A framework for knowledge sharing between autonomous agents

WILLIAM J. TEAHAN[1] and PETER G. TUFF
School of Computer Science, University of Wales, Bangor, U.K.

Abstract: Presently, the sharing of knowledge amongst agents is a topic that has yet to be explicitly formalized or have a framework provided for it. Many agent-based applications either employ a basic form of knowledge sharing, or employ a more complicated form, such as the communication by word-of-mouth. The purpose of this paper is to provide an agent-based formalization of knowledge sharing. Such a unique solution includes the design and implementation of an agent-based Application Programming Interface (API) capable of allowing different forms of knowledge sharing between agents. In addition, a GUI is provided that interfaces with the agent-based API and provides visualization of agent simulations. The functionality of the API, and accompanying GUI, is demonstrated by the implementation of an agent-based traffic simulation. Moreover, the results of the agent-based traffic simulation are used to evaluate the effectiveness of alternative knowledge sharing methods.

Key-Words: - Autonomous agents, Knowledge sharing, Word-of-mouth, Blackboard architecture.

1 Introduction

Knowledge can be defined as all information needed by a human being, or machine, to complete a task considered as being complex [3, p.232]. That is, knowledge concerning a specific topic is organised and retained by human or machine that can be utilised when decided on a task to complete. According to Ferber [*ibid.*] such knowledge can be divided into two categories: *knowing something* and *knowing how to do something*. *Knowing something* concerns the knowledge and understanding of objects and phenomena encountered. While the *knowing how to do something* relates to the analysis of the relationship between different phenomena [*ibid.*]. Knowledge relating to how to do something allows a human or machine to select an appropriate action given the current state of the world, and to anticipate the effect of the action on the state of the world.

Clifton and Teahan [2] employ a similar definition of knowledge when defining what they call *knowledgeable* agents. Under this definition, knowledge must be associated with an agent; knowledge cannot exist on its own. An agent has knowledge if it knows how to use information, consisting of data, to answer a question. If we consider selecting an appropriate action as the question, then the agent utilises its knowledge to provide and answer in the form of an appropriate action. Therefore, we can describe knowledge as information that enables the agent to select an action that is appropriate for the current state of the world and the adopted goal of the agent.

An agent is equipped with knowledge of an environment and knowledge of how its actions affect the environment. This inherent knowledge is recognised as the agent's knowledge base [5, p.195]. Arguably, the agent's knowledge is not limited to simply determining what action is appropriate. The agent's knowledge also incorporates information gathered by perceiving its environment and interactions with other agents [3, p.235].

1.1 Agent Interaction and Knowledge Sharing

Agents can share their knowledge by co-operating with other agents. The ability of agents to interact with the environment and with other agents, and share their knowledge, allows agents to cooperate to solve problems. Such collaboration promotes more effective problem solving. That is, sharing knowledge between agents increases the knowledge base of agents. Given that the quality of actions and reasoning of agents is determined by the accuracy and quality of the agent's knowledge, the greater its knowledge, the more effective the agent.

Two methods for agents to share knowledge are described below: *word-of-mouth* and *blackboard*.

Word of mouth: The word of mouth method of communication between agents replicates the word of mouth communication between humans. Human communication is divided into two categories, including mass and personal communication [4, p.296]. Mass media is an example of mass communication. For example, newspapers and television provide information for many people, in a one-to-many style. Conversely, personal communication is bi-directional in a one-to-one style. Therefore, information exchanged in one-to-one style and peer-to-peer fashion is recognised as word-of-mouth communication.

Word of mouth communication is employed by the WOM Agent System [4, p.296]. The agents are used by a search engine to retrieve relevant Web documents. Each agent contains a database, which stores a portion of available Web documents. When the user submits a query, a search request is sent to an available agent. The agent searches its database for relevant Web documents, if no relevant Web documents are located then the search request is passed to other agents in a word of mouth fashion. This process continues until an agent does locate relevant Web documents. Upon locating relevant Web documents, the information is returned via the route the search query was sent. Each agent along the route copies and stores the reply data. Thus, popular Web documents are propagated amongst agents. As the system propagates popular Web documents between agents, maintenance is reduced as unpopular Web documents are not distributed and are ignored. Furthermore, network load and traffic is reduced, as the most accessible agent, in terms of the user's network environment and location, is selected to perform the initial search [4, p.296].

Knowledge sharing between agents is not limited to direct interactions between individual agents. Knowledge can be distributed via agent accessible repositories. The blackboard method of knowledge sharing method is an example of an indirect form of knowledge sharing.

