A Mobile Agent-based Computing Environment for Pedestrian Tracking Simulation

Rong XIE, Prof. Ryosuke SHIBASAKI
Center for Spatial Information Science, The University of Tokyo
4-6-1 Komaba, Meguro-ku, Tokyo 153-8505
{xierong, shiba}@iis.u-tokyo.ac.jp

Abstract: The study of pedestrian behavior covers wide topics, including way finding, choice and decision make, as well as spatial cognition and environmental perception. To address the problem, simulation is now put forward as suitable technique and method for analyzing human spatial behavior. In the paper we present a development architecture for simulating tracking pedestrian in a distributed environment. We introduce and explore the potential of using mobile agent-enabled distributed implementation model as a tool for development and implementation of the simulation. Three kinds of mobile agents are designed for implementation of managing and querying data of pedestrian. Finally, simulation result of JR 10,000 passengers' movement is developed and implemented as a case study.

Key words: Human Spatial Behavior, Pedestrian Tracking Simulation, Distributed Computing, Mobile-Agent.

1. Introduction

The study of pedestrian behavior covers wide range of topics, including way finding, human migration, choice and decision make, as well as spatial cognition and environmental perception. To address the problem, simulation is now put forward as suitable technique and method for analysis of such human spatial behavior.

In this paper we present a development architecture for simulating tracking pedestrian in a distributed environment. Simulations have been built up for exploring complex pedestrian behavior in urban centers.

Our approach is based on the integration of three modules: organization of distribution data of pedestrian flow, distributed query processing, and simulation of human crows in distributed environment. We introduce and explore the potential of using mobile agent-enabled distributed implementation model and simulation as a tool for integrating these modules and examining the complex interactions between pedestrian and environment.

Some suggestions for development and implementation of the simulation are discussed in the paper. The distributed computing environment among these modules is established through a paradigm of mobile agent architecture. Three kinds of mobile agents, such as migration agent, information agent and collaboration agent are designed for implementation of managing and querying data of pedestrian. Finally, simulation result of JR 10,000 passengers' movement is developed and implemented as a case study in the paper.

2. Mobile Agent-enabled Distributed Implementation

1) Why Mobile Agents?

Recently, it is widely recognized that mobile agent technology is a promising design paradigm for network computing, which is also a natural successor to the object-oriented paradigm. Mobile agent is the basis of an emerging technology that promises to make it very much easier to design, implement, and maintain distributed systems.

We are interested in it by the benefits of its providing for the creation of distributed systems. [1] gives seven good reasons for mobile agent technology. Some advantages of mobile agent related to our research are: (1) to reduce the network load; (2) to provide load balancing; (3) to be robust and fault-tolerant.

The FIPA (Foundation for Intelligent Physical Agents) specification [2] is developed through direct involvement of the FIPA membership. It provides specification of basic agent technologies that can be integrated by agent systems developers to make complex systems with a high degree of interoperability. Based on this standard, we propose a mobile agent-enabled distributed implementation model.

2) Agent Management System

- Manager Agent

An organizational structure framework of agent management system is proposed in Fig. 1. It describes execution mode of an operation process. In this structure, an agency is defined as a manager agent with one (or null) set of tasks.

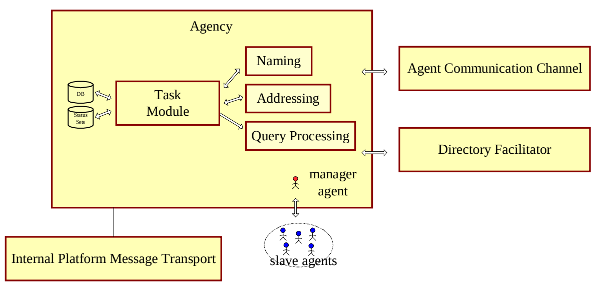

Fig. 1. Agent Management System Architecture

Agentcy = manager agent + task;
task = { $task_i$ };
$task_i$ :: =
<task_name><input><output><condition><implementation body>.

Manager agent provides an execution environment for other slave agents. It manages some mobile agents, deciding to establish what kinds of agents, their address to the remote node and tasks mobile objects should do. Fig. 2 gives UML representation of manager agent.

