Agent Negotiation Strategies for Composing Service Workflows

John McDowall[#1], Larry Kerschberg[#2]

[#]*Department of Computer Science, George Mason University*
4400 University Drive, Fairfax, VA 22030 USA
[1]jmcdowal@gmu.edu
[2]kersch@gmu.edu

Abstract—**Composing services into executable workflows presents many challenges. One of the most difficult challenges is deciding which option among several available workflows is the best value for the user. Cost, reliability, user preferences, and other factors must be balanced to develop the best recommendation. In this paper we describe a strategy for developing optimal recommendations in a system where each service is represented by an intelligent agent that employs a bidding strategy that balances its own interest and its relationship to other agents in the proposed workflow. This offers opportunities for agents to not only compete with each other, but also to form informal ad-hoc coalitions that cooperate to win more work.**

I. Introduction

Web services were promoted as the next generation of software reuse—once written, they could be invoked from anywhere and rapidly recomposed into new workflows as business needs changed. The hope was that users would be able to reconfigure their applications as needed without relying on dedicated software developers; ideally applications would automatically adapt to new services as they became available.

Reality has been very different. There are many difficulties inherent in recomposing heterogeneous services at runtime; an overview of these difficulties and one approach to addressing them can be found in [1]. Such a multi-agent approach provides a way to compose services into an executable workflow, but it does not address how to select among the resulting workflows to help the user select the option that best meets the user's needs.

In this paper, we propose a method for optimizing workflow recommendation to the user based on a combination of user preferences, the characteristics of individual service offerings, and incentives for using particular combinations of services in a given workflow. Our design uses intelligent agents as representatives of each service as well as the user, allowing the agents to negotiate among themselves to develop an optimal recommendation based on the circumstances of the current workflow, each agent's experience, and the preferences of each agent participating in the negotiation. Our design also allows for each service agent to learn from past experiences and adjust its negotiating strategy accordingly.

This paper is organized as follows: In Section II, we provide some background information on building executable workflows from services and on intelligent agent negotiations. In Section III, we describe a simple example we use to illustrate the principles at work. Section IV describes the process for developing a recommendation. In Section V we discuss strategies for improving that recommendation using a variety of negotiation techniques. We describe related work in Section VI and conclude the paper in Section VII.

II. Background

Our research is focused on creating executable workflows from available services. We begin with a process model specified in a language such as the Business Process Model and Notation (BPMN) [2].

A. Workflow Composition

To facilitate further discussion we provide the following definitions.

1) Parameter Equivalence: We say two parameters p_1 and p_2 are equivalent if the semantics of p_1 and p_2 are the same (e.g., both refer to a date) and if the format of p_1 and p_2 are the same (e.g., both express the date numerically in the format year-month-day).

2) Service Composition: We say two services A and B can be composed if, for each input parameter of B there is an equivalent output parameter of A.

3) *Workflow:* We define a workflow as a series of services that can be composed together to fulfill the goal of the original business process model.

For each activity within the process model, we search for services that perform the task described by that activity. Upon matching services to each activity, we analyze the available services for each activity to determine which can be composed with services in the succeeding activity. When this process is completed, we have discovered all possible workflows that can be composed from the available services to fulfill the goals of the original process. It is possible for any given service to be part of several potential workflows.

B. Intelligent Agents

Intelligent agents are software components that are empowered with a certain degree of autonomy to complete assigned tasks within certain bounds. They offer many advantages in diverse, highly dynamic systems. Chief among these advantages is their ability to negotiate among themselves in accordance with defined rules to optimize their behavior to rapidly changing circumstances. Approaches such as the Belief-Desire-Intention (BDI) model [3] define ways to structure agent behavior patterns in a way that allows agents to act in accordance with goals defined on a per-agent basis.

Additionally, the Foundation for Intelligent Physical Agents (FIPA) defines protocols for inter-agent communication that allow agents to compete, cooperate, and otherwise interact in a manner analogous to their human counterparts. Agent platforms such as the Java Agent Development Environment (JADE) [4] provide easy-to-use frameworks for creating intelligent agents in accordance with these specifications.