Blackboard: The use of a blackboard was originally developed in the context of artificial intelligence [3, p.128]. The blackboard model consists of problem solving modules, known as knowledge sources, which operate independently and do not communicate information with one and other directly [*ibid*.]. The knowledge sources exchange information indirectly using a shared base or blackboard, with the blackboard acting as a central repository of information. The content of the blackboard consists of facts and hypotheses generated during the process of problem solving.

This blackboard model has been adapted for use with search agents in Peer-to-Peer networks [1]. The agents append knowledge and retrieve the knowledge from blackboards stored at nodes in the network, thus promoting the indirect sharing of knowledge between different agents.

Therefore, knowledge sharing allows the individual agents to cooperate and exchange knowledge from their knowledge base. The ability to distribute knowledge between agents is beneficial, as it allows agents to gather knowledge about areas of the environment they have not explored. As the effectiveness of an agent, in terms of selecting the most appropriate actions, is determined by the amount and quality of knowledge it has, the sharing of knowledge promotes a more effective agent. Consequently, agents that collaborate and share knowledge should discover solutions to problems more effectively than a single sophisticated agent working in isolation.

However, agent-based knowledge sharing is a topic that has yet to be explicitly formalised or a framework provided for it. Many agent applications are implicitly employing a form of knowledge sharing either in a very rudimentary form (e.g. akin to ants chemically marking a good path, allowing other ants to locate and follow the same path), or in a more complex form of knowledge sharing, such as that occurs in the blackboard and word-of-mouth methods. Therefore, the authors believe that there is a growing need for a more concrete formalization in this area. The rest of this paper is concerned with the designing, implementing and the applying of a framework for knowledge sharing.

2 Design of an agent-based framework for Knowledge Sharing

The purpose of this section is to outline the design for a framework for knowledge sharing between autonomous agents. The authors propose an Application Programming Interface (API) that allows the user to create a virtual environment that simulates a real world environment, which is populated with agents capable of sharing knowledge related to a particular task. The knowledge being shared amongst agents is gathered by agents as they perceive and travel around the environment.

The following sections identify the design goals and provide an overview of the API. A detailed description of every class and method incorporated into the design of the API and a Java implementation can be found using the following URL:

www.informatics.bangor.ac.uk/~wjt/AIIA/ABKS.

2.1 Design Goals

Prior to the execution of an agent-based simulation, an environment is required to host the agents. The proposed virtual environment consists of a series of nodes, interconnected by links. The use of nodes and links allows the API to simulate a number of real-world networks, such as road traffic and computer networks. The ability to simulate a number of real-world environments broadens the appeal of this API. In addition, the API allows the user to recreate real-world networks with the same topology, thus, allowing spatially similar networks to be reproduced and simulated.

After the creation of the virtual environment, the API allows agents to be inserted into them. The goal of the agent is to locate its designated destination node from its present location. As the agent travels around the environment, it perceives and gathers information about the environment. Information gathered through observations is retained and can be distributed amongst other agents. The knowledge can be distributed in a word-of-mouth or blackboard fashion, for example. This allows alternative knowledge sharing methods to be compared in terms of the time taken by the agent to find its destination node and distance covered. The agent is able to use its knowledge to locate a route from its present location to its destination. In the absence of knowledge of the network, the agent will still endeavour to reach its destination node.

As the API is required to simulate a network, it also takes into account how traffic congestion affects the ability of the agent to traverse the network. Likewise, as the agent can observe the traffic delays, it will endeavour to avoid congested areas of the network.

The design goals of the Agent-based framework for Knowledge Sharing are identified as follows:

- To allow the user to randomly or manually produce a virtual environment consisting of nodes interconnected with links. Such an environment may simulate a real-world environment such as a computer network.
- To provide an ability to save the virtual environment to file, and reload the file for future use in other simulations.
- To allow agents, capable of knowledge sharing, to be added to the environment in a random or manual fashion. Agents endeavour to reach a randomly or manually assigned destination node.
- To simulate traffic congestion. That is, as the amount of agents traversing a section of the environment increases, the time taken to traverse the section also increases.
- To allow the user to specify whether the agents employ knowledge sharing in a word-of-mouth or blackboard fashion, or do not employ knowledge sharing.
- To allow agents to use their knowledge to find a route from their present node to their destination node. The agent avoids traffic by selecting the quickest route.
- To provide the ability to specify the “instincts” of the agent in the absence of knowledge of the environment. That is, determine how the agent selects a route when it has no knowledge of a route.
- To generate statistical information concerning its journey from the start node to the destination node. The statistical information is written to file.
- To provide a means of extending the API to include future knowledge sharing methods.
- To allow a GUI to interface with the API, providing a visualisation of an agent-based knowledge sharing simulation.

