NADIM-Travel: A Multiagent Platform for Travel Services Aggregation

Houssein Ben-Ameur [a],
François Bédard [b],
Stéphane Vaucher [a],
Peter Kropf [c],
Brahim Chaib-draa [d],
Robert Gérin-Lajoie [a],

[a] Center for Interuniversitary Research and Analysis on Organizations
Canada
{benameuh, vauchers, rgl}@cirano.qc.ca

[b] Département d'études urbaines et touristiques
Université du Québec à Montréal, Canada
bedard.francois@uqam.ca

[c] Département d'informatique et de recherche opérationnelle
Université de Montréal, Canada
kropf@iro.umontreal.ca

[d] Département d'informatique et de génie logiciel
Université Laval, Canada
chaib@ift.ulaval.ca

Abstract

With the Internet as a growing channel for travel services distribution, sophisticated travel services aggregators are increasingly in demand. A travel services aggregation platform should be able to manage the heterogeneous characteristics of the many existing travel services. It should also be as scalable, robust, and flexible as possible. Using multiagent technology, we designed and implemented a multiagent platform for travel services aggregation called NADIM-Travel. In this platform, a planner-coordinator agent manages users' requests as well as the response aggregation process. Service agents, on their part, act as gateways to external service providers, and are utilized to retrieve the responses.

Keywords: travel services aggregation, multiagent systems, information gathering.

1 Introduction

Increasingly presented with on-line offers for travel services, users need more sophisticated tools to help them surf through the huge amount of information and to choose the best fares and travel packages. Although information overload has long been an issue in tourism (Hwang et al. 2002), new technologies and business procedures are changing the way this information is accessible. As a consequence, travel agents and professionals are in need of new intelligent systems that: cleverly aggregate travel services, compare fares, dynamically build packages, and assist users in the buying process (Godart 2001; Staab et al. 2002).

The dynamic aggregation of travel services into packages of goods and services can be defined as the packaging of interdependent items and services from various heterogeneous sources (e.g. Websites of airlines, hotels, and car rental agencies; Global Distribution Systems (GDSs), Central Reservation Systems (CRSs)) in ways that best satisfy user constraints and preferences (Haller et al. 2000; Staab et al. 2002). Autonomous agents, regrouped as a multiagent system, are well equipped to track these heterogeneous external sources, to gather distributed information, as well as to coordinate and monitor information retrieval processes.

Some researchers have investigated the use of agent-based architectures to aggregate distributed travel and tourism information. MAPWeb (Camacho et al. 2001) for example is a multiagent planning and Web information-gathering infrastructure that was used to implement a travel assistant application prototype. Another example is Trip-Planner (Homb et al. 1999), an agent-based architecture for trip planning that uses information gathered from the Web to build personalized travel plans.

In this paper, we propose a multiagent platform for travel services aggregation. This platform involves a planner-coordinator agent that receives a travel-related request from a user and is responsible for successfully responding to the request. In addition, service agents act as gateways to the various external travel services providers. These agents use a distributed space we call “infospace” to share information. In its current form, this infospace is a shared virtual marketplace that is used by the agents to post requests and responses.

This paper is organized as follows: In this section (Section 1), we introduced the scope of the paper. In Section 2 we will present multiagent systems and travel services. In Section 3 we will discuss the proposed NADIM-Travel architecture. In Sections 4 and 5 we will outline the functions of the planner-coordinator agent and service agents. Finally, in Section 6 we will present results obtained from our preliminary NADIM-Travel test implementation.

2 Multiagent systems and travel services

2.1 Multiagent systems

An agent is in essence an autonomous software that proactively acts on its environment in order to accomplish tasks and process requests. Many complex distributed applications, if not most of them, require multiple agents, also called multiagent systems (MAS). In such systems, knowledge, action and control are distributed among the agents, which may cooperate, compete or coexist depending on the context. MASs have shown to be valuable for understanding, implementing and operating complex socio-technical systems as represented by e-business systems (Weiss 1999; Jennings et al. 2000).