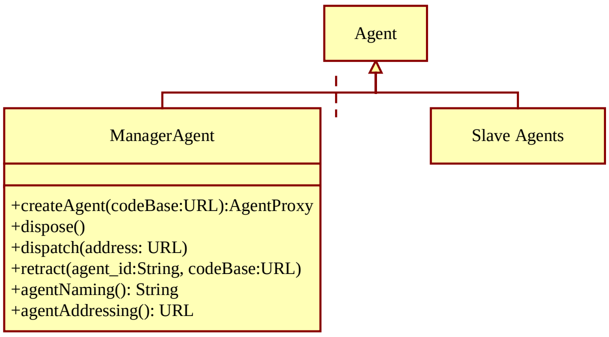

Fig. 2. UML Representation of the Manager Agent

- Slave Agents

Slave agent is a computer program that can accept tasks from its manager agent and dispatch itself to different nodes to search for suitable computational resources in order to finish the assigned tasks.

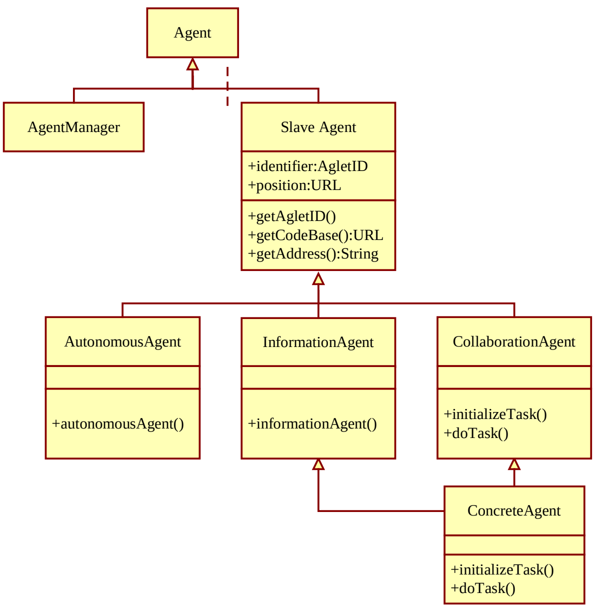

Fig. 3. UML Representation of the Slave Agents

In light of the various characteristics of agents, slave agents are classified into the following three categories of agents as described in Fig. 3, Among them, autonomous agent can migrate between hosting system to enhance the efficiency of computation and reduce the network traffic. While information agent plays the role of managing, manipulating or collecting information from many distributed sources. Collaboration agent's major characteristic is that it cooperates with other agents.

3. Distributed Query Processing

To implement distributed query processing, we can have two kinds of agent: (1) To design an information agent, and dispatch it to each server or node database to search for information and implement tasks. After agent finishes its tasks, it will come back with the result information. (2) In many applications, we usually design collaboration agents. Firstly, to divide a task into several sub-tasks and to assign them to multiple agents. After these agents finish their tasks, they will come back with result information. The result information will be combined to form the final result to show to user.

A typical query format submitted might be "report all objects that reside inside the region P at the time instants between time t_{1q} and t_{2q}, given the trajectories information of all mobile objects". The process of distributed query is shown as in Fig. 4. When user submits request, the following steps is conducted to reach the purpose: (1) receiving query request; (2) node determine and agent dispatching; (3) local data processing; (4) result returning and trajectory simulation.

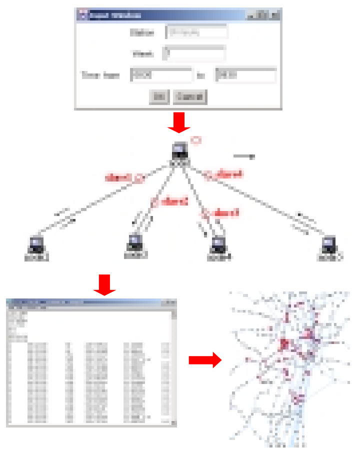

Fig. 4. Distributed Query Processing

1) Agent Design

After receiving quest request, implementing the lookup operation to decide location of nodes, and then implementing the collaboration operation to create several slave agents to dispatch to the remote nodes.

2) The Query Operation

JDBC is the database connectivity package included in the core Java API.*. JDBC gives you a database-independent interface for opening a connection to a relational database, issuing SQL calls to the database, and receiving a set of data as the results.