2.2 Design Overview

The Agent-based Knowledge Sharing API (ABKS) consists of eight classes, including: Agent, Element, Environment, Filestore, Knowledge-Sharing, Link, Node and Simulation class. The following table provides a brief summary of the classes used by the API.

API Class Summary

Class Name	Description
Agent	*The Agent class represents the agent entity and records the agent’s journey. In addition, it records the agent’s location and knowledge of the environment and determines the agent’s actions.*
Element	*The Element Interface is used by the Link and Node classes to ensure they share common functionality.*
Environment	*The Environment class stores instances of the Link and Node objects. The Environment class allows the user to create a network consisting of Nodes and Links.*
Filestore	*The Filestore class is used to store a network, consisting of Nodes and Links, to file. Furthermore, it allows previously stored networks to be restored and to be used in other simulations.*
Knowledge-Sharing	*The KnowledgeSharing class determines how knowledge is to be exchanged between individual agents. The class can be extended to include additional knowledge sharing methods.*
Link	*The Link class represents a link between two Node objects.*
Node	*The Node class represent a point in the network where links converge.*
Simulation	*The Simulation class governs the agents situated in the environment. That is, it requests that individual agents periodically update themselves.*

The design of the API is divided into the functionality concerning the construction of the virtual environment and the functionality concerning the execution of the agent-based knowledge sharing simulation. Prior to the execution of a simulation, the user must create a virtual environment. The virtual environment consists of Node and Link objects, which are instances of the Node and Link classes. The Node object represents a point in the environment; the Link objects reference and connect Nodes.

The use of Node and Link objects allows the API to create networks that can represent real-world networks. For example, a network can be created to embody a road traffic system. Therefore, the Nodes represent towns and the Links represents road connections between towns. The creation of Node and Link objects is controlled by the Environment class. The Environment class allows the operator to manually create Node and Link objects, and specify their location within the environment. Consequently, the operator may produce a network that recreates the topology of real-world networks. Alternatively, the Environment class allows the operator to randomly generate Node and Link objects, with their location also being randomly generated. The Filestore class may be called by the Environment class to save the network, consisting of Node and Link objects, to a file. In addition, the Filestore class may be called to restore a previously saved network.

Following the creation of the virtual environment, the functionality representing the simulation aspect of the API is invoked. The Simulation class is used to add Agents to the network. The agents are represented by instances of the Agent class. The Simulation class allows the operator to add agents manually to individual Node objects and specify a destination Node. Conversely, the agents can be randomly assigned to Node objects, accompanied by a randomly assigned destination Node. After the agents have been added to the environment, the Simulation class is used to govern the simulation. During the simulation, the class calculates the current agent traffic levels at different parts of the network. Traffic information and knowledge of the network or environment is passed to the agent. The Agent class, representing a single agent, uses this information to select appropriate actions based on its perception of the environment, and updates its internal status accordingly. For example, the Simulation class informs the agent that it can perceive Node linked to its present Node. The agent subsequently uses this information and decides to travel to the linked Node.

Furthermore, during a simulation the Simulation class regularly invokes the KnowledgeSharing class. The KnowledgeSharing class determines whether knowledge of the environment, such as traffic and topological information, is exchanged between agents. The KnowledgeSharing class represents alternative knowledge sharing methods employed by the simulation, such as the word-of-mouth or the blackboard method.

The goal of the agent is to reach the destination Node in the shortest amount of time steps or iterations of the simulation. After the agent reaches its destination Node, the Filestore class is invoked. The Filestore class allows statistics regarding the performance of the agent to be written to file.

3 Experimental Results

The purpose of this section is to apply the API, in conjunction with the GUI, to produce agent-based knowledge sharing simulations. The different simulations are used to compare the effectiveness of different knowledge sharing methods. The results of the simulations are compared and discussed.

