an overview of the pioneering work in this area. As electronic commerce became increasingly important, the work expanded to encompass situations with agents representing individuals or businesses with potentially conflicting interests. The Contract Net [25] provides an early architecture for the distribution of contracts and subcontracts to suppliers. It uses a form of distributed request-for-proposals. However, it does not discuss algorithms for determining what price to ask in a proposal. [11] use a more sophisticated negotiation protocol to allow the subcontracting of aspects of a business process to third parties. This is primarily treated as a one-to-one negotiation problem, and various heuristic algorithms for negotiation in this context are discussed in [5]. Other work in one-to-one negotiation includes the game-theoretic approach of [22] and the logic-based argumentation approach of [16].

Work has also focussed on effective algorithms for use in many-to-many negotiation environments such as a double auction. Gjerstad and Dickhaut [7] use a belief-based modelling approach to generate appropriate bids and offers. Their work is close in spirit to ours, in that it combines belief-based learning of individual agents' bidding strategies with utility analysis. However, it is applied to a single double auction marketplace, and does not allow agents to bid in a variety of auctions. Subsequent to this pioneering work, others have directly improved this approach [29] or worked on alternative algorithms in a similar environment. These include heuristic approaches ([4],[21]), stochastic analysis ([14],[15]), fuzzy logic [9] and dynamic programming [28]. Work has also been carried out to distribute [13] and generalise [20] the environment.

In addition to work on participation in a single many-to-many double auction, researchers have developed algorithms able to participate in many auctions simultaneously ([6], [17], [18], [10], [2], [1], [3]). These algorithms differ from the work presented here in that they focus on the purchase of one or more homogenous goods from multiple auctions, whereas we present an algorithm for the simultaneous purchase of component services and sale of one or more composite services in an auction environment. Schillo et. al [24] analyse task assignment in contract nets. They consider the problem of potential overcommitment by a supplier to several contractors caused by the delay between a supplier making an offer and the contractor selecting a supplier. This problem is related to that of overpurchasing in multiple auctions. The approach they take does not balance the utility of success against the risk of overcommitment, but instead tries to maintain the probability of overcommitment below a certain threshold.

Work has been carried out on the problem of simultaneous purchase of heterogeneous component services in an auction environment in response to a request for a composite service. Most notably, the Trading Agent Competition (TAC) ([30]) presents a problem where an agent must participate in several simultaneous auctions to purchase flights, accommodation and entertainment. Successful agents in this competition include ATTac ([26],[27]) and SouthamptonTAC [8]. The work presented in this paper differs from these in that it tackles a more generic version of the problem, where the utility of the buyers is not known. Unlike the TAC, both buyers and sellers must be simultaneously negotiated with through forward and reverse auctions. The algorithm we present is a generalization of that in [19], which focused only on forward auctions.

An alternative approach is to attempt to provide the right market mechanism in the first place, providing a centralized point of contact for all buyers and sellers to trade. Sandholm [23] proposes a sophisticated marketplace able to handle combinatorial bidding, and able to provide guidance to buyers and sellers as to which market mechanism to adopt for a particular negotiation. In the long term, as the different auction houses merge or fold and only a few remain, this approach will be ideal. In the short term, we expect improved market dynamics will occur through autonomous agents in multiple auctions.

6. CONCLUSIONS AND FUTURE WORK

In the future, service composition will play an essential role in e-commerce. Composite service products will be created on the fly in response to customer requests. In this paper, we have focused on the key problem of effective negotiation for service composition, and presented the specification of an algorithm to perform this task. The algorithm is able to participate simultaneously in request-for-quotes reverse auctions to win contracts for composite services, and in English auctions to purchase the components necessary. It is also able to make fixed-price purchases. To do this it chooses a set of forward and reverse auctions to participate in, and constantly re-evaluates this choice to determine if it is worth switching to a different set. It is able to trade off the risk of accidentally making purchases during this switching process against the benefit gained from a new set.

We have presented two forms of the algorithm - one which is more exact, the other which is more efficient. We hope to carry out experiments to determine if the simplifying assumptions in the latter algorithm still give good performance, and to explore alternative simplifying techniques to yield an algorithm which is both efficient and effective. We also plan to extend the algorithm to handle other forms of negotiation, in particular one-to-one negotiation, and to apply the techniques of [3] to handle various auction types with staggered opening times.

7. REFERENCES

[1] P. Anthony, W. Hall, V. Dang, and N.R. Jennings. Autonomous agents for participating in multiple on-line auctions. In *Proc. IJCAI Workshop on E-Business and the Intelligent Web*, pages 54–64, 2001.

[2] A. Byde. A dynamic programming model for algorithm design in simultaneous auctions. In *Proc.* 2^{nd} *Int. Workshop on Electronic Commerce (WELCOM-01)*, LNCS. Springer Verlag, 2001.

[3] A. Byde, C. Preist, and N. Jennings. Decision procedures for multiple simultaneous auctions. In *Proc.* 1^{st} *Int. Conf. Autonomous Agents and Multi-Agent Systems*, pages 613 – 620, 2002.

[4] D. Cliff and J. Bruten. Less than human: Simple adaptive trading agents for CDA markets. In *Proc. of the 1998 Symposium on Computation in Economics, Finance, and Engineering: Economic Systems*, 1998.

[5] P. Faratin, C. Sierra, and N. Jennings. Negotiation decision functions for autonomous agents. *Robotics and Autonomous Systems*, 3-4(24):159–182, 1998.