Handling query operation at local node after slave agent arrives at the local node. The following example shows a Java code segment that opens a database connection, executes a query and iterates through the results.

```
String url ← "jdbc:odbc:event";
Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");
Connection con ← DriverManager.getConnection(url);
Statement sm ← con.createStatement();
String tempstr ← "Select column_expression [,
                              column_expression…] From tablename
[WHERE logical expression];
ResultSet rs ← sm.executeQuery(tempstr);
while(rs.next())
      // ……
repeat
QueryResult result ← new QueryResult();
collaborate(new AgentResult(getAgentID()), result);
```

4. Pedestrian Tracking Simulation and Analysis

1) Experimental Data

We apply our approaches to a case study of 10, 000 passengers' movement in JR railway stations. In this case study, investigation data of JR 10000'98 passengers' movement from JR Higashi Nihon Kikaku [3] are used for experimental data. The whole project is organized by JR Higashi Nihon Kikaku and conducted during September to November in 1998 to know personal movement characteristics and pattern everyday. The study area is located inside the range of approximately 70 square km in Tokyo, including Tokyo Prefecture, Shinnagawa Ken, Saitama Ken, Chiba Ken, and Ibaraki Ken. With the help of questionnaire surveys, information about personal travel behavior by railway is recorded. The whole sample comprised 10,000 with ages ranging from 12 to 69 years. We organize these data in database and store them distributedly according to railway line.

2) Movement Pattern Query

- Personal Travel Simulation in One Day

SELECT line_in, station_in, time_in, line_out, station_out, time_out
FROM *nodei*.route
WHERE object_id = *object_id* AND day = *day*

- Passengers' Movement in JR Shinjuku Station in One Day

SELECT line_in, station_in, time_in, line_out, station_out, time_out
FROM *nodei*.route
WHERE day = *day* AND station_id = *Shinjuku_id* AND (time >= *time_in* AND time <= *time_out*

- Comparison of Passengers' Movement among JR Stations at Rush Hour

SELECT line_in, station_in, time_in, line_out, station_out, time_out
FROM *nodei*.route
WHERE day = *day* AND (station_id = *Shinjuku_id* OR station_id = *Ikebukuro_id* OR station_id = *Tokyo_id*) AND (time >= *time_in* AND time <= *time_out*)

- Passengers' Movement Simulation on JR Yamanote Line

SELECT line_in, station_in, time_in, line_out, station_out, time_out
FROM *nodei*.route
WHERE day = *day* AND (station_id = *station_id_onYamanoteLine*) OR (line_id = *YamanoteLine_id*) AND (time >= *time_in* AND time <= *time_out*)

3) Simulation Result

On the basis of these movement pattern queries, visualization can be further implemented for the analysis spatial behavioral pattern, such as distribution of passengers, trajectory simulation of passengers' movement. Here we give a simulation result as follows. In Fig. 5, simulation result shows two situations, one is that passengers use Yamanote line, the other is that passengers transfer, enter or leave stations on Yamanote Line, such as Ueno Station, Akibahara Station Tokyo Station etc. at given time period, example here represents snapshot of crowd situation and passenger density on this line at rush hour at about 8:30 am in the morning.

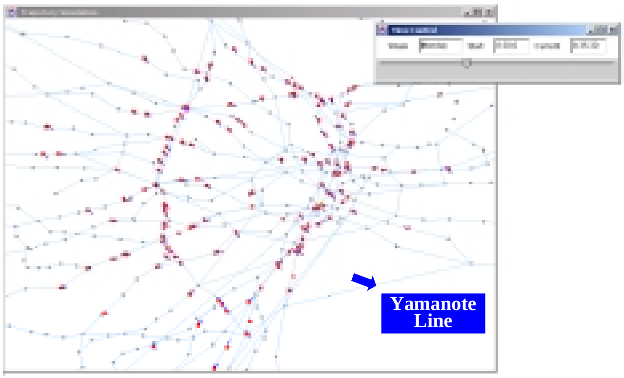

Fig. 5. Snapshot of Passengers' Movement on JR Yamanote Line

5. Concluding Remarks

We propose a development architecture for development and simulation of tracking pedestrian by mobile agent-enabled distributed implementation model. The simulation result shows the proposed model can be used as tool for exploring complex pedestrian behavior in distributed environment.

References

[1] Danny B. Lange, 1998. Mitsuru Oshima. Programming and Deploying Java Mobile Agents with Aglets, Addison-Wesley.
[2] FIPA, 1998. FIPA 98 Part 1 Version 1.0: Agent Management Specification, Foundation for Intelligent Physical Agents. Available at: http://www.fipa.org/specs/fipa00002/.
[3] URL: JR Higashi Nihon Kikaku Corporation. Available at: http://www.jeki.co.jp/marketing/.
[4] R. XIE, 2003. A Study on Data Modeling for Mobile Object Management and Distributed Simulation. Doctoral dissertation, The University of Tokyo.