Agent technology is becoming one of the most important and exciting areas of research and development in computer science today and is significant for applications such as: telecommunications, Internet information management, e-commerce, computer games, interactive cinema, information retrieval and filtering, user interface design, industrial process control, and open systems. The successful adoption of this technology for these applications will have a profound impact on their respective industries as well as on the future conceptualization and implementation of computer systems.

In the following sections, we will present the main characteristics of travel services and explain why a multiagent approach is appropriate for an aggregation platform in this sector.

2.2 Travel services aggregation

When investigating the tourism and travel industry, we tried to identify the main characteristics of its broad range of available services. Travel services are largely heterogeneous, ranging from global legacy systems like Global Distribution Systems (GDSs) to proprietary software run by small hotels. Though highly distributed, these services are increasingly dynamic and available to end users through the Internet.

- Travel services are heterogeneous: The heterogeneity of travel services stems from the fundamental disparity of its business domains (e.g. accommodation, transportation, entertainment, insurance) and their technological differences due to a lack of standards. The majority of the travel industry systems were built as proprietary systems, making it difficult for third party software to interact with them. Organizations like the Open Travel Alliance: OTA (http://www.opentravel.org [September 24, 2003]) and the International Air Transport Association: IATA (http://www.iata.org [September 24, 2003]) are making noticeable efforts to bring some order into the chaos. The standardized XML messages defined by OTA, moreover, are also an important step towards the homogenization of travel services.
- Travel services are highly distributed: It is virtually impossible to centralize all those services in one single place. The travel business being fiercely competitive,

there are now so many service providers and distribution channels that no single GDS can aspire to incorporate them all.

- Travel services are dynamic: The information available for users, travel agents and professionals is very dynamic. For example, prices, availability, and rules change continually. The user needs to have a real-time access to up-to-date information.
- Travel services are increasingly accessible: The Web is an increasingly growing distribution channel for travel services. This makes these services immediately accessible to end users while reducing the market shares of intermediaries. Such public availability tends to transform customer behaviour, with end users "shopping" for travel services.

2.3 A multiagent platform for travel services aggregation

When surfing for best fares, combining items, or constructing holiday packages, users are increasingly in need of tools and assistance. Yet a robust and scalable application for the aggregation of travel services seems difficult to realize due to the above-mentioned characteristics of travel services. While academic prototypes of travel services aggregations have produced interesting results, they lack the scalability and robustness to be effectively implemented in real world industrial applications. In our research, we will focus on the design and implementation of a multiagent travel services aggregation that is robust, scalable, open and flexible.

- Robustness: the platform should be fault tolerant. For example, if an external service provider fails to provide the connection or if a network error occurs, the system should not shut down but continue to run transparently for end users.
- Scalability: the platform should be able to support hundreds to thousands of distributed agents connected to external service providers and serve thousands of users simultaneously.
- Flexibility: agents should be added and removed dynamically without needing to restart the platform.
- Openness: agents should be easily developed and integrated into the platform by third party developers.

We also want the platform to take user needs and preferences into account and to intelligently recommend travel packages and trip plans.

We have named the proposed architecture NADIM: Negotiating Agents in DIstributed Markets. The implemented platform applied to the travel sector is called NADIM-Travel. In the following section, we will introduce the NADIM-Travel architecture in its function as a travel services aggregation.

3 NADIM-Travel architecture

NADIM (Negotiating Agents in Distributed Markets) platform was designed as a distributed multiagent architecture. Two axes are fundamental for developing an

agent-based travel services aggregation: 1) planning and coordination and 2) information gathering. Planning and coordination is executed by planning-coordinator agents (section 4), while information gathering is carried out by service agents (section 5). User requests are processed by the planning-coordinator agent, which asynchronously sends sub-requests to the service agents. Each service agent is connected to a travel service system (e.g. Website, GDS, legacy system, database) that acts as a gateway between NADIM and external systems. The service agents then process the sub-requests and asynchronously respond to the planning-coordinator agent. In a final step, the responses are collected, aggregated and packaged by the planning-coordinator agent and presented to the user. These processes are outlined in sections 4 and 5.