The Agent-based knowledge sharing API provides the functionality to simulate knowledge sharing between agents. Such knowledge sharing can be performed in a word-of-mouth fashion whereby agents that meet, as they traverse the network, exchange knowledge. Similarly, the API incorporates a blackboard knowledge sharing method based on the Blackboard Resource Discovery Mechanism (Al-Dmour and Teahan 2004, p.1). The blackboard method does not permit direct knowledge sharing amongst agents; alternatively, agents distribute knowledge via repositories or blackboards situated at Nodes. Any agent traversing a Node exchanges knowledge with the blackboard. Blackboards have not been applied to knowledge sharing amongst agents. Therefore, the purpose of the simulations outlined below is to evaluate the effectiveness of the blackboard method in comparison with the word-of-mouth method.

In addition, the API allows a simulation to be performed without any knowledge being shared amongst agents. Consequently, the agents move around the network in a random fashion, until they reach their designated destination Node. The authors believe that the agents' ability to find their destination Node is improved when knowledge is distributed amongst agents. Therefore, simulations employing a knowledge sharing method should result in more effective agents than those simulations devoid of knowledge sharing.

The goal of an agent is to reach its designated destination Node. Upon reaching the destination Node, the agent generates statistical information concerning its journey. The statistical information includes the amount of Nodes traversed, distance travelled and journey time. This information is subsequently used to calculate the average Nodes traversed, distance travelled and journey time for each agent. In addition, the percentage of agents successfully reaching the designated destination Node can also be calculated. This information is used to evaluate and compare the effectiveness of the word-of-mouth and blackboard knowledge sharing methods, along with simulations devoid of knowledge sharing.

The following section outlines and justifies the structure of the simulations that were performed.

3.1 Outline of Simulation

All of the simulations outlined below were performed with no knowledge sharing, and with the word-of-mouth and blackboard knowledge sharing methods.

3.1.1 Simulation 1: Random 100 Nodes and 100 Agents.

The first simulation consists of network of 100 randomly created Nodes. The Nodes are randomly linked, with no links allowed to cross. Figure 1 represents a screenshot of the network to be used in the simulation. A hundred agents are added to the network. This network is large and complex, with a large number of agents. The simulation runs for 10000 time steps or iterations.

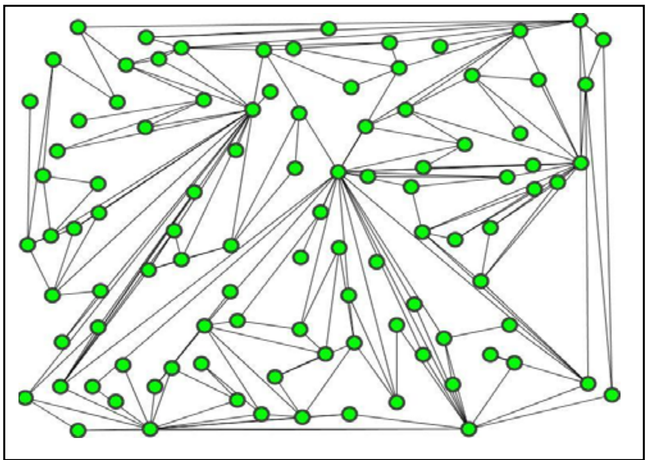

Figure 1.Network with 100 randomly generated Nodes.

2.1.1 Simulation 2: Random 100 Nodes and 50 Agents.

The second simulation uses the same network employed during the first simulation. (See Figure 1.) Therefore, the network consists of 100 randomly created Nodes, which are randomly linked, with no links allowed to cross. Simulation 2 differs to Simulation 1 as only 50 agents are to be added to the network. Again, the simulation runs for 10000 time steps or iterations.

3.1.2 Simulation 3: Random 50 Nodes and 100 Agents.

The third simulation employs a network consisting of 50 randomly generated Nodes. Figure 2 represents a screenshot of the network to be used in the simulation. The Nodes are randomly linked, with no links allowed to cross. In addition, 100 agents are added to the network. The network employed in this simulation is simpler than the network used in Simulation 1 and 2. That is, there are fewer Nodes and Links, but the same amount of agents exist. The simulation is to run for 10000 time steps or iterations.

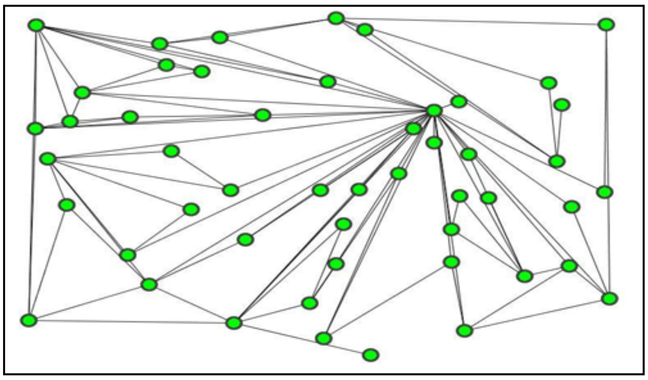

Figure 2. Network with 50 randomly generated Nodes.