Several projects over the years have made use of agent-based systems to enable workflow composition, [5], [6] being only two examples among many. Our design moves a step beyond these, enabling the agents to negotiate the terms of under which they will execute any given workflow invocation. We make use of the FIPA Contract Net Protocol [7] and Iterated Contract Net Protocol [8], combined with an affinity matrix that describes each agent's preferences, to govern the negotiation process among the agents in order to arrive at an understanding that is acceptable to both the user and the agents executing the workflow process.

III. Motivating Example

We use a simple but illustrative example to demonstrate the process of composing a workflow and negotiating the terms of service. Consider a simple flight reservation process as depicted in Fig. 1. This basic model encapsulates the process of searching for available flights between two cities, selecting a flight, completing the reservation, and processing the payment. Each of these steps can be performed by discrete services. For example, Google offers a service that finds all flights between selected cities on a chosen day. Likewise, individual airlines offer services for finding flights they offer between cities they serve, as well as services for reserving the desired flights. Services like Expedia and Travelocity offer comparable services. While each vendor would prefer that a user complete the entire process using that vendor's suite of services, it is possible to combine services from different vendors to complete the transaction. Indeed, combining services from different vendors may offer the user better value, more convenience, or other benefits.

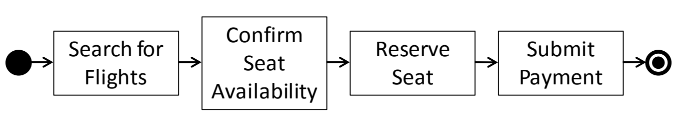

Fig. 1 Simplified flight reservation model

The travel domain is a popular one for research into recommender systems. Whereas travel-related recommender system such as [9] focus on recommending specific products directly to the traveler, our research concentrates on a more general recommendation scenario; the travel domain happens to offer a familiar scenario that most readers can relate to.

In order to focus on the process of developing a recommendation, this example ignores the case where a single service can perform multiple steps in the workflow model. This example is also confined to a single-threaded process where the outputs of a single service are all that are needed to invoke the next service; cases where the outputs of multiple services are needed to form the inputs of the next service are certainly a consideration in real-world applications, but we feel using such an example would muddy the discussion of the main points in this paper. We also limit our recommendation criteria to a single aspect of the service offering. The principles apply equally to any aspect or combination of Quality of Service (QoS) attributes that the services and user may find important. For instance, responsiveness might improve with increasing service cost. Our example uses the cost of the service for a single invocation because this will be readily understood and because it enables a simple comparison.

Table I shows a fictional but representative sample of service offerings that can be composed in multiple possible ways to complete a flight reservation. Each entry in the table shows the vendor and the task the service performs. We omit specific inputs and outputs for the sake of clarity; for our purposes we will assume that all service offerings in this example can be composed with services from other vendors.

TABLE I
Sample Service Offerings

Vendor	Service	Cost
ABC Travel	Search for Flights	$1
	Confirm Seat Availability	$2
	Reserve Seat	$3
	Submit Payment	$1
Trip Planner	Flight Search	Free
	Confirm Seat Availability	$1
Search Giant	Flight Search	Free
Tiger Airways	Search for Flights	$1
	Confirm Seat Availability	$1
	Reserve Seat	$2
	Submit Payment	$2
Payment Buddy	Submit Payment	$1

There are many possible compositions of these services. A detailed discussion of how services are discovered, how we build the list of potential combinations and winnow that list down for presentation to the user is beyond the scope of this paper. We have developed an initial implementation using a limited set of services and are researching mechanisms to improve service discovery and matchmaking. It is enough to say that for each composition that is deemed viable, a broker agent is assigned to represent that combination of services for the remainder of the recommendation process. For the sake of simplicity, we will look at only three compositions here. Table II shows three sample service compositions and the total cost to use each composition (this does not include the actual cost of the tickets, only the cost to use the services).

TABLE II
SERVICE COMPOSITIONS

Label	Composition		
	Vendor	**Service**	**Cost**
A	Tiger Airways	Search for Flights	1
	Tiger Airways	Confirm Seat Availability	1
	Tiger Airways	Reserve Seat	2
	Tiger Airways	Submit Payment	2
	Total Cost for A		**6**
B	Search Giant	Flight Search	0
	Trip Planner	Confirm Seat Availability	1
	Tiger Airways	Reserve Seat	2
	Payment Buddy	Submit Payment	1
	Total Cost for B		**4**
C	ABC Travel	Search for Flights	1
	ABC Travel	Conform Seat Availability	2
	ABC Travel	Reserve Seat	3
	Payment Buddy	Submit Payment	1
	Total Cost for C		**7**

Given these options, our task is to recommend the workflow that meets the user's requirement (to complete a flight reservation) at the lowest cost while also meeting the goals of the constituent services.