The main idea behind the NADIM architecture is the convergence of shared information into one common "virtual market" called infospace. Requests, responses and all communication between the agents are carried out in this space. It is, in fact, an example of a Linda-like tuple space (Gelernter 1985). The advantages of using Linda-like coordination have been analyzed in the field of mobile agent coordination (Cabri et al. 1998). There are two major reasons for which we believe that a tuple space is the optimal coordination mechanism for a travel service aggregation.

First of all, its associative access mechanism encourages temporal and spatial uncoupling. This uncoupling allows requests to be posted flexibly and without knowing when an answer might come or who might answer it. Interested agents can use the pattern matching features of the space to locate requests they can service. In the proposed architecture, we envision, for example, a coordinator agent requesting the seat availability for flights from Montreal to Paris. In this scenario, respective service agents that are connected to international airlines' Websites could submit responses to the request. The second major reason for propagating a Linda-like space is that the tuples of a space can collectively form complex data structures, as the elements composing these structures are independent of their creators (Carriero and Gelernter 1989).

These two features permit the creation of a graph coordination structure where agents can obtain new requests to respond to. In addition to the classic Linda operations (in, out, read), infospace also offers distributed notification capabilities. Service agents will be notified of requests involving communicating with external sources, while planner-coordinator agents will be notified when service agents publish responses. Agents can also ask infospace to notify them every time a specific type of request is published.

Infospace can be viewed as a "one buyer-many sellers" e-market where participants are softwares rather than humans. The coordinator agent is a buyer, and service agents are suppliers. The planner-coordinator agent acts like a human buyer: (i) it identifies its needs (sub-requests creation), (ii) posts requests and receives responses from

suppliers (service agents), (iii) chooses the best responses and aggregates them, (iv) eventually negotiates better responses, and concludes transactions.

We have implemented a generic infospace using Sun's JavaSpaces technology (http://java.sun.com/products/javaspaces [September 24, 2003]). A JavaSpace is a shared memory that allows distributed processes to write and read objects as entries. A process may register into a JavaSpace with a specific template, requesting to be notified every time a written entry in the JavaSpace matches that given template.

A virtual aggregated market exchange between all agents is illustrated in Fig. 1. Requests are sent to the planner-coordinator agent, which divides them into sub-requests using specific local domain knowledge. The sub-requests are then published in the infospace. Service agents read the requests from the infospace and retrieve responses from distant service providers' systems. The service agents proceed by publishing the responses in the infospace. These responses are evaluated and aggregated by the planner-coordinator agent. In a final step, the global responses are presented to the end user.