3.1.3 Simulation 4: Random 20 Nodes and 100 Agents.

The fourth simulation employs a network consisting of 20 randomly generated Nodes. The Nodes are randomly linked, and the Links are allowed to cross. Allowing Links to cross produces a complicated network consisting of many Links. Figure 3 represents a screenshot of the network to be used in the simulation. In addition, 100 agents are to be added to the network. The simulation runs for 10000 time steps or iterations.

3.1.4 Simulation 5: Road Traffic Network.

In the final simulation, the network has been manually created. The network structure is loosely based on the motorway road network of the England and Wales. Nodes represent major towns and cities, whilst the Links represent the motorway connections between the towns and cities. (See Figure 4.) A hundred agents added to the network and the simulation runs for 10000 time steps or iterations. Each agent represents a car journey; the driver has no knowledge of the road network prior to starting the journey and gathers knowledge by observing the network and through exchanges with other agents or blackboards.

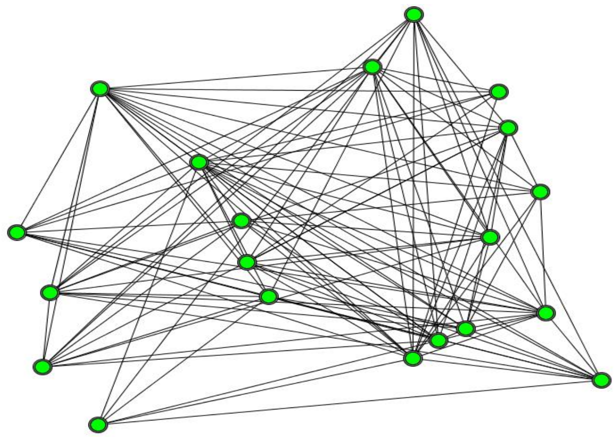

Figure 3. Network with 20 randomly generated Nodes, with crossings of links.

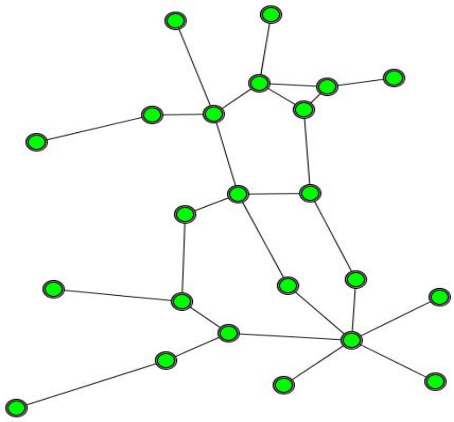

Figure 4. Network based on England and Wales Motorway Structure.

The purpose of Simulation 1 is to provide definitive statistical analysis of the performance of the different knowledge sharing methods. That is, the simulation employs a large complicated network accompanied by a large amount of agents, thus providing a large sample set. Simulation 2 is identical to Simulation 1, apart from having only half the number of agents. The purpose of Simulation 2 is to show whether a smaller number of agents, resulting in less knowledge being distributed, affects the performance of the agents. Conversely, Simulation 3 reduces the number of Nodes but still employs 100 agents. As the blackboard method employs a single blackboard at every Node, reducing the number of Nodes to 50 halves the amount of blackboards used in the simulation. Simulation 3 demonstrates whether the reduction of Nodes has an adverse affect on the blackboard and word-of-mouth methods. Simulation 4 reduces the amount of Nodes to 20 and provides an alternative Link structure. The network has significantly more Links connected to individual Nodes, compared with the first three simulations. Lastly, Simulation 5 employs a simple, manually created, network that represents the motorway structure of England and Wales. This network is significantly simpler than previous networks as it employs a small number of Nodes and Links.

3.2 Simulation Results

The table corresponding to each of the simulation results displays the percentage of the assigned agents that find their designated destination Node. In addition, the tables show the average amount of Nodes traversed, distance covered and journey time of the agents as they travel from their starting Node to their destination Node.