IV. DEVELOPING A RECOMMENDATION

Many factors play into developing a recommendation, including user preferences, QoS, and the cost of a given service. We focus here on cost due to the major role it plays in business decisions and the extent to which cost models influence the behavior of both buyers and sellers. Monetary cost is a readily measurable proxy for many aspects of a product; this is particularly when the product is a service. A reputation for quality, accuracy, and timeliness will increase demand for a service and allow the seller to increase the price. This idea is captured in the old adage "you get what you pay for." There are certainly exceptions, and there are other ways to measure desirability of an offering, but in this paper we focus on cost because it is readily understood and is often the primary driver in selecting a service provider.

In a competitive marketplace where many vendors offer competing versions of similar services, developing the right combination of service offerings and price points is a complex task. Internal factors play a significant part in these decisions. While one business may want to maximize revenue, another may pursue a strategy focused on increasing market share even if that means operating at a short-term loss. A service vendor may want to penetrate new markets and therefore offer package deals that significantly discount the cost of one service when used in conjunction with another service. Other factors also play into pricing decisions; sometimes a "pain in the neck" factor is applied to bids when a vendor finds itself working with customers or other vendors that it finds difficult to work with.

All these factors drive how offerings are priced. In fact, cost plays such an important part in service offerings that companies closely guard their pricing information; the cost models used to generate bids are some of the most closely guarded proprietary information a company has. If competitors can predict a firm's bids that firm is at a significant disadvantage and risks being under-bid on a regular basis. By the same token, a firms is also at a disadvantage when its prices don't react quickly to rapidly changing market dynamics.

At the same time service vendors are trying to develop an ideal pricing strategy, consumers are trying to get their best value they can, weighing many factors to arrive at a decision. Still, in many cases the major driver of their decision will be cost.

Using the limited example described in the preceding section, we describe two methods for developing a recommendation regarding which service composition the user should invoke. We first discuss a simple recommendation technique and then describe our improved method.

A. Simple Recommendation

Given the information provided, comparing the available service compositions is a straightforward matter. As shown in Table II, composition A is built from services offered by a single vendor, and the total cost of using composition A to find and reserve a flight is $6.

Composition B assembles services from four different vendors to create a workflow that minimizes the advertised cost of the services while still accomplishing the goal. The cost to reserve a flight using composition B is $4.

Composition C combines services from two vendors with a total cost of services of $7.

Each of these compositions will meet the user's underlying need to find and reserve a flight. Knowing that, we can compare the service offerings to develop a recommendation for which option the user should choose based on his or her preferences. If the user's primary concern is to minimize cost, then composition B is the obvious choice at $4.

Alternatively, the user may have a strong preference for reserving seats directly with the airline, and that preference may override the cost consideration. For example, Tiger Airways may have a loyalty program that offers reward points for using the airline's reservation system exclusively. In that case, composition A is the preferred alternative despite its higher cost.

Preferences can also be negative. Perhaps the user had a very bad experience using the Tiger Airways reservation system and they want to avoid using any part of it if at all possible. In that case, composition C is the best choice, as it is the only one that is completely Tiger-free.

B. Improving Recommendations

A major strength of agent-based systems is the ability of the agents to compete and cooperate amongst themselves to best meet competing priorities. In a simple agent negotiation scenario, each agent is independent and is competing and negotiating with other independent agents, all of them working to achieve their own goals.

In our case, where agents represent individual services that are composed into an executable workflow, it is not sufficient for an agent to work to achieve its own goals. It must alternately compete and cooperate with other agents that participate in the agent's own workflow as well as in competing workflows. In each workflow composition, there are three stakeholders whose interests must be considered:

- the user: a person who is trying to achieve some business goal,
- the service agent: a software component that acts as the point of contact and interface for a particular service,
- the vendor: an organization that sells the use of one or more services.