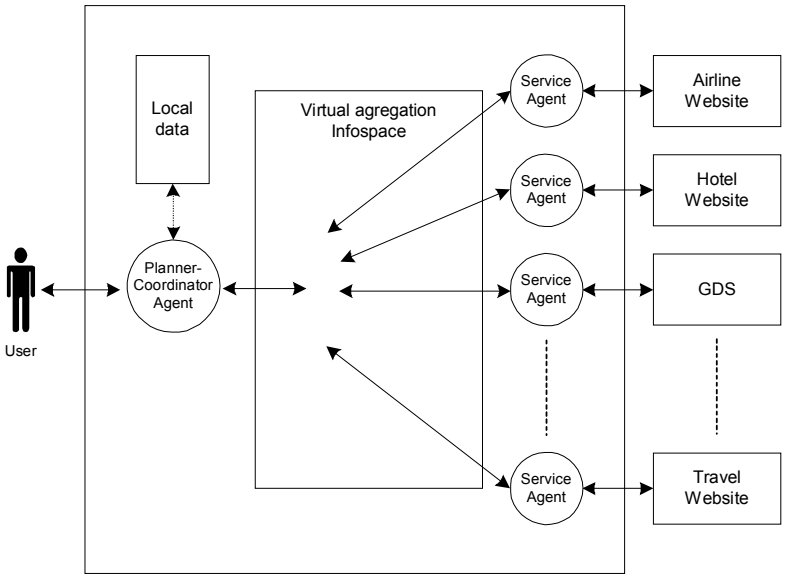

Fig. 1. NADIM-Travel multiagent architecture

4 The planner-coordinator agent

The planner-coordinator agent is in charge of managing user requests. Fig. 2 shows the agent architecture to be discussed in this section.

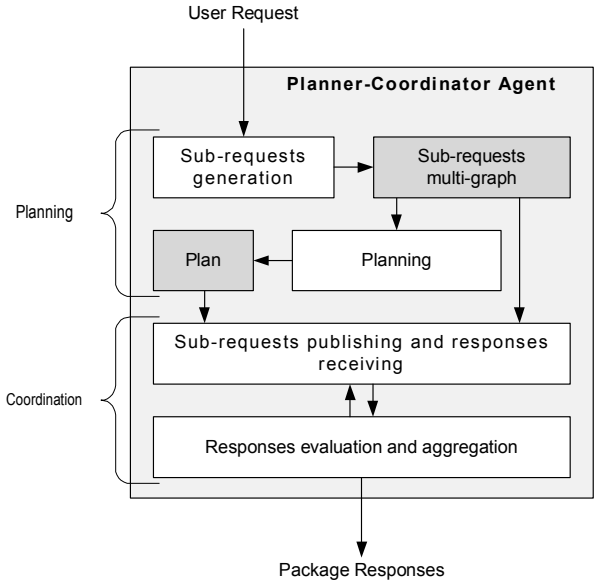

Fig. 2. Planner-Coordinator agent architecture

4.1 Creating structured sub-requests

The coordinator agent first extracts a global request from a user package request and then divides it into sub-requests, each of which represents an item from the package. Splitting up user requests is not a simple task since users might not explicitly ask for a complete and well-defined combination of items and services. In order to process an incomplete request, the agent needs to extrapolate implicit sub-requests using its knowledge of the travel sector. For example, a request for a vacation package leaving Montreal and visiting Paris and London, should be divided into sub-requests for finding: a transatlantic flight, local transportation between requested destinations (by bus, train, ferry, domestic flight, or rental car), hotel reservations, and other package items such as concert or museum tickets.

The agent should arrange the sub-requests in a structured manner that corresponds to the dependencies and relations between the package items. We propose to structure sub-requests into a multi-graph (a graph with one or many arrows (or lines) between two nodes), with the arrows identifying the sub-requests. This structure enables us to express AND/OR relations between sub-requests.

Let us suppose, for example, that a traveller wants to plan a roundtrip travel from Montreal to Paris and London, including a hotel reservation in Paris and London. This request may be divided into the following sub-requests including i) roundtrip flight from Montreal to Paris, ii) roundtrip flight from Paris to London *OR* roundtrip train ticket from Paris to London, iii) hotel reservation in Paris, and iv) Hotel reservation in London. In NADIM-Travel, these sub-requests would be represented by a multi-graph

as shown in Fig. 3 where a single line between two nodes expresses an AND relation, while multiple lines between two nodes expresses an OR relation. Of course this is a very simple example; in real situations the graph could be much more complex.

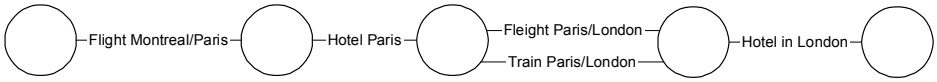

Fig. 3. Structured representation of sub-requests into a multi-graph

The planner-coordinator agent uses local information to build the sub-requests multi-graph. This local information is a knowledge base about the travel sector and includes facts such as "*it is possible to take the train to go from Paris to London"* as well as geographical locations and distances.