[6] P. Garcia, E. Giminez, L. Godo, and J. Rodriguez-Aguilar. Possibilistic-based design of bidding strategies in electronic auctions. In *Proc. 13th Biennial European Conference on Artificial Intelligence*, 1998.

[7] S. Gjerstad and J. Dickhaut. Price formation in double auctions. *Games and Economic Behaviour*, 22(1):1–29, 1998.

[8] M. He and N. Jennings. Southamptontac: Designing a successful trading agent. In *Proc* 15^{th} *European Conf. on AI (ECAI-2002)*, pages 8–12, 2002.

[9] M. He, H. Leung, and N. Jennings. A fuzzy logic based bidding strategy for autonomous agents in continuous double auctions. In *IEEE. trans.*, 2002.

[10] T. Ito, N. Fukuta, T. Shintani, and K. Sycara. BiddingBot: A multiagent support system for cooperative bidding in multiple auctions. In *Proc.* 4^{th} *Int. Conf. on Multi-Agent Systems*, pages 182–189, 2000.

[11] N. R. Jennings, P. Faratin, M. J. Johnson, P. O. O'Brien, and M. E. Wiegand. Using intelligent agents to manage business processes. In *First International Conference on the Practical Application of Intelligent Agents and Multi-Agent Technology (PAAM-96)*, pages 345–360, April 1996.

[12] B. Laasri, H. Laasri, S. Lander, and V. Lesser. Generic model for intelligent negotiating agents. *International Journal of Intelligent and Cooperative Information Systems*, 2(1):291–317, 1992.

[13] E. Ogsten and S. Vassiliadis. A peer-to-peer agent auction. In *Proc.* 1^{st} *Int. Conf. Autonomous Agents and Multi-Agent Systems*, pages 151 – 159, 2002.

[14] S. Park, E. Durfee, and W. Birmingham. Emergent properties of a market-based digital library with strategic agents. In *Proc.* 2^{nd} *Int. Conf. on Multi-Agent Systems*, 1998.

[15] S. Park, E. Durfee, and W. Birmingham. An adaptive agent bidding strategy based on stochastic modelling. In *Proc.* 3^{rd} *Int. Conf. on Autonomous Agents*, 1999.

[16] S. Parsons, C. Sierra, and N. R. Jennings. Agents that reason and negotiate by arguing. *Journal of Logic and Computation*, 8(3):261–292, 1998.

[17] C. Preist, C. Bartolini, and I. Philips. Algorithm design for agents which participate in multiple simultaneous auctions. In F. Dignum and U. Cortes, editors, *Agent Mediated Electronic Commerce III*, Lecture Notes in AI. Springer Verlag, September 2001.

[18] C. Preist, A. Byde, and C. Bartolini. Economic dynamics of agents in multiple auctions. In *Proc.* 5^{th} *Int. Conf. on Autonomous Agents*, 2001.

[19] C. Preist, A. Byde, C. Bartolini, and G. Piccinelli. Towards agent-based service composition through negotiation in multiple auctions. *AISBJ*, 1(1), 2001.

[20] C. Preist and C. Merida-Campos. Automated negotiation in many-to-many markets for imperfectly substitutable goods. In Julian Padget, David Parkes, Norman Sadeh, Onn Shehory, and William Walsh, editors, *Agent-Mediated Electronic Commerce IV*, volume 2531 of *Lecture Notes in Artificial Intelligence*. Springer Verlag, 2002. to appear.

[21] C. Preist and M. van Tol. Adaptive agents in a persistent shout double auction. In *Proc.* 1^{st} *Int. Conf. on the Internet, Computing and Economics*. ACM Press, 1998.

[22] J. Rosenschein and G. Zlotkin. *Rules of Encounter*. MIT Press, 1994.

[23] T. Sandholm. eMediator: A next generation electronic commerce server. In *Proc.* 4^{th} *Int. Conf. on Autonomous Agents*, 2000.

[24] M. Schillo, C. Kray, and K. Fischer. The eager bidder problem: A fundamental problem of dai and selected solutions. In *Proc.* 1^{st} *Int. Conf. Autonomous Agents and Multi-Agent Systems*, pages 599 – 606, 2002.

[25] R. G. Smith. The Contract Net protocol: High level communication and control in a distributed problem solver. *IEEE Transactions on Computing*, 29(12):1104–1113, 1980.

[26] P. Stone, M. Littman, S. Singh, and M. Kearns. ATTac-2000:

An adaptive autonomous bidding agent. In Jörg P. Müller, Elisabeth Andre, Sandip Sen, and Claude Frasson, editors, *Proceedings of the Fifth International Conference on Autonomous Agents*, pages 238–245, Montreal, Canada, 2001. ACM Press.

[27] P. Stone, R. E. Schapire, and J. A. Csirik. ATTac-2001: A learning, autonomous bidding agent. In Julian Padget, David Parkes, Norman Sadeh, Onn Shehory, and William Walsh, editors, *Agent-Mediated Electronic Commerce IV*, volume 2531 of *Lecture Notes in Artificial Intelligence*. Springer Verlag, 2002. to appear.

[28] G. Tesauro and J. L. Bredin. Strategic sequential bidding in auctions using dynamic programming. In *Proc. AAMAS02*, pages 591 – 598, 2002.

[29] G. Tesauro and R. Das. High-performance bidding agents for the continuous double auction. In *Proceedings of IJCAI workshop on Economic Agents, Models and Mechanisms*, 2001.

[30] M. P. Wellman, A. Greenwald, P. Stone, and Peter R. Wurman. The 2002 trading agent competition. In *Proceedings of the* 14^{th} *Innovative Applications of Artificial Intelligence Conference*, 2002.