http://ansinet.com/itj

ITJ

ISSN 1812-5638

INFORMATION TECHNOLOGY JOURNAL

ANSInet
Asian Network for Scientific Information
308 Lasani Town, Sargodha Road, Faisalabad - Pakistan

Information Technology Journal 10 (6): 1113-1125, 2011
ISSN 1812-5638 / DOI: 10.3923/itj.2011.1113.1125

Embedding a Multi-agents Collaboration Mechanism into the Hybrid Middleware of an Intelligent Transportation System

[1]Steven K.C. Lo and [2]Huan-Chao Keh
[1]Department of Information Management, Jinwen University of Science and Technology, New Taipei City 231, R.O.C, Taiwan
[2]Department Of CSIE, Tamkang University, New Taipei City 251, R.O.C, Taiwan

Abstract: Even if wireless networks and mobile computing technologies have comprehensively developed in recent years. Letting people extract information anywhere at anytime is the goal of this development. But in the telematics domain, the drivers can obtain road information through radio or certain in-car equipment, there is still a wide gap with regard to the synchronization of this information with actual road conditions. In the absence of adequate information, drivers often react to conditions with behaviors that do not contribute to their own driving goals but rather cause more complicated traffic conditions. Therefore, this study employs a process known as multi-agent collaboration. Information exchanged between the features and established mutual communication and collaboration mechanisms is applied in Intelligent Transportation Systems (ITS). By allowing drivers to have distributed communication with other vehicles to share driving information, collect information and/or submit their own reasoned driving advice to other drivers, many traffic situations could effectively be improved and the efficiency of the computing processes could be improved through distributed communication. This study proposes an architecture design for middleware that includes vehicle information, navigation, announcements and communication which could prove to be a more convenient and efficient intelligent transportation system.

Key words: Multi-agent system, intelligent transportation system, on-board unit, roadside unit, access point

INTRODUCTION

With the significant developments in wireless networks and mobile computing technology in recent years, capturing the mobile sources and locations of data has become a very important information service. For this reason, we use multi-agents (Al-Sakran, 2006; Dastani and Gomez-Sanz, 2005; De Meo *et al*., 2007; Davidsson and Wernstedt, 2002; Horling *et al*., 2006; Kong *et al*., 2006; Lesser, 1999; Lo *et al*., 2009; Ottens *et al*., 2009; Rahimi *et al*., 2006; Tweedale *et al*., 2007; Zambonelli *et al*., 2003) communication, coordination mechanisms (Jennings *et al*., 1998; Liu and Yao, 2004; Zhu *et al*., 2010) and existing mobile agent technology to integrate and improve the system. We develop a set of basic (i.e., access point) operations for modular information systems and query their environment and mobile agent data to design the message transmission mechanism. In the multi-agent communication and coordination mechanism, we propose four different interactive modes. Based on these modes, we create transport mechanism messages that can primarily follow the movement of mobile agent locations, enabling the prompt and accurate sending and receiving of information. The message template includes space for a mobile agent. The message transmission mechanism consists of three main components: (1) central server (the control server) (2) access point (regional based station) and (3) mobile data sources (sources). The main components of the mechanism involves Single Agent Systems (SAS) and Multi-Agent Systems (MAS) (Ellis and Wainer, 1994; Ilarri *et al*., 2008; Kotz and Gray, 1999; Lo *et al*., 2006; Shah *et al*., 2009; Yau *et al*., 2003), as shown in Table 1. The related mechanism is designed to execute tasks in the background through an application that drives an event trigger.

According to studies on cognitive access Points, it is possible for mobile communication systems to set up a wireless base station. The message transmission (Deugo, 1999) principle is also designed to allow for adjustments to the application module settings which is called access architecture in this study. The mobile

Corresponding Author: Steven K.C. Lo, Department of Information Management, Jin-Wen University of Science and Technology, No. 99, Anzhong Rd., Xindian Distirict, New Taipei City 231, R.O.C., Taiwan
Tel: +886-2-8212-2000/6345 Fax: +886-2-8212-2285/6345

Table 1: Agent system classification

	Single-Agent System (SAS)		Multi-Agent System (MAS)	
Class	Local	Network	Central and distributed	Fixed and migrating
Application	Personal assistants meeting schedulers	Personal assistants smart mailboxes	Distributed problem solving	Telecommunication

agent (Baek and Yeom, 2006; Hart *et al*., 2001) message on Point transport mechanism is informal in its application and principles. The mobile agent applies this transmission mechanism in the wireless network which has the advantages of reducing servo host loading, increasing the quality of network-connectivity, effectively managing information and facilitating easy cross-platform communication. It also provides the message object with mobility, autonomy and scalability. In the future, we hope that through this action mechanism, an agent can be integrated into highly portable mobile devices (O'Hare *et al*., 2006; Ozen *et al*., 2004), such as notebook computers, Personal Digital Assistants (PDA) and mobile phones. Last, the agent's mobility and autonomy can be enhanced through this mechanism. Thus, no matter where we go, we can access the most immediate and accurate information.

In addition, the study uses a defined interaction communication mode with an improved message transmission mechanism for freeway information. In the absence of adequate information, drivers often react to conditions with behaviors that do not contribute to their own driving goals but rather cause more complicated traffic conditions. For example, drivers compete to overtake each other in traffic but after a while find that their vehicles are still surrounded by other vehicles, showing that overtaking does not contribute to effective forward motion. In a traffic jam (for example, if the speed of the vehicle in front is too slow or there has been a car accident), it would be helpful for drivers to know whether they should change from the freeway to another road or wait patiently because the traffic jam may be resolved quickly. Therefore, we use a multi-agent collaboration and message transmission mechanism so that drivers can distribute communication with other vehicles, share their driving information, collect information and/or submit their own reasoned driving advice to the drivers. In addition, we propose a hybrid architecture that integrates OBU and RU to assist and apply in the fields of intelligent transportation systems and VANET (Bernardos *et al*., 2007; Cencioni and Pietro, 2008; Hu *et al*., 2007; Isaac *et al*., 2008; Javanmard and Ashtiani, 2009; Kane *et al*., 2008; Lee *et al*., 2008; Santa *et al*., 2008; Zhou *et al*., 2009).