4.2 Planning the information retrieval process

Once sub-requests are extracted and structured into a multi-graph, the agent has to plan the execution of these sub-requests. At this point, the coordinator decides which ones should be published concurrently and which ones should be published sequentially. In the travel package example shown in 4.1, it is obvious that it should look for transatlantic flights availabilities before looking for hotel reservations and train availabilities. This decision may also be determined by the fact that information about transatlantic flight schedules and prices is readily available and does not require any commitments.

4.3 Coordinating and monitoring information search

After generating a plan, the planner-coordinator agent will execute it. At this point, the coordinator publishes the structured sub-requests multi-graph in the infospace and then activates the first sub-requests according to its plan. After that, it is notified when service agents publish new responses. Depending on feedback from an evaluator module, it can activate new sub-requests, deactivate previously activated ones, or modify them. The coordination is carried out using a loosely coupled publish-subscribe model, with the agent unaware of the entities it might interact with. It is for this reason that it cannot be qualified as a completely centralized coordination. The agent's decisions depend only on its own request, which is to find the best solution for a requested package. We call it an implicit semi-centralized coordination.

4.4 Evaluating and aggregating responses

An evaluation module of the planner-coordinator agent carries out the response aggregation by taking into account the user needs and preferences as expressed in the request phase. A utility function is used to evaluate individual responses and sub-requests and to aggregate them into a global response for the initial user request.

5 Service agents

Service agents are the system's gateway to external sources of travel services. We propose an agent factory that instantiates service agents with the required capabilities and knowledge for interacting with external service providers. These agents are cognizant of the source's market model and of the protocols it uses. They are able to interact with a source, request information, eventually start negotiations, and conclude transactions. A service agent is able to determine which requests it can service. It proactively reads these requests and tries to find an acceptable answer. It then posts its responses or results in the infospace. The complexity of these agents can vary tremendously, ranging from simple translator slaves to sophisticated agents that carry out complex negotiations.

We have implemented some service agents as wrappers to a series of travel Websites. Using a technique called "screen scraping", the service agent take an XML-OTA request message from the infospace, extract data such as departure and arrival locations, and then submit this information to the travel Website. The site then returns an HTML response. Using an XSLT processor (http://www.w3.org/TR/xslt [September 24, 2003]), the agent translates this response into an XML-OTA message and then publishes it in the infospace. In this sense, the service agent can be seen as a translator between semi-structured data (HTML pages) and structured data (XML-OTA messages). Fig. 4 illustrates this service agent's transformation chain.

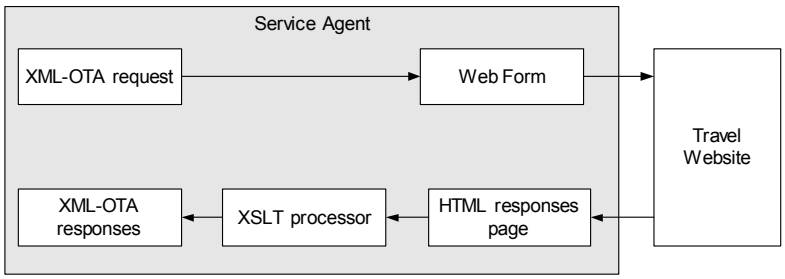

Fig. 4. Website connector service agent

6 Some results

We have launched a NADIM-Travel prototype that connects to twelve Canadian travel Websites. Presently three travel agencies in Montreal use the prototype for beta-testing purposes. The prototype helps these agencies access Web airfares that are not available on GDSs. NADIM-Travel was deployed following a SAP (Service Application Provider) model, which means that it runs on central servers that are remotely accessed (via Web interfaces) by the travel agencies. For the time being, the travel agencies are not testing the packaging functionality. NADIM-Travel is running under Linux on a Pentium III, 866 Mhz server, with 1 Gig of Ram.

