TAGA: Travel Market Framework in Agentcities (Demonstration)

Youyong Zou, Tim Finin, Li Ding, Harry Chen, Rong Pan
Computer Science and Electrical Engineering
University of Maryland, Baltimore County

Travel Agent Game in Agentcities (TAGA) is a framework that extends and enhances the Trading Agent Competition (TAC) system to work in Agentcities, an open multiagent systems environment of FIPA compliant systems. TAGA makes several contributions: auction services are added to enrich the Agentcities environment, the use of the semantic web languages RDF and DAML+OIL improve the interoperability among agents, and the DAML-S ontology is employed to support service registration, discovery and invocation. The FIPA and Agentcities standards for agent communication, infrastructure and services provide an important foundation in building this distributed and open market framework. TAGA is intended as a platform for research in multi-agent systems, the semantic web and/or automated trading in dynamic markets as well as a self-contained application for teaching and experimentation with these technologies. It is running as a continuous open game at http://taga.umbc.edu/ and source code is available for research and teaching purposes.

1 Introduction

The Trading Agent Competition (TAC) [7] is a test bed for intelligent software agents that interact through simultaneous auctions to obtain services for customers. TAC trading agents operate within a travel market scenario, buying and selling goods to best serve their given travel clients. TAC is designed to promote and encourage research in markets involving autonomous trading agents and has proven to be successful after three consecutive year's competition.

Although TAC's framework, infrastructure and game rules have evolved over the past three competitions, the assumptions and approach of TAC limit its usefulness as a realistic test bed for agent based automated commerce. TAC has used centralized auctions as the sole mechanism for service discovery, communication, coordination, commitment, and control among the participating software agents. The agents communicate with the central market servers through simple socket interfaces, exchanging pre-defined message. The abstractness and simplicity of the TAC approach helped to launch it as a research vehicle for studying bidding strategies, but are now perceived as a limiting factor for exploring the wide range of issues inherent in automated trading in open environments.

Agentcities [4] is an international initiative designed to explore the commercial and research potential of agent-based applications by constructing an open distributed network of platforms hosting diverse agents and services. The ultimate goal is to enable the dynamic, intelligent and autonomous composition of services to achieve user and business tasks, thereby creating compound services to address changing needs.

Inspired by TAC, we developed Travel Agent Game in Agentcities (TAGA) built on the foundation of FIPA technology and the Agentcities infrastructure. The agents and services use standard FIPA supported languages, protocols and services to create the travel market environment providing TAGA with a stable communication environment in which messages in expressive semantic languages can be exchanged. . The travel market is the combination of auctions and varying markets including service registries, service brokerage, wholesalers, peer-to-peer transactions, bilateral negotiation, etc. This provides a much richer test bed for experimenting with agents and web services as well as a rich and interesting scenario to test and challenge agent technology.

2. TAGA Game and Agents

We have designed TAGA as a general framework for running agent-based simulations and games. Our first use of TAGA has been to build a travel competition along the lines of that used in the first three TACs. In this competition, *customers* travel from City A to City B and spend several days there. A *travel package* includes a round-trip flight ticket, corresponding hotel accommodation and ticket to entertainment events. A *travel agent* (an entrant to the game) competes with other travel agents in making contracts with customers and purchasing the limited travel services from the *Travel Service Agents.* Customer selects the travel agent with best travel itinerary. The objective of the travel agent is to acquire more customers, fulfill the customer's travel package, and maximize the profit.

TAGA provides a flexible framework to run the travel market game. Figure 1 show the structure of TAGA. The collaboration and competition among six kinds of agents that play different roles in this market, simulating the real world travel market. We have found that basing our implementation on FIPA compliant agents has made the framework extremely flexible. We'll briefly describe the different agents in our initial TAGA game.

The *Auction Service Agent* (ASA) operates all of the auctions and markets in TAGA. Market types currently include English and Dutch auctions as well as other dynamic markets similar to Priceline and Ebay's fastbuy.

A *Service* Agent (SA) offers travel related service units such as airline tickets, lodging and entertainment tickets. Each class of travel related service has multiple providers with different service quality level and with limited service units;

A *Travel Agent* (TA) is a business that helps customers acquire travel service units and organizes travel plan. The units can be bought either directly from the service agents, or through an auction server.

A *Bulletin Board Agent* (BBA) provides a mechanism through which helps customer agents can find and engage one or more travel agents.

A *Customer Agent* (CA) represents an individual customer who has particular travel constraints and preferences. Its goal is to engage one or more TAs, negotiate with them over travel packages, and select one to try to purchase.

The *Market Oversight Agent* monitors the simulation and updates the financial model after each reported transaction and finally announces the winning TA when the game is over.