ESTABLISHED MULTI-AGENTS COLLABORATION MECHANISM

This section mainly focuses on the features of multi-agent interactions. We define the existence patterns of their interactions. When an agent wants to communicate with other agents, it cannot go beyond the scope of its basic mode of communication. In general, we can discuss many different types of communication among agents, such as one-to-one, one-to-many or many-to-many interactions. We first explore the simple one-to-one model. This study develops an agent announcement state transition diagram which represents the fact that an agent wants to communicate or interact with another agent. If an agent wants to access information from others or wants to share its own information, it must declare this intention. The announcement state transition is shown in the Fig. 1.

In Fig. 1, there are five possible states: announcing, suspending, mutating, critical and accepting. These states represent the possible states of the announcement process if one agent wants to communicate with another agent. We now explain the significance of each state.

Announcing: Announcing is an initial state of an Agent. This state indicates that this agent can communicate with other agents and pass messages.

Suspending: When the agent completes its task or is no longer able to coordinate, it can return from the suspending state to the initial announcing state to wait. Mutating: After the implementation state of the agent, if there is any other declaration of implementation, we can return the state of the agent back to announcing through a mutating state and re-coordinate.

Critical: The agent enters this critical state through a sequential process of coordination and implementation.

Accepting: When the agent can only accept messages from other agents, then it enters this state.

A collaborative model of agents: In general, animals interact with each other in several different interactive modes. For instance, they cooperate with each other to survive, behave selfishly or compete. The interactive behavior could be the result of sharing a living space with others or competing with others for benefits and/or achievement of a common goal. In the multi-agent interaction model, there is a similar notion of synergy. The purpose of this study is to identify the possible forms of interactive behavior between agents, namely, cooperation, altruism, selfishness and

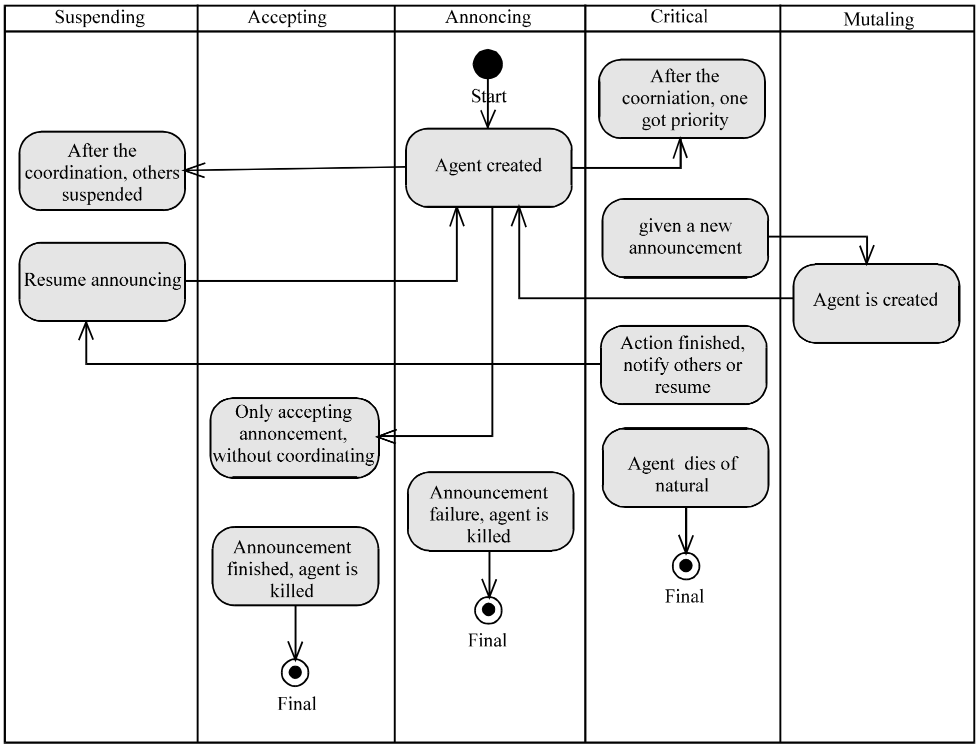

Fig. 1: Agent announcement state transition

Algorithm 1: Agent announcement state transition algorithm

```
Aunounce ( )
{if Destination Agent = free then
    if Check (Announce) = true then
    Reply (accept);
    else
    if Check (Announce) = not to conform to logic or not complete then
    Reply (repair);
    else
    Reply (error);
else
    Wait ( )
    end if;
    end if
    end if }
    Wait ( )
{if n <= times then n++;
    Reply (wait);
    Aunounce ( );
    else
    Reply (again);
end if }
```

competition. Figure 2 shows the interactive behaviors of the source agent communicates with the destination agent.

Cooperation: The communication and coordination between the two agents leads to win-win results. Such interaction is called cooperation. The cooperation mode of interaction between agents is shown in Fig. 3. Based on the definition of cooperation, the source agent can specify a complex task or make a request for other agents to complete. This model can effectively reduce execution time and obtain efficient outcomes.

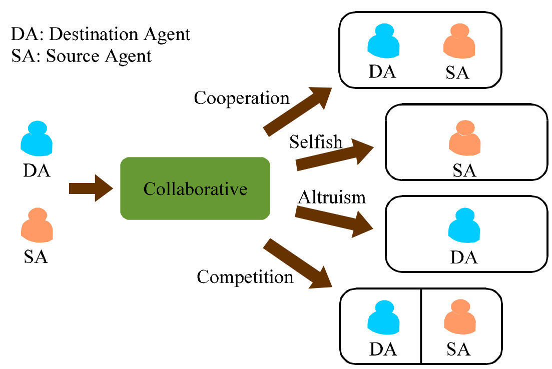

Fig. 2: Multi-agent communication and collaboration

Description:

- Source agent A strategically decides to send a demand (or request) to the Destination agent B and C and hopes to receive their responses
- Destination agent B strategically decides to send a demand (or request) to the Source agent A and Destination agent D and hopes to receive their responses