We are capable of quickly designing and implementing new service agents (within one or two days depending on the complexity of the travel Website). Agents may be added in run-time without having to restart the platform. Moreover, agents may be duplicated during run-time if an access overload is noticed insuring the platform's robustness and flexibility.

In order to evaluate agents' access and data translation time, we also carried out performance tests on four agents that were respectively connected to Expedia.ca, AirCanada.ca, FlyTango.com, and JetsGo.net. The test generated 500 random requests, sending one to every agent every 30 minutes.

For a given user request, agents' response time was found to depend on four processing steps.

- *Step 1*: Translating the request, then filling out the Website form and returning it.
- *Step 2*: Waiting for Website to send back the HTML page(s) containing the responses.
- *Step 3*: Extracting the required information from the HTML, then translating it into XML-OTA responses.
- *Step 4*: Posting responses in the JavaSpace.

The table below shows each agent's mean processing time for the four steps.

Agent	**Step 1**	**Step 2**	**Step 3**	**Step 4**	**Total**
Expedia Website agent	0.077 sec	7.989 sec	5.379 sec	0.209 sec	13.654 sec
JetsGo Website agent	0.067 sec	3.588 sec	0.991 sec	0.166 sec	4.812 sec
FlyTango Website agent	0.107 sec	5.433 sec	0.910 sec	0.128 sec	6.578 sec
Air Canada Website agent	0.041 sec	1.732 sec	0.497 sec	0.026 sec	2.296 sec

One important result is that agents' response time depends mostly on the second step, i.e. the waiting time for the Website response. Extracting and translating the information from HTML to XML-OTA (step 3) takes less than 1 second (mean-time). In the case of the Expedia.ca agent, that step took more than 5 seconds (mean-time). This is due to the fact that HTML pages from the Expedia.ca Website are voluminous and erroneous (many HTML errors), requiring that the agent take the time to clean the data before being able to process it.

The graph below illustrates the processing time distribution for the Air Canada Website agent. The peaks that can be observed for step 2 (waiting for the Website response) are mostly due to high traffic access to the Website.

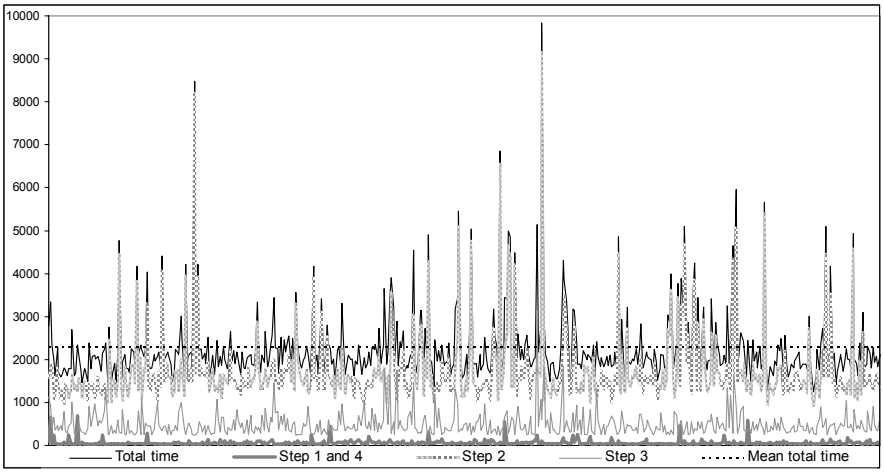

Fig. 5. Processing time distribution for the Air Canada Website agent

The asynchronous model of NADIM-Travel allows the user to see agents' responses as soon as these are issued. That way, users don't have to wait for the slowest agent's response to be able to see fastest one.

