Visitor Guide System (VGS) Based on Agent Layer Architecture

Nwe Mar Khaing
University of Computer Studies (Mandalay)
nwemarkhaing87@gmail.com, myintmyintmaw09@gmail.com

Abstract

The recent developments focus on context awareness and information presentation to individual or group of visitors. Any visitors will not interest too long on the road, they want to take the shortest tour for his desired trip. At this point, Visitor Guide System support as a guide without asking guide. Our proposed system is intended to evaluate the visitor guide system for interesting places of district in Myanmar. In our visitor guide system (VGS), suggested two pass control structure based on three vertical layer architecture and multi-agent systems that consist of a collection of functional agents. An agent can construct the sequence of actions to achieve its goal. In this paper, such system would facilitate cooperation and increases its effectiveness and also serve as a basic development of visitor guide. Visitor Guide System (VGS) take places to be visited with selected route and source to destination route to find the shortest path by using A search strategy without time consuming. A* search is one kind of heuristically informed search strategy that reduce the search cost.*

Keywords: Multi Agent, Three Vertical Layer Architecture, A Search Strategies*

1. Introduction

Nowadays, travel planning (TP) is becoming a vast field and for economy of any country it plays an important role. Travel Planning is searched for better routes, source and destination route system enabling the visitor to build his best travel plan for his desired trip. At this point, Visitor Guide System [1] (VGS) support visitors to choose the right path and time saving to visit a number of district in Myanmar.

The system is vital to success among the visitor and administrator to communicate and interaction. The underlying three agent layer architecture, named Visitor Guide System (VGS), provides a set of based agent organized around a presentation layer, intermediate layer and data layer. Agents are defined as autonomous entities capable of flexible behavior denoted by reactiveness, pro-activeness, and social ability. Multi agent system (MAS) consists of diverse agents that communicate and coordinate to pursue a common goal.

The description of the overall multi agent architecture for detailing the agents and interface involved from the visitor side and administrator side. The used underlying optimization problem and schedule heuristic are depicted together with some performance tests on the planning of the system. There are various heuristic searches in problem solving. From the various approaches, A* search has been chosen for VGS development use. The system can offer the result obtained using this heuristic will minimize the distance traveled.

This paper is organized as follows: first, Related Work is involved in section 2. Theory Backgrounds are introduced in section 3. System Architecture dealt with three vertical layer architecture based on agent in section 4. And section 5 describes conclusion.

2. Related Works

Travel planning take approach of providing tools for selecting hotel, interesting places, Myanmar traditional gift shops, car transportation and rentals in steps. Yeung et al. present a Multi-Agent based Tourism Kiosk for Hong Kong based on Internet Information categories with the Knowledge Query and Manipulation Language (KQML) as the agent communication language. Carig A. Knoblockn presents Building Software Agents For Planning, Monitoring and Optimizing Travel [7].

The state-of-the-art research in the field of transportation scheduling, cluster-first and route-second planning techniques have been widely covered. Borndorfer et at.[2] presented such a two-phase approach applied to several instances provided by an operator in Berlin. Search techniques have

also been applied by J.Jaw [3] presenting a local search variant based on the direction of solutions with the neighborhoods of feasible solutions.

3. Theory Background

3.1 Agent

An agent is a computer system, situated in some environments, that is capable of flexible autonomous action in order to meet its design objectives [4]. Other define agent as:

- "a system that independently handles parts of the problem based on small independent knowledge based"[6].
- "an autonomous entity that interacts with the environment, and adapts its state and behavior based on interaction" and
- "an agent is a computational entity.

3.2 Agent Architecture

The term agent architecture intuitively suggests a framework for the implementation of the agent. Agent architecture considers the issues surrounding the development and implementation of an agent based on a selected theoretical foundation and can be more formally defined as for building agent. It specifies how the agent can be decomposed into construction of a set of components modules and how the modules should be made to interact. Architecture encompasses techniques and algorithms that support the methodology.

3.3 Hybrid Agent Architecture

An agent is a capable of reactive and proactive behavior an obvious decomposition involves creating separate subsystems to deal with these different types of behaviors. This deal leads to a class of architectures in which the various subsystems are arranged into a hierarchy of interacting layers. Two types of control flow within layer architecture as follows:

Horizontal layering: In horizontal layering, software layers are each directly connected to the sensory input and output. Each layer itself acts like an agent, producing suggestions as to what action perform. This layering approach, all layers are at the same level and execute independently.

Vertical Layering: In vertical layered architecture, sensory input and action output are each deal with by at most one layer. All layers are hierarchically ordered with the complexity of layers increasing with their level. The interaction between the bottom layer and top layer cannot be direct and has to be done through intermediate layers. In this approach, all layers are executed in parallel.