	No Knowledge Sharing	Word-of-Mouth	Blackboard
% of destinations found	*21%*	*89%*	*94%*
Ave. # of nodes traversed	*23.2*	*18.4*	*15.9*
Distance Covered	*3799.6*	*2918.9*	*2392.6*
Time Steps	*4542.3*	*3507.7*	*2848.9*

Table 1. Simulation 1 results: Random 100 nodes and 100 agents.

The results in Table 1 for Simulation 1 show that the blackboard knowledge sharing method was the most successful in terms of agents reaching their destination Node, with 94% of agents reaching their destination. In addition, the blackboard simulation's agents on average traversed fewer Nodes, covered less distance and experienced a shorter journey time than the other methods.

The results in Table 2 for Simulation 2 show that the blackboard knowledge sharing method has the greatest success rate of agents reaching their allotted destination Node. Furthermore, the blackboard method outperforms the other methods in terms of average traversal of Nodes, distance covered and journey time. Compared to the results of Simulation 1, all three methods show a decrease in the percentage of agents locating their destination. In addition, the methods experience an increase in the average traversal of Nodes, distance covered and journey time compared with Simulation 1.

	No Knowledge Dist.	Word-of-Mouth	Blackboard
% of dest-inations found	*15%*	*88%*	*90%*
Ave. # of nodes traversed	*22.1*	*21.77*	*21.0*
Distance Covered	*3499.6*	*3428.8*	*3222.1*
Time Steps	*4056.6*	*4001.3*	*3715.0*

Table 2. Simulation 2 results: Random 100 nodes and 50 agents.

	No Knowledge Dist.	Word-of-Mouth	Blackboard
% of dest-inations found	*41%*	*92%*	*94%*
Ave. # of nodes traversed	*16.2*	*9.3*	*8.4*
Distance Covered	*3290.0*	*1789.9*	*1649.0*
Time Steps	*4537.6*	*2205.4*	*2084.9*

Table 3. Simulation 3 results: Random 50 nodes and 100 agents.

Compared with Simulation 1 and 2, the results for Simulation 3 (see Table 3) show that the success rate, in terms of agents reaching their destination Nodes, has increased for all three methods. Ninety four percent of agents employing the blackboard method located their destination Node, which is the highest success rate out of the three methods. In addition, the blackboard method had the lowest average traversal of Nodes, distance covered and journey time.

In Simulation 4 (see Table 4), all of the agents employing the blackboard knowledge sharing method have successfully reached the designated destination. Furthermore, the Blackboard method displays the best results in terms of lowest average traversal of Nodes, distance covered and journey time.

For Simulation 5 (see Table 5), again, the blackboard method proves to be the most effective knowledge sharing method in terms of lowest average traversal of Nodes, distance covered and journey time. Interestingly, the percentage of agents reaching their designated destination Node is an equal (99%) for both the word-of-mouth and blackboard method.

	No Knowledge Dist.	Word-of-Mouth	Blackboard
% of dest-inations found	*65%*	*99%*	*100%*
Ave. # of nodes traversed	*10.9*	*4.3*	*3.6*
Distance Covered	*3546.6*	*1360.1*	*1150.5*
Time Steps	*4431.6*	*1677.6*	*1471.1*

Table 4. Simulation 4 results: Random 20 nodes and 100 agents.

	No Knowledge Dist.	Word-of-Mouth	Blackboard
% of dest-inations found	*67%*	*99%*	*99%*
Ave. # of nodes traversed	*17.3*	*9.4*	*8.1*
Distance Covered	*1391.8*	*769.2*	*638.4*
Time Steps	*3037.7*	*1593.6*	*1455.6*

Table 5. Simulation 5 results: Road traffic network.

3.3 Analysis of Results

The five simulations show that the blackboard knowledge sharing method provides the highest success rate, in terms of agents reaching their designated destination Node. The percentage of agents reaching the designated destination Node is represented by a graph in Figure 5.

Clearly, both the blackboard and word-of-mouth knowledge sharing methods are significantly more effective than the simulation employing no knowledge sharing. For four of the simulations, the blackboard method has a higher percentage of agents reaching their destination. The exception is Simulation 5, where the percentages are equal at 99%. Significantly, although the percentages are the same, the simulation employing the blackboard method has lower average traversal of Nodes, distance covered and journey time for the agents completing their journey. Therefore, the blackboard method is the superior knowledge sharing method as the average agent has covered the shortest distance to reach its destination node. In addition, the journey time is also less. Consequently, in Simulation 5 the agents employing the blackboard method reached their destination node sooner than the word-of-mouth agents did. Figure 6 shows a graph representing average journey times for the agents of each simulation.