We were able to run NADIM-Travel for one month, stopping and restarting agents without platform interruptions or system crashes. We think that we succeeded in developing a flexible and robust platform for the aggregation of travel services. The agent-based architecture assured a better modularization and decoupling of the platform. The infospace paradigm has proven to be efficient in this case, allowing a flexible distribution of the platform on different machines for better performance. Screen scraping agents based on XSLT are a good approach to translate semi-structured HTML into structured XML-OTA documents. While the response delay time for the agents is skewed by the remote access time to Websites, other processing steps proved to be very fast.

7 Conclusion

In this paper we presented a novel approach for travel services aggregation. The proposed architecture is based on multiagent technologies, where the autonomy of agents makes the platform robust, scalable and dynamic. The development of the NADIM-Travel prototype is not finished. In the near future, we will implement more service agents that connect to hotel, car rental and entertainment Websites. In addition, we are still working on the planner-coordinator agent. We aim to build sophisticated algorithms for the dynamic packaging of on-line travel services.

We are also investigating the integration of Web services technology into the current NADIM-Travel architecture. NADIM-Travel's agents would thus have a Web service façade that significantly opens the platform to external systems. With integrated Web

services technology, NADIM-Travel would also be easier to integrate and use by third party users.

Acknowledgments

"Valorisation Recherche Québec" project "Prototypes Avancés en Commerce Électronique", supports this research with the partnership of Synopsis and Voyatech enterprises. The research is carried out at CIRANO (Centre Interuniversitaire de Recherche en ANalyse des Organisations). We particularly thank Sylvain Riopel from CIRANO for his inestimable work in the project.

References

Cabri, G., Leonardi, L., & Zambonelli, F. (1998). How to Coordinate Internet Applications based on Mobile Agents. *Proceedings of the 7th IEEE Workshop on Enabling Technologies: Infrastructures for Collaborative Enterprises (WETICE 98)*: 104-109.

Camacho, D., Borrajo, D., Molina, J. M., & Aler, R. (2001). Flexible Integration of Planning and Information Gathering. *Proceedings of the European Conference on Planning* 2001: 73-84.

Carriero, N., & Gelernter, D. (1989). Linda in Context. *Communications of the ACM* 32(4): 444-458.

Gelernter, D. (1985). Generative Communication in Linda. *ACM Transactions on Programming Langages and Systems* 7(1): 80-112.

Godart, J. M. (2001). Using the Trip Planning for Computer-Assisted Customization of Sightseeing Tours. *Proceedings of the International Conference on Information and Communication Technologies in Tourism, ENTER 2001*: 377-386.

Haller, M., Pröll, B., Retschitzegger, W., Tjoa, A M., & WagnerIntegrating, R. R. (2000). Heterogeneous Tourism Information in TIScover - The MIRO-Web Approach. *Proceedings of the International Conference on Information and Communication Technologies in Tourism, ENTER* 2000: 71-80.

Homb, A., Mundhe, M., Kimsen, S., & Sen, S. (1999). Trip-planner: An Agent Framework for Collaborative Trip Planning. *Proceedings of the AAAI-99 Workshop on Mixed-Initiative Intelligence.*

Hwang Y. H., Gretzel, U., & Fesenmaier, D. R. (2002). Behavioral Foundations for Human-Centric Travel Decision-Aid Systems. *Proceedings of the International Conference on Information and Communication Technologies in Tourism, ENTER 2002*.

Jennings, N. R., Faratin, P., Norman, T. J., O'Brien, P., & Odgers, B. (2000). Autonomous agents for business process management. *International Journal of Applied Artificial Intelligence* 14(2): 145-189.

Staab, S., Werthner, H., S., Ricci, F., Zipf, A., Gretzel, U., Fesenmaier, D. R., Paris, C., & Knoblock, C. (2002). Intelligent Systems for Tourism. *IEEE Intelligent Systems, Trends & Controversies* 17(6): 53-64.

Weiss, G. (1999). *Multiagent Systems*. MIT Press.