3.4 Advantages of Vertical Layer Agent Architecture

Layering is a powerful means for structuring functionalities and control and thus is a valuable tool for system design supporting several desired properties such as reactivity, deliberation, cooperation and adaptability. The main idea is to Structure the functionalities of an agent into two or more hierarchically organized layers that interact with each other in order to achieve coherent behavior of the agent as a whole. The benefit of agent layer architecture can be discarded software components that unwanted. The agent layer provides the legacy system because layer enabling them to communicate and cooperate with other software components. The benefits of layer architecture based on agent:

- supports design based on level of abstraction,
- support enhancement can add layer without affecting lower-level layers and
- support reuse different implementations of a layer should be interchangeable.[5]

4. System Architecture

Our proposed system consists of three layer agent architecture. These are presentation layer (user assistant agent), intermediate layer (information broker agent) and data layer (database agent) as illustrated as figure 1. These layers are hierarchically ordered form bottom layer to upper layer. These agents communicate in order to provide relevant information to the visitor based on his/her location. The initial system is composed of three agents:

- User Assistant Agent (UAA) runs on user's location and provides the user with interface to the system. The main role of User Assistant agent is to assist user with relevant suggestions, which are applicable on current user situation. The UAA helps users in collaboration and sharing information. The UAA send request information to Information Broker Agent by upward activation. And then accept the display result to the visitor by downward commitment from Information Broker Agent.
- Information Broker Agent (IBA)is a key agent which execute location from user assistant agent with A* search method to find the optimal route for visitor. A web application protocol is used to

complement the communication between the visitor (UAA) and the Information Broker Agent, for sending request and transferring the actual presentations of selected location. The HTTP protocol provide the web interface for receiving and sending request from the UAA and retrieving by activating that access request to the Database Agent.

- Database agent is a data layer. This layer contains all available information presentation of 60 districts which responds to the presentation request by providing list of those available. The information broker agent in intermediate layer is retrieved the required data from the database agent by upward activation and reply information by downward commitment. This database provides access to data that is maintained within the boundaries of the system about the district of Myanmar.

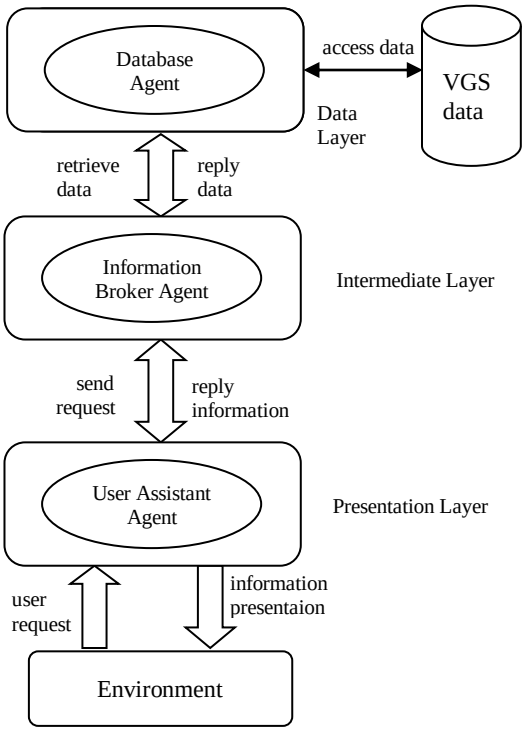

Figure1. Three Vertical Layer Agent Architecture

4.1 Agent Interaction Process

Figure 2 illustrates the agent interaction process of proposed system. In this process, includes visitor side and administrator side. In visitor side, the visitor can get freely to enter the system. The visitor can choose to find the shortest path between source to destination or pass through for their desire trip. When the visitor sends request to the User Assistant (UA) agent in presentation layer, it cooperates with Information Broker (IB) agent. This agent search the A* method to achieve the goal and to satisfy the visitor requests with information and the required information is retrieved from database (DB) agent. A* is a informed (heuristic) search strategies and to find the shortest path is used straight line distance. Instead of searching the distance itself, the system displays the result that the shortest path of district in Myanmar and display relevant information and image for each district.

At the administrator side, the system can edit, add or delete the location of district latitude and longitude, straight line distance and the distance of location between the district and adjacency district. Uses the required data of district are described in next section 4.2.