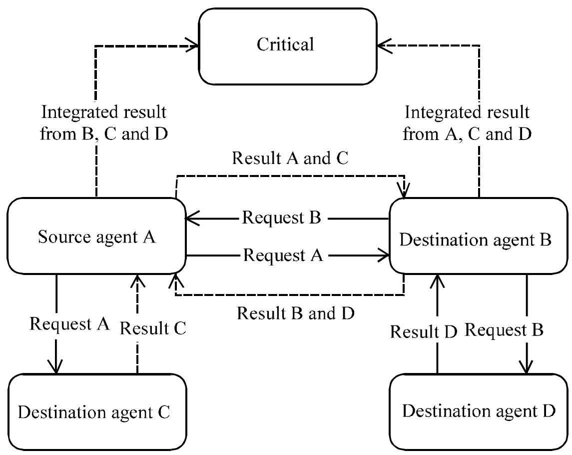

Fig. 3: Cooperative behavior model

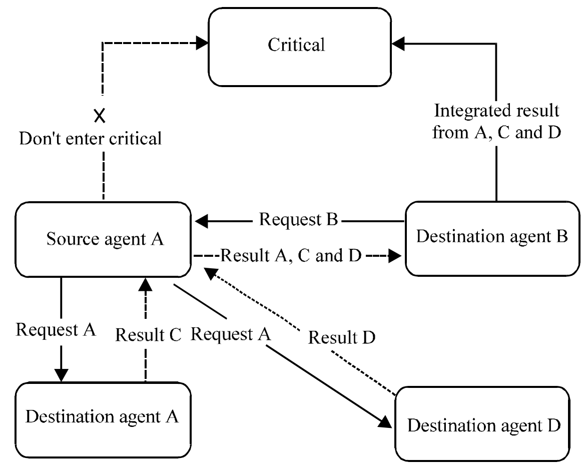

Fig. 4: Altruism behavior model

Algorithm 2: Cooperative behavior

```
SAA=new Agent; //Source Agent A
DAB=new Agent; //Destination Agent B
Result DACFunction (Agent Source A)
{DAC=new Agent;//Destination Agent C
rst=DAC.ExecuteTask (Source A);
Destroy (DAC);
Return (rst);}
Result SAAFunction (agent source B)
{ rst=SAA.ExecuteTask (Source B);
Return (rst);}
while (IsTaskFlag ==OK)
{   ResultC = DACFunction (SAA);
ResultA = SAAFunction (DAB);
SAA.Restult = ResultC+ResultA;          }
Destroy (SAA);
```

- When Destination agent B and C complete the task requested by Source agent A, Destination agent B and C send their responses to Source agent A
- Similarly, when Source agent A and Destination agent D complete the task requested by Destination agent B, Source agent A and Destination agent D send their responses to Destination agent B
- As Source agent A and Destination agent B respond to each other's demands, they share the processing results with each other. Thus, Source agent A and Destination agent B can achieve their co-operative goal together

Altruism: When two agents start communicating and coordinating, the Source Agent (SA) offers beneficial behavior to the Destination Agent (DA) but the Destination agent is not asked to do anything in return; the behavior of the Source agent is only beneficial to others. In this study, we call such communication Altruism. The altruism Interaction mode and the corresponding procedure between agents can be expressed as in Fig. 4. The Source agent completes the related tasks and passes them to a particular agent or provides relevant information to help another agent successfully accomplish its objectives. Therefore, this particular agent can achieve his objectives through the altruism mechanism.

Algorithm 3: Altruism behavior

```
SAA = new Agent; //Source Agent A
DAB = new Agent; //Destination Agent B
Result DACFunction (Agent Source)
{DAC = new Agent;//Destination Agent C
rst = DAC.ExecuteTask (Source );
Destroy (DAC);
Return (rst);}
Result DADFunction (Agent Source)
{DAD = new Agent;//Destination Agent D
rst = DAD.ExecuteTask (Source);
Destroy (DAD);
Return (rst);}
Result SAAFunction (Agent Source)
{ rst = SAA.ExecuteTask (Source);
Return (rst);}
while (IsTaskFlag = = OK)
{   ResultC = DACFunction (SAA);
ResultD = DADFunction (SAA);
ResultA = SAAFunction (DAB);
SAA.Restult = ResultC+ResultD+ResultA;          }
Destroy (SAA);
```

Description:

- SA-A strategically decides to make a request to DA-C and to obtain its result
- DA-B strategically decides to make a request to SA-A and DA-D and to obtain its result
- When SA-A and DA D fulfill the request made by Agent B, SA-A and DA-D send their processing responses to DA-B

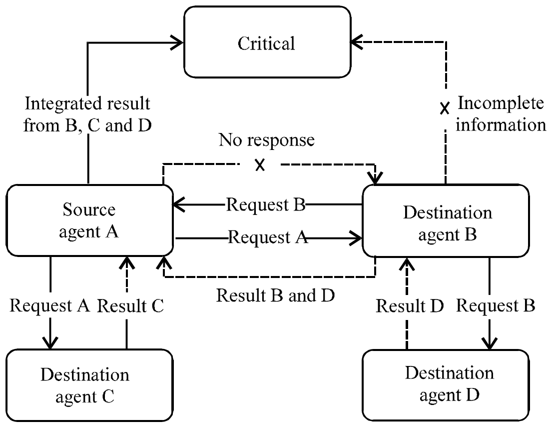

Fig. 5: Selfish behavior model

Algorithm 4: Selfish behavior

```
SAA = new Agent; //Source Agent A
Result DABFunction (Agent Source)
{DAB = new Agent; //Destination Agent B
rst = DAB.ExecuteTask (Source);
Destroy (DAB);
Return (rst);}
Result DACFunction (Agent Source)
{ DAC = new Agent; //Destination Agent C
rst = DAC.ExecuteTask (Source);
Destroy (DAC);
Return (rst);}
While (IsTaskFlag ==OK)
{  ResultB = DABFunction (DAB);
ResultC = DACFunction (DAC);
If (ResultB and ResultC)
OK;
else
FAIL;}
Destroy (SAA);
Destroy (SAB);
```

- As SA-A responds to the request made by DA-B, it will simultaneously share the processing results from DA-C such that DA-B benefits from the altruistic behavior of SA-A

Selfishness: When the two agents communicate and coordinate with each other, the Source agent may take selfish actions against the Destination agent. Such communication is labeled selfish. Figure 5 displays an agent showing the Selfish mode of communication. The given agent and other agents establish an initial stage of interaction. We assume that only a specific agent takes advantage of others and refuses to share its own resources and expertise with others.