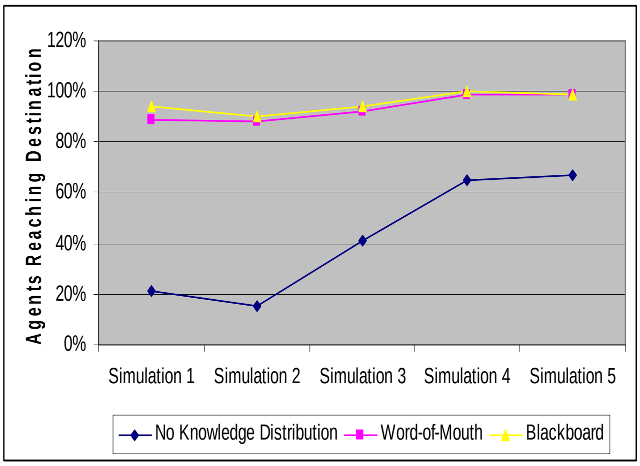

Figure 5. Percentage of agents reaching destination.

Note that the percentage of agents reaching the designated destination has generally increased for all three methods when comparing the results from Simulation 1 through to Simulation 5. This general rise in percentage can be attributed to a difference in the amount of nodes in a network, compared to the amount of agents. Figure 7 displays a modified version of the graph displayed in Figure 5. That is, the x-axis has been reorganised in decreasing node to agent ratio from left to right. For example, Simulation 1 has 100 Nodes and 100 agents, therefore, the simulation has a node-to-agent ratio of one.

The graph indicates that all three methods have experienced an increase in the percentage of agents reaching their destination Nodes as ratio of Nodes-to-agents has decreased. Note there is one exception, in Simulation 4 the simulation employing no knowledge sharing has experienced a slight reduction in percentage. This slight fluctuation can be ignored, as the difference in Node-to-agent ratio between Simulation 4 and 5 is small. More importantly, for each simulation the blackboard method is proven to be the most successful knowledge sharing method. Even when the number of nodes is reduced, resulting in fewer blackboards, and the amount of agents kept constant, the blackboard method still outperforms the word-of-mouth method. In addition, when the number of nodes is halved (comparing Simulation 1 with 3), the percentage of agents reaching their destination node remains at 94% for the blackboard method. Therefore, the blackboard method does not suffer adversely when the number of Nodes is reduced, resulting in fewer blackboards. Arguably, when there are fewer Nodes and the amount of agents remains constant, the Nodes must experience higher levels of agent traffic. Consequently, a blackboard is likely to store and share more knowledge; therefore, the performance of the method is not reduced.

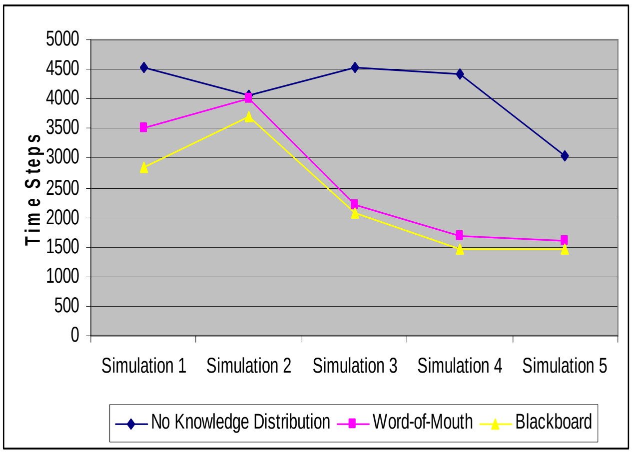

Figure 6. Average time steps to reach destination.

Whilst the blackboard method has been identified as the superior knowledge sharing method, both the word-of-mouth and blackboard share similarities in performance. Both methods are reliant on agents to transport knowledge to different areas of a given network. For example, blackboards remain static and the agents store and retrieve knowledge from the blackboards as they travel through the Node. Likewise, when employing the word-of-mouth method, agents exchange knowledge with other agents as they meet on their journey around the network. In addition, the agents perceive and gather knowledge regarding the topology and traffic levels of the network. Therefore, the greater the number of agents employed by a simulation, with the size of the network remaining the same, the greater the volume of knowledge gathered and distributed amongst agents. As the number of agents added to a network is increased, the percentage of agents locating their destination Nodes also increases. Such a rise in performance is witnessed when comparing Simulation 2 with Simulation 1. That is, the number of agents has doubled, resulting in a performance increase from 88% to 89% for the word-of-mouth method and a performance increase from 90% to 94% for the blackboard method.