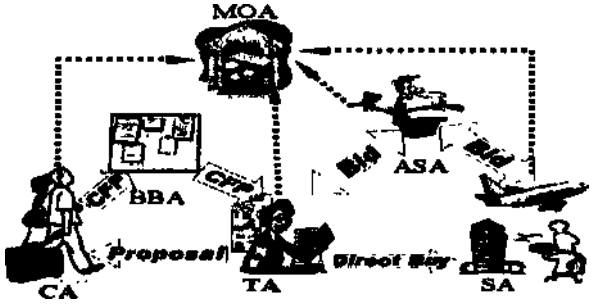

Figure 1: TAGA Architecture

The basic cycle of the game is as follows. A customer-generating agent creates a new with particular travel constraints and preferences chosen from a distribution. The CA registers with the BBA, which facilitates contact with a set of TAs, each of which must decide whether to propose a travel package for the CA. Those that do, contact the necessary ASAs and SAs and assemble an itinerary to propose to the CA. Note that a TA is free to implement a complex strategy using both aggregate markets (ASAs) as well as direct negotiation with SAs. The final proposal to a CA includes a set of travel units, a total price and a penalty to be suffered by the TA if it is unable to complete the transaction. The CA negotiates with the TAs ultimately selecting one from which to purchase an itinerary based on its constraints, preferences and purchasing strategy (which might, for example, depend on a TAs reputation). Once a TA has a commitment from a CA, it attempts to purchase the units in the itinerary from the ASAs and SAs. There are two outcomes possible: the TA acquires the units and completes the transaction with the CA resulting in a satisfied CA and a profit or loss for the TA, or the TA is unable or unwilling to purchase all of the units, resulting in an aborted transaction and the invocation of the penalty (which can involve both a monetary and a reputation component).

3. Discussion

TAC relies on a few centralized market servers to handle all interactions and coordination, including service discovery, agent communication, coordination, and game control. In contrast, TAGA framework uses a distributed peer-to-peer approach based on standard agent languages, protocols and infrastructure components (FIPA [8], Agentcities), emerging standards for representing ontologies, knowledge and services (RDF, DAML+OIL, DAML-S [1]) and web infrastructure (e.g., Sun's Java Web Start). Several FIPA platform implementations are currently used within TAGA, including Jade [2] and APP [5], demonstrating agent interoperability.

We see two contributions in our work. First, TAGA provides a rich framework for exploring agent-based approaches to ecommerce like applications. Our current framework allows users to create their own agent (perhaps based on our initial prototype) to represent a TA, SA and to include it in a running game where it will compete with other system provided and user defined agents. We hope that this might be a useful teaching and learning tool. Secondly, we hope that TAGA will be seen as a flexible, interesting and rich environment for simulating agent-based trading in dynamic markets. Agents can be instantiated to represent customers, aggregators, wholesalers, and service provides all of which can make decisions about price and purchase strategies based on complex strategies and market conditions.

4. Conclusion and future work

As a role-playing market game, TAGA can be used for business research on marketing and auction strategy. In the Agentcities community, TAGA serves as a test-bed for FIPA agent communication in a distributed and open environment. For our own research, has allowed us to explore the integration of multi-agent systems technology (FIPA) and the semantic web.

The Agentcities project is exploring the delivery and use of agent-based services in an open, dynamic and international setting. We are working to increase the integration of TAGA and emerging Agentcities components and infrastructure and will include agents running on handheld devices using LEAP [3]. We are also working to enhance the ontologies which underlie TAGA and to move them from RDF and DAML+OIL to the W3C's Web Ontology Language OWL [5].

Acknowledgments

The research described in this paper is partly supported by DARPA contract F30602-00-2-0591 and Fujitsu Laboratories of America.

References

[1] Ankolenkar et. al., DAML-S: Web Service Description for the Semantic Web, Proc, 1st Int. Semantic Web Conference, Sardinia, June 2002.

[2] F. Bellifemine, A. Poggi, G. Rimassa, Developing multi agent systems with a FIPA-compliant agent framework, in Software - Practice And Experience, 2001 n31, pp. 103-128

[3] F. Bergenti and A. Poggi, LEAP: a FIPA Platform for Handheld and Mobile Devices, ATAL, 2001.

[4] J. Dale et.al., Agentcities: Challenges and Deployment of Next-Generation Service Environments. Proc. Pacific Rim Intelligent Multi-Agent Systems, Tokyo, Japan, August 2002.

[5] J. Dale and L. Coccaroni, Pizza and a Movie: A Case Study in Advanced Web Services. In: Agentcities: Chalenges in Open Agent Environments Workshop, Autonmous Agents and Multi-Agent Systems Conference 2002, Bologna, Italy, July 2002.

[6] M. Dean, et. al.. OWL Web Ontology Language 1.0 Reference. W3C Working Draft.

[7] M. Wellman, et. al., The 2001 Trading Agent Competition. In 14th Conference on Innovative Applications of Artificial Intelligence, pp. 935-941, Edmonton, 2002.

[8] FIPA agent standards, http://www.fipa.ory/