Description:

- SA-A strategically decides to make a request to DA-B and DA-C and hopes to receive their processing results in return
- DA-A strategically decides to make a request to SA-A and DA-D and hopes to receive their processing results in return
- When DA-B and DA-C fulfill the request of SA-A, DA-B and DA-C send their processing responses to SA-A.
- DA-D completes the request from DA-B and sends its results back to DA-B but SA-A refuses to send the processing results from DA-C and its own resources to DA-B
- SA-A obtains resources from DA-C, obtains processing results from DA-D through DA-B and also gathers its own resources to achieve its goal on time. However, DA-B is unable to obtain comprehensive information from SA-A; DA-B will not achieve its purpose in communication because of the selfish behavior of SA-A

Competition: When two agents communicate and coordinate with each other such that the action of the Source agent is not helpful to the destination agent and the Destination agent does the same in return, then the outcome is not conducive to either side; in this study, we define such communication as competition.

Figure 6 shows the interactions among agents in the competitive mode of communicating. The agent begins an initial stage with other agents. At this moment, the Source agent takes the advantage of a specific agent and does not share its own resources to further the best interests of others.

Description:

- SA-A decides to send a request to DA-B, DA-C and DA-D. Furthermore, it also hopes to receive their results
- DA-B decides to send a request to SA-A, DA-C and DA-D. Furthermore, it also hopes to receive their results
- When DA-C and DA-D fulfill the request from SA-A, they (that is, DA-C and DA-D) send their processing results to SA-A. However, DA-B does not necessarily send its resources to the SA-A, depending on its own needs and competitive strategy
- When DA-C and DA-D fulfill the request from DA-B, they (that is, DA-C and DA-D) send their processing results to DA-B. However, SA-A does not necessarily send its resources to DA-B, depending on its own needs and competitive strategy
- SA-A and DA-B will respond partly, fully or not at all, depending on their mutual needs, to achieve their competitive goals

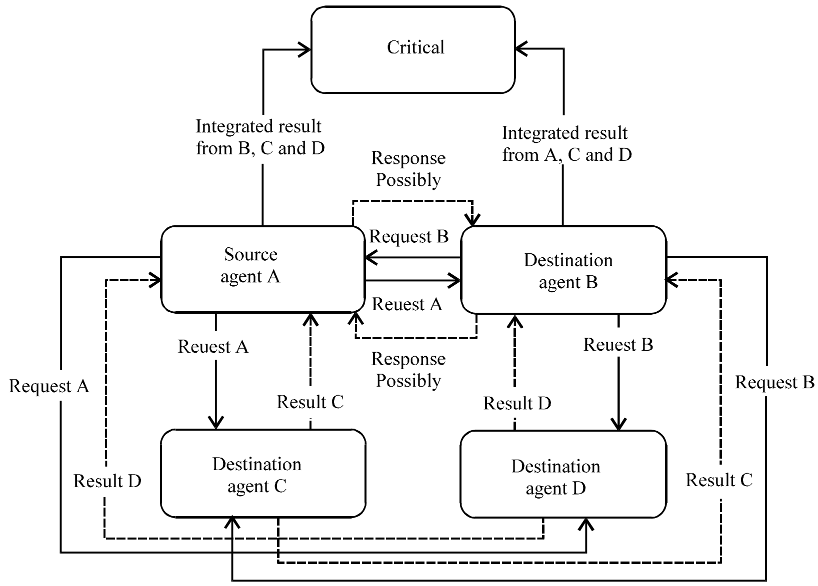

Fig. 6: Competitive behavior model

Algorithm 5: Competitive behavior

```
SAA = new Agent; //Source Agent A
Result DABFunction (Agent Source)
{DAB = new Agent;//Destination Agent B
if (Source.goal != DAB.goal)
{ rst = DAB.ExecuteTask (Source);}
else
{ Break;}
Destroy (DAB);
Return (rst);}
Result SAAFunction (Agent Source)
{if (Source.goal != SAA.goal)
{ rst = SAA.ExecuteTask (Source);}
else
{ Break;}
Return (rst);}
While (IsTaskFlag ==OK)
{ ResultA=SAAFunction (DAB);
ResultB=DABFunction (SAA);                    }
Destroy (SAA);
```

- SA-A and DA-B have been given resources from DA-C and DA-D but they lack detailed information about each other, causing them to engage in a series of competitive behaviors to achieve their goals if these goals cannot be integrated, both of them might be affected. One agent might grab or seek additional resources to be successful

IMPROVEMENT OF THE MULTI-AGENTS MESSAGE TRANSMISSION MECHANISM

Following the assessments of Dwight Deugo regarding message delivery mechanisms, it across five kinds of agents aimed at shortening the distance between the messaging and sending time, more reliable messages are provided as a guarantee to reduce the use of resources and construction costs. We present an improvement on the multi-agent message transmission mechanism.

Access point

Mobile data source transmission mechanism: As shown in Fig. 7, at each node, data are generated and changed by the source based on the agent's movement which is called the Mobile Data Source (hereafter, MDS). Each MDS message is sent directly to the information jurisdiction area (or access point). If the agent is moving, the MDS changes the original data but we also need to update new information back to the access point which refreshes the storage and publishes the sources.

When the access point obtains information from the MDS, it retransmits it to the central server for comprehensive integration. The central server receives the incoming information from different parts of the access points and makes changes to the storage and then it broadcasts the updated information back to all access points. However, the information that is sent back is not the simple data that local access point needs but the most comprehensive information that could be requested. Therefore, the different parts of the access point must seek relevant information within their respective storage areas and release information to enable mobile agents to quickly access the data that they need.

Regional-based multi-agent message transmission mechanism-district mode: Through the access point MDS transmission mechanism described above, we can see the role of every access point and its accompanying features. We now explore these activities from the perspective of the mobile agents, specifically how they obtain the desired information at any given time and use it as a reference for their next step.

As shown in Fig. 8, when the Mobile Agent (hereafter, MA) performs activities at each node, it is necessary to obtain valid reference information in the shortest possible time to prepare for the next step. Sometimes, the MA needs only certain region-specific information; then the MA needs only to send an inquiry signal through the access point in its precinct. Once the access point receives the MDS data, it will quickly transfer messages to the central server. After receiving all of the data from different access points, the central server will analyze and integrate the data and return updated, comprehensive data to different parts of the access point. The different parts of the access point will pick up the information from each area and place it in storage and then it can send an immediate reply to the MA.