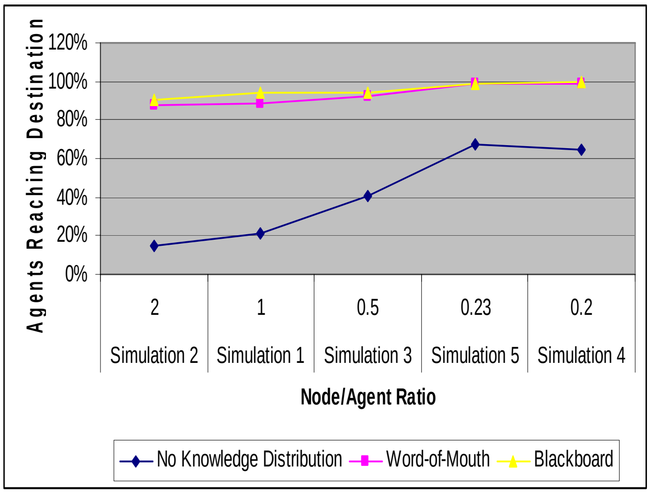

Figure 7. Percentage of agents reaching their destination sorted by node/agent ratio.

3 Discussion

A framework for agent-based knowledge sharing has been designed and implemented in the form of an API. The API possesses the functionality to create a new virtual environment or simulate an existing real-world environment. In addition, the API allows agents to be created and added to the environment. Such agents act autonomously and endeavour to reach their designated destination point or goal. The agents display intelligence as they perceive their environment and gather knowledge regarding the topology and traffic levels of the environment. The knowledge is recorded in inner state and can be distributed amongst other agents or blackboards. The agents search this inner state to locate and follow a route to their destination point.

In addition, a GUI has been developed to provide a visualisation of the agent-based knowledge sharing simulations. The API allows the agents of the simulation to employ a word-of-mouth or blackboard method of knowledge sharing, or no knowledge sharing. The ability to employ alternative knowledge sharing methods has enabled their performance to be compared. The results show that the blackboard-based architecture consistently outperforms the word-of-mouth method in traffic simulation application.

The four networks employed by the simulations provided a good range of alternative network structures to test the different knowledge sharing methods. As predicted, the performance of the word-of-mouth and blackboard methods was superior to the simulations not employing any knowledge sharing. Interestingly, both the word-of-mouth method and blackboard method display similar statistical performance, in terms of percentage of agents reaching their designated destination node and average traversal of nodes, distance covered and journey time. Arguably, this similarity in performance is a consequence of both knowledge sharing methods being dependent on the movement of agents to gather and transport knowledge. Therefore, as the number of agents assigned to the network has increased, the average performance of the individual agents has improved.

Although the results for the word-of-mouth and blackboard methods are similar, the blackboard method produced the better agent performance statistics for the five simulations. The Agent-based knowledge sharing API described in this paper has adapted the Blackboard Resource Discovery Mechanism (Al-Dmour and Teahan 2004, p.1) to create a novel knowledge sharing method based on the blackboard architecture. Further, the API has made it possible to compare the performance of the word-of-mouth method with the blackboard method. Interestingly, the statistical analysis indicates that the blackboard method is superior to the tried and tested word-of-mouth knowledge sharing method.

References:

[1] Al-Dmour, N. A. and Teahan, W. J. 2004 "*The Blackboard Resource Discovery Mechanism for P2P Networks*". In: 16th IASTED International Conference on Parallel and Distributed Computing Systems, 9-11 November 2004 Cambridge., USA.

[2] Clifton, T. and Teahan, W., 2005. "Knowing-Aboutness: Question-Answering using a logic-based framework". In: 27th European Conference on Information Retrieval (ECIR), 21-23 March 2005 Santiago de Compostela, Spain.

[3] Ferber, J., 1999. "*Multi-Agent Systems: An Introduction to Distributed Artificial Intelligence*". Harlow, England: Addison-Wesley.

[4] Ohtani, T. and Minami, T., 1998. "*The word-of-mouth agent system for finding useful Web documents*". In: Asia Pacific Web Conference, 27-30 September 1998 Beijing. Fujitsu, 295-300.

[5] Russell, S. J. and Norvig, P., 2003. "*Artificial Intelligence: A Modern Approach*". 2nd ed. New Jersey: Prentice-Hall.