The user and service agent are direct participants in the composition (the user may be represented by a user agent, but their interests can be considered identical). The vendor is indirectly represented by the service agent. When bidding for a workflow, every given agent has to balance several competing priorities:

- a service agent's desire to maximize use of the service it represents,
- a vendor's desire to see the maximum use of its services across all the vendor's offerings,
- a vendor's desire to gain or maintain an advantage over competitors,
- multiple vendors' desire to cooperate with each other (as in a partnership),
- the user's desire for a workflow that best meets his own priorities.

In our design, the user is represented by a User Agent that keeps track of the available workflows and the best recommendation available so far. Each workflow is represented by a Broker Agent that monitors the status of that workflow's current offer and coordinates communication among the Service Agents to help them optimize their offer for that particular workflow. Each service is represented by a Service Agent that negotiates for that service on behalf of the vendor. It is important to note that any given Service Agent may be a part of multiple workflows, and may make different offers to the Broker Agent based on the circumstances of that workflow.

The User Agent communicates with the Broker Agents using the FIPA Iterated Contract Net Protocol [8]; each Broker Agent uses the same protocol to communicate with each of the Service Agents in its workflow.

1) Initial Bids: Each Service Agent has a base price as shown in Table II. This is the price the Service Agent offers before negotiations begin; it also has some maximum discount it is willing to offer under the right circumstances. While Service Agents may advertise their base price, we expect each Service Agent would conceal its maximum discount for competitive reasons. These bids are presented to the user for the same reason cars have sticker prices: sometimes people are willing to pay the amount shown. This may be because cost is not a concern, or it may be that speed is a higher priority, or it may be due to myriad other factors.

To govern its negotiating strategy, each Service Agent maintains an Affinity Matrix that captures that agent's preferences and the extent of a discount it can offer the Broker Agent in light of the particular circumstances of any given workflow. The Affinity Matrix can be represented by a table. Continuing the flight reservation example above, an Affinity Matrix for an agent representing a service from Payment Buddy is shown in Table III

TABLE III
PAYMENT BUDDY AFFINITY MATRIX

Vendor	Affinity
ABC Travel	+0.9
Trip Planner	+0.2
Search Giant	+0
Tiger Airways	-0.7
Payment Buddy	+1.0

Each vendor the Service Agent is aware of is represented in the Affinity Matrix and has a corresponding Affinity value between +1.0 and -1.0. An Affinity of +1.0 indicates that the Service Agent will offer the maximum possible discount to the Broker Agent for each service provided by that vendor in the current workflow. Negative affinities represent a degree of hostility to that vendor; this enables a Service Agent to offer a "negative discount"—a penalty—for using services from a competing vendor in the same workflow. Using the example in Table III, the Service Agent representing a Payment Buddy service would offer 90% of the maximum discount for each service in the workflow offered by ABC Travel, but it would charge a penalty equal to 70% of the maximum discount for each service in the workflow offered by Tiger Airways.

When the Broker Agent receives initial bids from each Service Agent, the Broker Agent builds a list of vendors and how many services each vendor has in this particular workflow. Each Broker Agent passes its bid to the User Agent, which selects the best bid and then notifies each Broker Agent of the current winning bid and requests a revised bid. Each Broker Agent then passes the vendor information back to its participating Service Agents and requests revised bids.

2) *Subsequent Bids:* Each Service Agent receives a request for a revised bid from the Broker Agent together with a list of other vendors that are participating in that Broker Agent's workflow. The list of vendor information includes the number of services being provided by each vendor participating in the workflow. With this information, the Service Agent calculates its next bid based on the following formula:

$$n = total\ number\ of\ services\ in\ a\ given\ path$$
$$n_j = number\ of\ services\ in\ workflow\ from\ vendor\ j$$
$$d_j = discount\ for\ using\ services\ from\ vendor\ j$$
$$d = total\ discount\ offered$$

$$d = \sum d_j \left(\frac{n_j}{n}\right) for\ each\ vendor\ j$$

Assume the Payment Buddy service is willing to offer a maximum 50% discount on its service. Using the Affinity

Matrix in Table III and the service compositions in Table II, when we apply the above formula we get revised costs as shown below.

Composition A uses no Payment Buddy services, so there is no change to its offering and no need to calculate a revised bid.