If the MA moves to the next node, the data transmission from the access point will not be interrupted due to the movement of the MA. Because the transmitted information from the access point is continuous, as long as the receiving side of the MA is not closed, the transmission data will continue to come in until it reaches the jurisdiction of the next access point. At that moment, the original data will stop the transmission; thereafter, the MA will receive the information from the newly-located access point. The MA not only can obtain data from a single access point but they also can obtain very comprehensive information because all received MDS information will be passed to the central server for integration and release. If the MA does not want to know the information within a specific region but only wants to know the consistency of information among regions, then the MA can send an inquiry signal directly to the central server. When the central server receives all messages sent from each access point, it will integrate real-time, updated information and place it in storage so that it can reply immediately to the MA with the results.

As shown in Fig. 9, when ma (1) wants to move to another area, originally it received the transmitted information from access point (1) but when ma (1) reaches the district from access point (2), then access point (1) will stop its data transmission. Thereafter, ma (1) receives transmitted information from access point (2). When ma (2) is moving, it is expected to be able to predict region-wide data as a reference for the next step. Ma (2) can send an inquiry directly to the central server so that the central server can integrate all incoming data from each access point. Therefore, the central server requires only

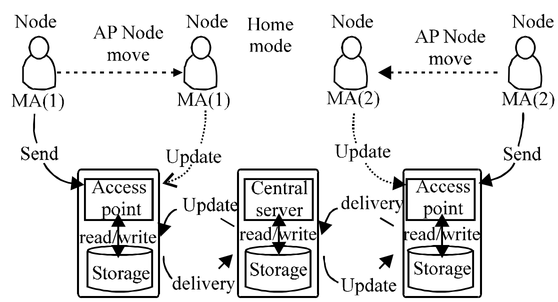

Fig. 7: Access point action of sources delivery mechanism

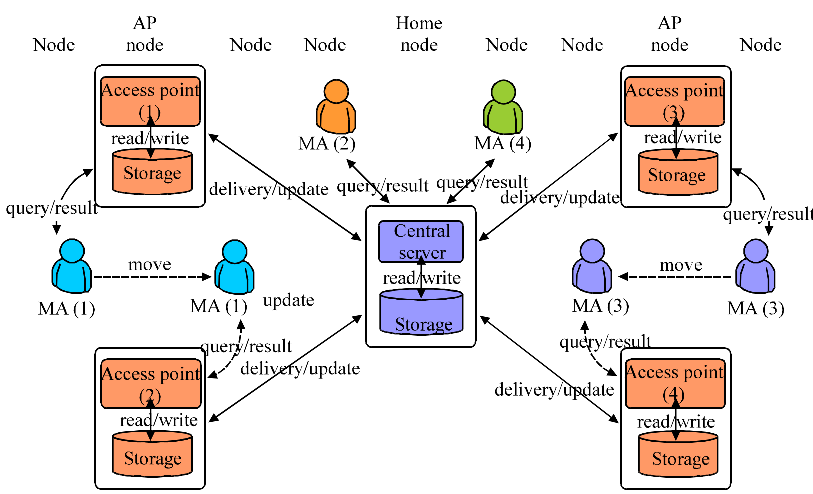

Fig. 8: Message transmission mechanism in district mode: the overall transmission infrastructure

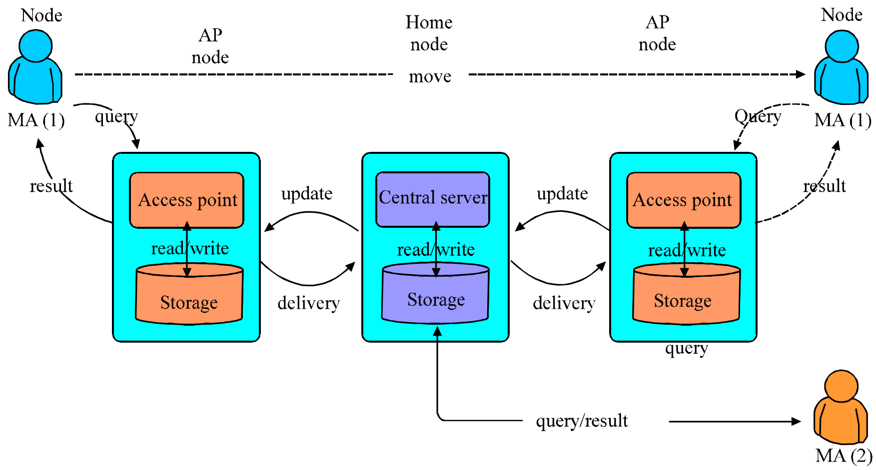

Fig. 9: Message transmission mechanism in district mode: An agent delivery infrastructure

real-time and comprehensive data to place in the storage; it could immediately reply to MA (2) with the requested data.

Wide-area-based multi-agent message transmission mechanism-domain mode: The regionally-based access point message transmission mechanism is mainly applicable to the regional wireless communications infrastructure. MAs not only can obtain information within the specific region through the access point but also can apply directly to the central server to obtain the all information set. This message transmission process is considered to pose too high a risk of overloading the central server. If none of the MAs go through access points to access the information, then all of them will access the data through the central server, resulting in a slow rate of data transfer, incomplete information and even network failure. Therefore, to make the message transmission from access point to mobile agent more efficient and to support extensive use, it needs to be upgraded cross-regionally and be able to function either synchronously or asynchronously. Most MAs are able to receive the most immediate and most accurate information by providing their host with information to move more efficiently in traffic and facilitating more secure data transfer. Therefore, based on present analysis and research, we propose a messaging mechanism based on the advanced access point to mobile agent pathway to achieve the desired goals.

As shown in Fig. 10, after the access point has received the message from MDs, it passes the information to the central server. The central server gathers all data from each access point, completes integration and analysis and then sends the comprehensive updated data back to each access point. There is one major difference from district mode: each access point no longer only places the data from the given region into storage but also places it in conjunction with the comprehensive updated information in storage.

When the MA moves to another node, it can even estimate pre-treatment reference values based on the current node through the comprehensive information it receives from the access point. Of course, to avoid the ma directly grabbing information from the central server and to reduce the host loading traffic problem, all data are broadcast into the storage of each access point. This splits the data flow across every access point and allows them receive the data cooperatively. The MA is thus able to grasp more complete sources more efficiently in real time.

As shown in Fig. 11, when MA (1) wants to move to another location, it can read data that have been updated, integrated and transferred from the central server and plan pre-emptive action. There are several mobile agents (i.e., MA (2) to (x)) within the framework path from access point (1) to (2). The comprehensive information requested by the MA which is provided by the central server, is deposited in the storage areas of access point (1) and (2). All MAs within the scope of this route can be assigned to access point (1) and (2) to receive data. In this way, it is possible to share the loading flow and control the entire network's traffic on each of the servo hosts. However, MA (2) through MA (x) are in the path range between access point (1) and (2). The information on this part of the path is provided exclusively from access point (1); thus, when MA (2) through MA (x) obtain a message from this route, it must be fully managed by access point (1).