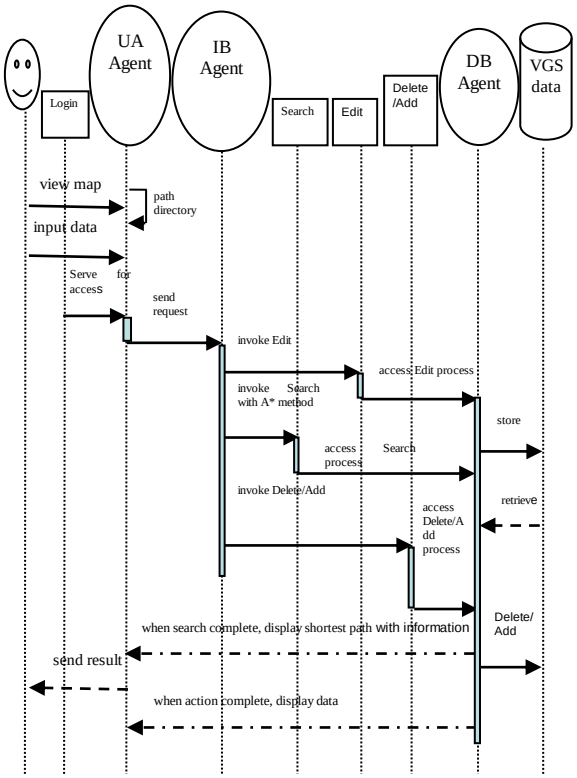

Figure 2. Sequence Diagram of Agent Interaction Process

4.2 Route Maintenance of VGS Database

In this section, the system uses the following table in the database. Locations are shown with the city (district) name and Latitude and Longitude. Before finding the shortest path from one district to another district, must calculate heuristic straight line distance h_{SLD} values. h_{SLD} is straight line distance between source city (district) to destination city (district). Straight-line distance is admissible because the shortest path between any two districts is a straight-line. Table 1 is Heuristic Straight Line Distance table from Mandalay (district) to other city extracted from the database. A* algorithm replace this value h(x)

into the equation.

A* Algorithm description

The algorithm searches the routes that appear to be most likely towards the goal. The algorithm traverses various paths from start to goal for each node x traversed, it maintains three values. These are

- g(x): the actual shortest distance traveled from initial node to current node.
- h(x): the estimated or heuristic distance from current node to goal.
- f(x): the sum of g(x) and h(x).

1. Starting with the initial node $\mathbf{x_0}$. Put the initial node and the set of tentative node to be traversed on a list called OPENSET.
2. Create a list called CLOSEDSET that is initially empty.
3. If OPENSET is empty, exit with failure. If not so, execute the following step.
4. Select the first node on OPENSET, remove it from OPENSET, and put it on CLOSEDSET. Call this node **x**. **x** is the node in OPENSET having the lower f-values.
5. If **x** is a goal node, exit successfully with the solution obtained by tracing to reconstruct the path by working backwards from the goal node along the pointers from $\mathbf{x_o}$ to $\mathbf{x_0}$.
6. Expand node **x**, generating the set, **Y**, of its neighbors that are not already ancestors of **x**. Install these members of Y as neighbors of **x**.
7. Establish a pointer to **x** from each of those members of **Y** that were not already evaluated. Add these members of **Y** to OPENSET. For each member, **y**, of **Y** that was already on OPENSET or CLOSEDSET, redirect its pointer to **x** if the best path to **y** found so far is through **x**. For each member of **Y** already on CLOSEDSET, redirect the pointers of each of its descendants so that they point backward along the best paths found so far to these descendants.
8. Go to step 3.

Table 1. Heuristic Straight Line Distance

City Name	Latitude (N)	Longitude (E)	h_{SLD}
Bago	17˚18’	96˚28’	323
Magway	20˚27’	94˚33’	148
Mandalay	21˚58’	96˚05’	0
Mawlamyine	16˚29’	97˚37’	393
Myingyan	21˚27’	95˚23’	68
Nyaung-U	21˚11’	94˚54’	93
Taungoo	18˚56’	96˚25’	213
Thaton	16˚55’	97˚22’	360
Yangon	16˚47’	96˚09’	358

5. Conclusion

In this paper, present the multi-agent based on vertical layer architecture and informed search strategies for searching location of district in Myanmar to guide for visitors. By using the vertical layer architecture, the system can modify as needed and can be discarded that unwanted. This system demonstrates how the complex requirements of a highly visitor guide system are supported for the visitor to visit the selected district of desire trip. The system can provide the visitor as a decision maker whether or not to visit the selected district within the shorter time. As a result, the system able to propose an agent based on vertical layer architecture that may facilitate the development of visitor guide system.

References

[1] Belloti F., Berta R, de Gloria A and Magarone M. (2002) User Testing a Hypermedia Tour Guide, in IEEE, April-June.

[2] Borndofer, R., Grotschel, M., Klostermeier,F., Kittner, Telebus Berlin: “Vehicle Scheduling in a Dail-a-Ride System. Berlin,1997.

[3] Jaw, J.et al.: “A heuristic algorithm for the multiple-vehicle advance request dial-a-ride problem” Transportation Research, 1986.

[4] Jening, N.R, K.Sgeara and Wooldridge.M. “A Roadmap of Agent Research and Development”, In Autonomous Agent and Multi-Agent system, 1998.

[5] Shen Weiming, Douglas H. Norrie and Jean-Paul A.Barthes “Multi-agent system for concurrent intelligent design and manufacturing”.

[6] Wooldrige, M. and N.R Jennings. “Intelligent Agents Theory and practice”. The Knowledge Engineering Review, 10(2), 1995

[7] Yeung, C., J. Yen and P-F Tung, " A Multi-Agnet Based Tourism Kiosk On Internet "