For Composition B uses a mix of services and the Payment Buddy Service Agent calculates a discount of 14.25% for its next bid as follows:

$$d = 50 * 0\left(\frac{1}{4}\right) + 50 * 0.2\left(\frac{1}{4}\right) + 50(-0.7)\left(\frac{1}{4}\right) + 50 * 1\left(\frac{1}{4}\right)$$

The discount the Payment Buddy agent offers for Composition C is 33.75%, calculated in the same manner and shown by:

$$d = 50 * 0.9\left(\frac{3}{4}\right) + 50 * 1\left(\frac{1}{4}\right)$$

We can calculate the second round of bids using the additional Affinity Matrices in Table IV, Table V, and Table VI below (the Search Giant service is free, and thus requires no Affinity Matrix). For simplicity, we will assume each Service Agent's maximum discount is 50% and we will dispense with showing the calculations for the revised bids from each service.

TABLE IV
TIGER AIRWAYS AFFINITY MATRIX

Vendor	Affinity
ABC Travel	-0.2
Trip Planner	+0.3
Search Giant	+0
Tiger Airways	+1.0
Payment Buddy	-0.3

TABLE V
TRIP PLANNER AFFINITY MATRIX

Vendor	Affinity
ABC Travel	-0.9
Trip Planner	+0
Search Giant	+0
Tiger Airways	+0.3
Payment Buddy	+0

TABLE VI
ABC TRAVEL AFFINITY MATRIX

Vendor	Affinity
ABC Travel	+1.0
Trip Planner	-0.5
Search Giant	+0
Tiger Airways	-0.3
Payment Buddy	+0.9

Using these Affinity Matrices, we can calculate the second bid and compare it to the original bid to see the effect of the various discounts on the options offered to the user in Table VII.

TABLE VII
UPDATED BIDS

Label	Bids			
	Vendor	**Service**	**1st**	**2nd**
A	Tiger Airways	Search for Flights	1	0.50
	Tiger Airways	Confirm Seat Availability	1	0.50
	Tiger Airways	Reserve Seat	2	1
	Tiger Airways	Submit Payment	2	1
	Total Cost for A		**6**	**3**
B	Search Giant	Flight Search	0	0
	Trip Planner	Confirm Seat Availability	1	0.96
	Tiger Airways	Reserve Seat	2	1.75
	Payment Buddy	Submit Payment	1	0.86
	Total Cost for B		**4**	**3.57**
C	ABC Travel	Search for Flights	1	0.62
	ABC Travel	Conform Seat Availability	2	1.24
	ABC Travel	Reserve Seat	3	1.85
	Payment Buddy	Submit Payment	1	0.67
	Total Cost for C		**7**	**4.38**

As Table VII shows, there have been significant changes to the bids offered by each. In this case no bids actually increased, although that is certainly a possibility under the proper conditions. As the second round of bids shows, Composition A, which had been the second-most expensive option has become the least expensive option. And while all the bids have decreased, Compositions A and C show the most significant decreases due to the significant discounts offered by some services as a consequence of being composed with services from preferred vendors.

The example shown here is simple for purposes of illustration. Each Service Agent would likely have different discount levels for different vendor options instead of the steady 50% used here. Additionally, the User Agent will factor in some preferences of its own, perhaps weighting individual selections based on a preference for a particular vendor or a preference for workflows composed from services offered by a single vendor.

V. IMPROVING THE RECOMMENDATION

The basic protocol described above provides a mechanism for intelligent agents to compete and attempt to improve their offers for the purpose of increasing their usage. We have identified several potential improvements to the recommendation process that may improve the bidding process. Given that each Service Agent is acting independently and potentially interacting with multiple Broker Agents during each bidding cycle, we believe the complexity of the interactions resulting from one or more of these enhancements may yield emergent behaviors that are not readily apparent from an analysis of the basic algorithm. Toward that end, we are in the process of designing experiments to explore this behavior.

A. Time-Varying Discounts

Our basic algorithm provides for two rounds of bidding: a first offer and then a "best and final" offer. As we described above, this is a simple but effective mechanism for developing a recommendation that balances the desires of the user to get the best deal and the desires of the service vendors to attract the most business combined with a desire to reinforce preferred business relationships.