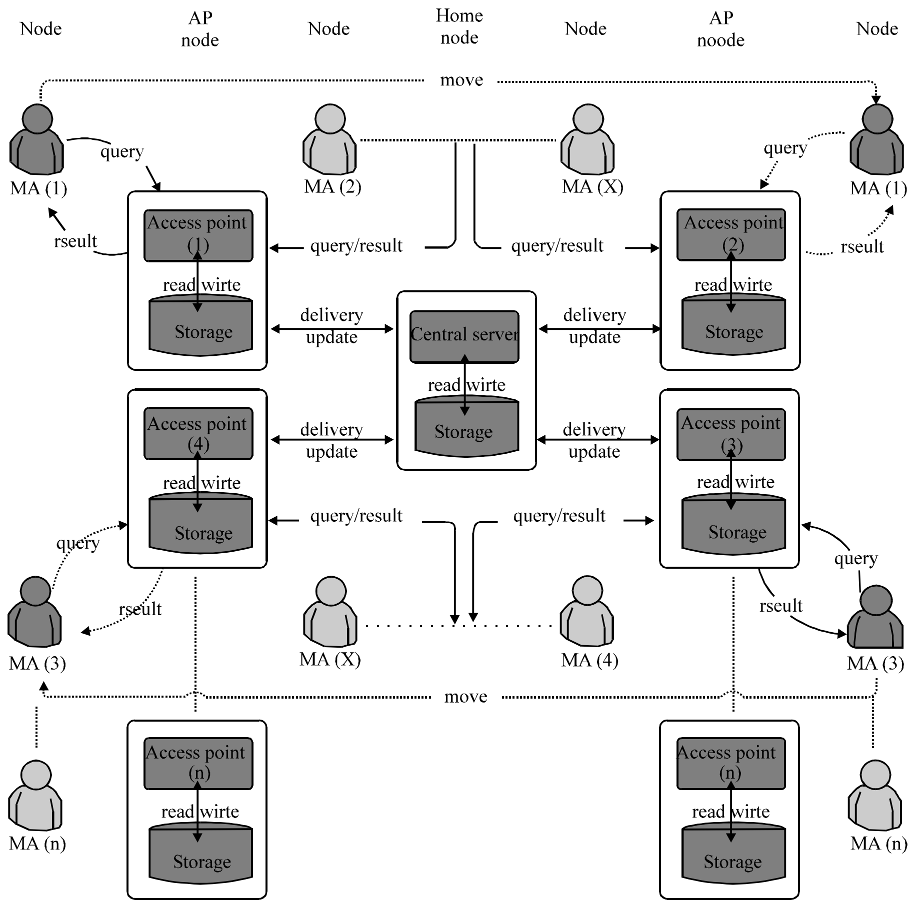

Fig. 10: Message transmission mechanism in domain mode: the overall transmission infrastructure

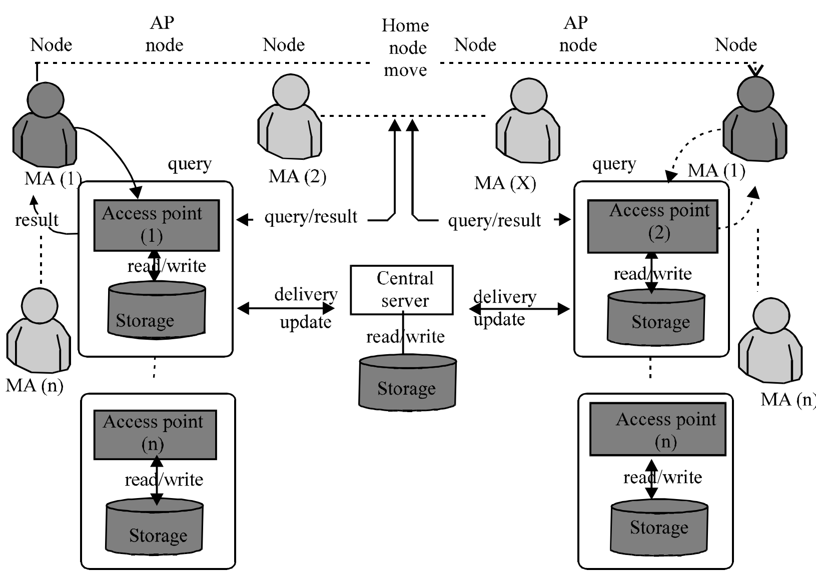

Fig. 11: Message transmission mechanism in domain mode: an agent delivery infrastructure

THE MIDDLEWARE ARCHITECTURE BETWEEN VEHICLE AND INFRASTRUCTURE

We now draw on our definitions of the multi-agent coordination mechanism and message transmission models (that is, district mode and domain mode, respectively) discussed above. We propose a transportation middleware architecture between the application layer (i.e., the vehicle) and the network layer (i.e., the freeway infrastructure). In Fig. 12, the application layer mainly represents the operation system on OBU. It shows the requests from the user and the results from the inference engine. The network layer represents related devices on RU, such as an access point or base station. It is composed of four parts, namely, the vehicle information manager, navigation manager, announcement manager and communication manager. We define them as follows.

Vehicle information manager: A vehicle Information agent can manage all information on a vehicle, including speed, destination, functions and so on. Moreover, it can record these data into a database and communicate with other managers.

Navigation manager: A vehicle positioning agent can process the dynamic position of vehicle. Then the Navigation Information agent can integrate the information and store it into the knowledge base. It also can provide them with data from a temporal/spatial inference engine to provide driving suggestions.

Announcement manager: This manager is responsible for communicating with other vehicles. It can send its own information to others for reference and it can also receive messages or information from other vehicles. When it receives information from others, it allows the navigation manager to evaluate it. So, this manager would play an important role in interacting with other vehicles. Moreover, when it wishes to announce information to other vehicle, it can follow the agent announcement state transition mechanism to communicate.