One potential improvement is to provide each Service Agent with a variable discount scale, where after the initial bid, each Service Agent alters its bid by a small amount, and that amount grows as additional bids are requested from the Broker Agent, up to some maximum discount the Service Agent is willing to offer.

One interesting effect this may bring out concerns those bids that rise over time because some Service Agents will be increasing their bids when they are composed in a workflow where the other services are offered by vendors they do not want to cooperate with. This may result in bids that diverge very quickly, and it might also result in cases where the best offer available was available on the first bid.

B. Consortium Building

Our initial design uses Service agents that are seeded with their preferences at design time; the Service Agents apply the same discounts repeatedly over time. A potential improvement to their behavior is to have them adapt their discount levels as they gain experience.

One possible strategy is for each Service Agent to keep track of how often it is paired with Service Agents from each vendor of interest, and to adjust its discount levels accordingly. As multiple Service Agents independently adjust their discount levels based on their experiences with Service Agents from other vendors, we may see informal "consortia" develop, where groups of vendors offer increasing and mutually reinforcing discounts that drive increasing amounts of use for those workflows where these cooperating Service Agents are grouped together.

Another strategy might be for a Service Agent to increase the discounts offered when it is grouped with Service Agents from vendors that it is not regularly grouped with. This may lead to cases where business relationships are forged over time as the agents gain more experience.

Using either strategy, the discounts can be adjusted either through altering the specific discount amount, or by altering the affinity levels each Service Agent maintains.

C. Failure-Based Discounts

This modification is similar to the Consortium Building modification in that it adapts discount levels based on experience. However, this strategy is designed to increase a Service Agent's chances of being selected for use.

Each Service Agent has some optimal level of use, a level where it is maximizing its use but can still support all incoming requests without adversely affecting performance. Any Service Agent attempting to respond to too many service requests will be forced to delay some responses, decreasing its availability (and potentially making it less desirable to users and other service vendors).

Under this modification, as a Service Agent bids on each workflow it keeps track of how often it is part of the winning bid and how often it is not. As long as the Service Agent still has some excess capacity it increases its rate of discount when bidding in an effort to drive increased traffic. The discount level can be stabilized once the optimal level of use has been reached. By the same token, the Service Agent can decrease its discount level if it senses that it is exceeding its optimal usage level, using the varying discount as a way to moderate its traffic level.

D. Cutthroat Bidding

Unlike the modifications described above, this one focuses on the technique used by the User Agent to solicit bids from Broker Agents.

After receiving the first bid, instead of requesting improved bids from all Broker Agents based on the best bid received in the first round, the User Agent eliminates the worst (highest) bid from the pool of bidders and solicits improved bids from the remaining Broker Agents.

By itself, this modification would have the advantage of shortening the bidding cycle in cases where there are many competing Broker Agents. When combined with the Failure-Based Discounts modification or the Consortium Building modification, this option might drive more extreme adaptations among the Service Agents than would otherwise be the case.

VI. Related Work

For decades, researchers have been exploring how groups of individuals, each one acting independently, can behave as if they are working toward a common goal. In describing markets, Adam Smith famously referred to this phenomenon as "the invisible hand" [10]. Performing controlled studies on large groups of individuals interacting in a market-like environment was nearly impossible until the development of multi-agent systems. As these systems have advanced over the past few decades, researchers have taken advantage of them to experiment with a wide variety of models to describe the interaction of individuals in a market environment.

An early example is the ATTac-2000 system [11] which used an adaptive strategy to improve agents' competitiveness against each other in bidding to fulfill a travel order. While ATTac-2000 adapted to the results of past bidding competitions, its model was based on competitive agents; agents were never placed in a position where helping each other was advantageous.

The research described in [12] includes aspects of dynamic pricing that bears some resemblance to the work described here, but that work is focused on the allocation of computing resources among several competing consumers, but it does not include cooperative aspects where different agents may be working toward a shared goal.

Like many research projects in this area [13] concentrates on bidding strategies for auctions, where each agent is trying to outbid the others while minimizing its own costs. Again, this scenario does not lend itself to agent cooperation because an honest auction is an inherently competitive environment.

The use of cooperation among agents in service-oriented architectures is discussed in [14], but not in the context of assembling services into executable workflows. Despite this, their work develops concepts that we expect to be helpful as our research progresses.

VII. Conclusions

The ideas proposed in this paper are not intended to constitute a comprehensive bidding strategy for environments that combine cooperation and competition in multi-agent systems. Our intent is to improve agent negotiation in the context of composing services into executable workflows, not to describe a comprehensive negotiation strategy. Even so, we believe this work will be instructive to others researching negotiation strategies in multi-agent systems, particularly in circumstances where agents alternatively cooperate and compete with each other based on the circumstances of each situation.

Going forward, we plan to expand upon these ideas as we progress beyond single-path workflows into workflows that include concurrent parallel paths and alternative branches. One of the considerations of these more complex cases is how a Service Agent should adjust its bids in cases where a single Service Agent appears in multiple segments of the same workflow composition (as in the case where the same service is part of different concurrent parallel paths in a workflow). Another intriguing consideration we will explore is how to calculate bids when there are alternate paths within a workflow, and a Broker Agent must consider the probability any given path will be executed when factoring in the bids of Service Agents on that path.

References

[1] J. McDowall and L. Kerschberg, “A multi-agent approach for generating ontologies and composing services into executable workflows,” in *Proceedings of the 2010 EDBT/ICDT Workshops*, Lausanne, Switzerland, 2010, pp. 1-12.

[2] “Business Process Model and Notation (BPMN) Version 2.0.” Object Management Group, Jan-2011.

[3] M. Georgeff, B. Pell, M. Pollack, M. Tambe, and M. Wooldridge, “The Belief-Desire-Intention Model of Agency,” in *Intelligent Agents V: Agents Theories, Architectures, and Languages*, vol. 1555, J. P. Müller, A. S. Rao, and M. P. Singh, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 1-10.

[4] F. Bellifemine, A. Poggi, and G. Rimassa, “JADE: a FIPA2000 compliant agent development environment,” in *Proceedings of the fifth international conference on Autonomous agents*, Montreal, Quebec, Canada, 2001, pp. 216-217.

[5] L. Kerschberg, H. Jeong, Y. U. Song, and W. Kim, “A Case-Based Framework for Collaborative Semantic Search in Knowledge Sifter,” in *Proceedings of the 7th international conference on Case-Based Reasoning: Case-Based Reasoning Research and Development*, Belfast, Northern Ireland, UK, 2007, pp. 16-30.

[6] J. Yoon and L. Kerschberg, “A Functional Approach to XML-Based Dynamic Negotiation in E-Business,” in *The Functional Approach to Data Management, P.M.D. Gray, et al., Editors*, Heidelberg: Springer, 2003, pp. 393-416.

[7] “FIPA Contract Net Interaction Protocol Specification.” Foundation for Intelligent Physical Agents, 06-Dec-2002.

[8] “FIPA Iterated Contract Net Interaction Protocol Specification.” Foundation for Intelligent Physical Agents, 06-Dec-2002.

[9] K. Alodhaibi, “Decision-Guided Recommenders With Composite Alternatives,” George Mason University, Fairfax, VA, 2011.

[10] A. Smith, *An Inquiry into the Nature and Causes of the Wealth of Nations*. London: Methuen & Co., Ltd., 1776.

[11] M. Kearns, M. L. Littman, S. Singh, and P. Stone, “ATTac-2000: An Adaptive Autonomous Bidding Agent,” *Journal of Artificial Intelligence Research*, vol. 15, Dec. 2001.

[12] B. Pourebrahimi, K. Bertels, S. Vassiliadis, and L. O. Alima, “A Dynamic Pricing and Bidding Strategy for Autonomous Agents in Grids,” in *Agents and Peer-to-Peer Computing*, vol. 5319, S. R. H. Joseph, Z. Despotovic, G. Moro, and S. Bergamaschi, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010, pp. 55-71.

[13] J. Ma and M. L. Goyal, “Designing a Successful Bidding Strategy Using Fuzzy Sets and Agent Attitudes,” in *Web-based Support Systems*, J. Yao, Ed. London: Springer London, 2010, pp. 359-370.

[14] X. Wang, T. N. Wong, and G. Wang, “Service-oriented architecture for ontologies supporting multi-agent system negotiations in virtual enterprise,” *Journal of Intelligent Manufacturing*, Nov. 2010.