Communication manager: the communication manager is in charge of communication between other managers and network elements. When it wants to send a message to communicate, it can follow the district mode or domain mode. The central server information manager is equivalent to the core system management module. It is responsible for its own area access point and for processing data. In addition, the information manager is responsible for communicating internal messages with the mobile agent manager of the access point, including various query types. The administrator needs to send the best mobile agent based on the best data access strategy to process the task and to deposit the response into the database of the access point. Finally, if the system's data management strategy supports the updating of cache

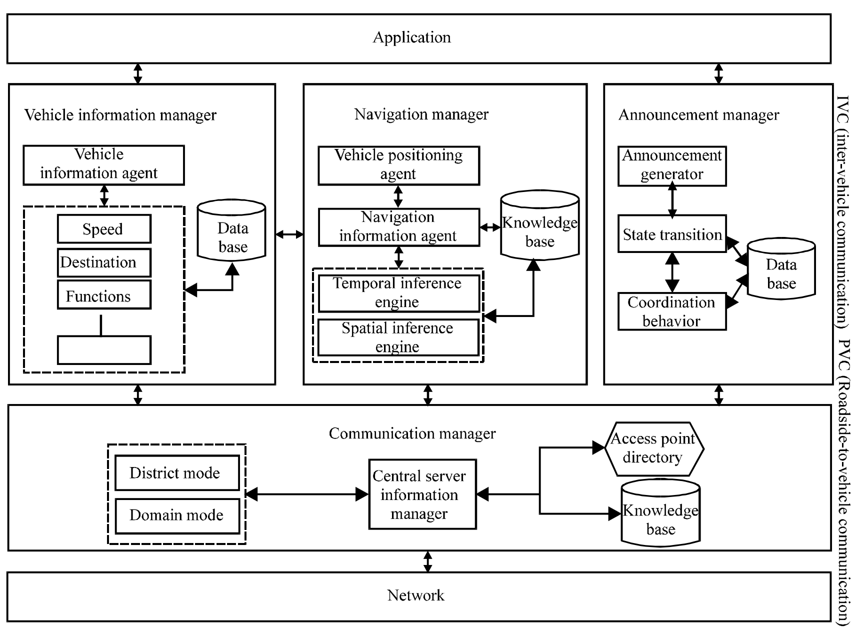

Fig. 12: Overview of middleware architecture

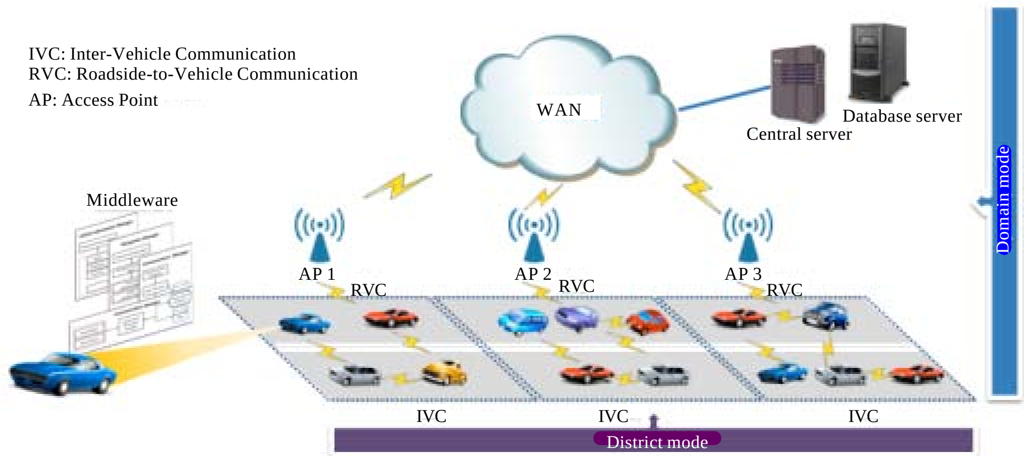

Fig. 13: Integration of message transmission and middleware

Data, then the information manager will be responsible for the maintenance and management of the cache on the access point. In addition, the knowledge base is a repository with considerable capacity for data storage. The information manager is responsible for data management. If the data do not support the updated cache data management strategy and the long-distance data are changed, then the copy in the database should be deleted. However, when the cache data management strategy does support the updates, the information manager is responsible for maintaining the consistency of the database information.

As shown in Fig. 12 and 13, based on these communication behaviors (cooperation, altruism, selfish and competition) each agent's information will be interactively shared and transmitted. This part of the study will be defined as the inter-vehicle communication (ivc) and it may facilitate regional management through the access point of each region. Access points for each region may be integrated through the central server with access points in other regions and thus share their useful regional data with other regions. We can use the message transmission mechanisms described in this study. We define as Roadside-to-Vehicle Communication (RVC).

In Fig. 13, access point manages the defined cell area and all mobile data sources and stores the data from each operation in the directory. Via the wireless communication media for all mobile terminals, such as access points or base stations, it provides the wireless data for mobile terminals. From the client (such as obu) it must be accessed through the access point for data processing and communication transmission.

The central server information manager is the core system management module. It is responsible for its own area access point and for processing data. It is responsible for communicating the internal messages with the mobile agent of the access point which includes query types. The administrator needs to send the best mobile agent based on the best data access strategy to process the task and to deposit the response into the database of the access point. Finally, if the system's data management strategy supports the updating of cache data, then the central server will be responsible for the maintenance and management of the cache on the access point. Whether cars interact each other in every dotted line or they interact with access point and central server, they must follow the message transmission mechanism under the district mode and domain mode.

CONCLUSION

Based on the importance of current mobile information access needs, this study proposes multi-agent interaction involving collaborative models and a messaging communication mechanism which can be generated using a basic structure. This structure meets the requirements of wireless-based location information services through location-related data retrieval. This study presents a system framework and various types of dynamic data management systems to access data. The framework is more efficient than the traditional basic strategy and also less costly. We use a traffic information application on the freeway as an example of which vehicles can coordinate with each other and take advantage of regional and wide-area wireless environments to achieve the most effective dispersion of computation costs.

This study proposed the message transmission mechanism to aim at shortening the distance between the messaging and sending time, more reliable messages are provided as a guarantee to reduce the use of resources and construction costs and these mechanisms and framework could be used as the V2V, the vehicle to AP and AP to the central server communication and collaboration. Moreover, each AP will send the data back to the central server to reduce loading and latency effectively. This is also an expandable architecture and very novel solution.

REFERENCES

Al-Sakran, H., 2006. An agent-based architecture for developing e-learning system. Inform. Technol. J., 5: 121-127.

Baek, J.W. and H.Y. Yeom, 2006. A timed mobile agent planning approach for distributed information retrieval in dynamic network environments. Informat. Sci., 176: 3347-3378.

Bernardos, C.J., I. Soto, M. Calderon, F. Boavida and A. Azcorra, 2007. VARON: Vehicular Ad hoc route optimisation for NEMO. Comput. Communicat., 30: 1765-1784.

Cencioni, P. and R.D. Pietro, 2008. A mechanism to enforce privacy in vehicle-to-infrastructure communication. Comput. Communicat., 31: 2790-2802.

Dastani, M. and J.J. Gomez-Sanz, 2005. Programming multi-agent systems. Knowledge Eng. Rev., 20: 151-164.

Davidsson, P. and F. Wernstedt, 2002. A multi-agent system architecture for coordination of just-in-time production and distribution. Proceedings of the ACM Symposium on Applied Computing, (ASAC'02), ACM Press, pp: 294-299.

De Meo, P., A. Garro, G. Terracina and D. Ursino, 2007. Personalizing learning programs with X-Learn, an XML-based, user-device adaptive multi-agent system. Informat. Sci., 177: 1729-1770.

Deugo, D., 1999. Mobile agent messaging models. School of Computer Science, Carleton University.

Ellis, C. and J. Wainer, 1994. A conceptual model of groupware. Proceedings of the ACM Conference on Computer Supported Cooperative Work, (ACCSCW'94), USA., pp: 79-88.

Hart, D., M. Tudoreanu and E. Kraemer, 2001. Mobile agents for monitoring distributed systems. Proceedings of the 5th International Conference on Autonomous agents, (ICAA'01), ACM Press, pp: 232-233.

Horling, B., V. Lesser, R. Vincent and T. Wagner, 2006. The soft real-time agent control architecture. Autonomous agents Multi-Agent Syst., 12: 35-92.

Hu, F., A. Rughoonundon and R. Krishnaram, 2007. Using emulation instead of simulation to analyze synchronized higher layers performance in wireless networks. Inform. Technol. J., 6: 941-953.

Ilarri, S., E. Mena and A. Illarramendi, 2008. Using cooperative mobile agents to monitor distributed and dynamic environments. Infor. Sci., 178: 2105-2127.

Isaac, J.T., J.S. Camara, S. Zeadally and J.T. Marquez, 2008. A secure vehicle-to-roadside communication payment protocol in vehicular ad hoc networks. Comput. Commun., 31: 2478-2484.

Javanmard, A. and F. Ashtiani, 2009. Analytical evaluation of average delay and maximum stable throughput along a typical two-way street for vehicular ad hoc networks in sparse situations. Comput. Commun., 32: 1768-1780.

Jennings, N.R., K. Sycara and M. Wooldridge, 1998. Roadmap to agent research and development. Autonomous agents Multi-Agent Syst., 1: 7-38.

Kane, L., B. Verma and S. Jain, 2008. Vehicle tracking in public transport domain and associated spatio-temporal query processing. Comput. Commun., 31: 2862-2869.

Kong, S.H., S.D. Noh and Y.G. Han, 2006. An agent-based collaborative assembly process planning system. Int. J. Adv. Manuf. Technol., 28: 176-183.

Kotz, D. and R.S. Gray, 1999. Mobile agents and the future of the Internet. ACM Operat. Syst. Rev., 33: 7-13.

Lee, W.H., S.S. Tseng and C.H. Wang, 2008. Design and implementation of electronic toll collection system based on vehicle positioning system techniques. Comput. Commun., 31: 2925-2933.

Lesser, V.R., 1999. Cooperative multiagent systems: A personal view of the state of the art. IEEE Trans. Knowledge Data Eng., 11: 133-142.

Liu, J. and C. Yao, 2004. Rational competition and cooperation in ubiquitous agent communities. Knowledge Based Syst., 17: 189-200.

Lo, S.K.C., J.C. Lin, H.C. Keh, S.J. Wu and Y.H. Lin, 2006. The intelligent agent of the billing service between the PSTN and VOIP. Inform. Technol. J., 5: 51-56.

Lo, S.K.C., H.C. Keh, Y.H. Lin and W. Jo-Chi, 2009. Integrated the intelligent agent behavior model and billing service into communication system. Inform. Technol. J., 8: 668-677.

O'Hare, G.M.P., M.J. O'Grady, C. Muldoon and J.F. Bradley, 2006. Embedded agents: A paradigm for mobile services. Int. J. Web Grid Serv., 2: 379-405.

Ottens, K., N. Hernandez, M. Gleizes and N. Aussenac-Gilles, 2009. A multi-agent system for dynamic ontologies. J. Logic Comput., 19: 831-858.

Ozen, B., O. Kilic, M. Altinel and A. Dogac, 2004. Highly personalized information delivery to mobile clients. Wireless Netw., 10: 665-683.

Rahimi, S., J. Bjursell, M. Paprzycki, M. Cobb and D. Ali, 2006. Performance evaluation of SDIAGENT, a multi-agent system for distributed fuzzy geospatial data conflation. Inform. Sci., 176: 1175-1189.

Santa, J., A.F. Gomez-Skarmeta and M. Sanchez-Artigas, 2008. Architecture and evaluation of a unified V2V and V2I communication system based on cellular networks. Comput. Commun., 31: 2850-2861.

Shah, N., R. Iqbal, A. James and K. Iqbal, 2009. Exception representation and management in open multi-agent systems. Inform. Sci., 179: 2525-2561.

Tweedale, J., N. Ichalkaranje, C. Sioutis, B. Jarvis, A. Consoli and G. Phillips-Wren, 2007. Innovations in multi-agent systems. J. Network Comput. Appl., 30: 1089-1115.

Yau, S.M.T., H.V. Leong and A. Si, 2003. Distributed agent environment: Application and performance. Inform. Sci., 154: 5-21.

Zambonelli, F., N.R. Jennings and M. Wooldridge, 2003. Developing multiagent systems: The Gaia methodology. ACM Trans. Softw. Eng. Methodol., 12: 317-370.

Zhou, L., G. Cui, H. Liu, D. Luo and Z. Wu, 2009. NPPB: A broadcast scheme in dense VANETs. Inform. Technol. J., 9: 247-256.

Zhu, C.Y., Y.Y. Du and P. Li, 2010. A routing-based dynamic workflow. Inform. Technol. J., 9: 561-568.