Personal Intelligent Travel Assistant

A distributed approach

M.J. Beelen L.J.M. Rothkrantz

Delft University of Technology, Mekelweg 4, 2628 CD Delft, The Netherlands

Astract - *The Personal Intelligent Travel Assistant provides the user with a personalised, up-to-date advice on a handheld device, which takes into account actual delays and user preferences. The travellers themselves provide information about the delays of trains, using GPS to estimate whether a user is experiencing a delay. Furthermore a website of the Dutch Railways (NS) provides information on delays for 83 stations in the Netherlands. This information can be imported into the prototype. To find optimal routes, considering train changes and the user preferences, the DYNET algorithm of Eduard Tulp (Tulp91) was selected and implemented. By extending the static information on the train schedule with dynamic information on delays, a prototype of a dynamic route guidance system was designed and implemented, which runs on the Sharp Zaurus handheld and a distributed server backend. The system was realised by agents modelled after the Beliefs, Desires and Intentions agent model, using Jade combined with Jadex, forming a FIPA compliant distributed agent platform.*

1. Introduction

In The Netherlands one million people use public transportation every working day. There is no system, helping travellers to find optimal routes in public transportation, which includes information on delays of trains. Within the research program Seamless Multimodal Mobility of the Netherlands Research School for TRAnsport, Infrastructure and Logistics (TRAIL), research is carried out into the Personal Intelligent Travel Assistant (PITA). The PITA is an application running on a handheld device, providing communication between travellers and public transportations companies. It guides the user with an up-to-date personalised advice on what public transportation to use, based on current, available delay information. Public transportation companies can get a better insight in the traveller flows, giving them the possibility to optimise the scheduling of trains and carriages. A future application is the reservation of public transportation through the PITA system.

2. Tools

To realise the system a combination of JADE and Jadex was used as the agent platform. The Java Agent DEvelopment framework (JADE) is an Open Source implementation of the FIPA standard (see (BCPR03)). Specific features are the possibilities for distribution of an agent platform over multiple machines and a set of graphical (debugging) tools. Jadex is developed by the Distributed Systems and Information Systems Group at the University of Hamburg (see (PBL03)). It is an extension for the JADE platform which makes it possible to use the BDI (Beliefs, Desires and Intentions) model (see for instance (Wool00)) to model agents. The BDI model was first used as a philosophical model for modelling rational (human) agents and has later been introduced as a software development paradigm. The mental attitudes belief, desire and intention are used to model autonomous interacting entities that pursue their own goals and act rationally. In this case the rational behaviour of the entities means the entities try to take the best measure in order to reach their goals, seen from an ideal concept of intelligence and not with respect to human behaviour (see (RN95)).

The BDI model is represented in Jadex by beliefs, goals and plans. A BDI-agent has its own belief base, containing facts about the agents perception of the world. Furthermore, each agent has a list of goals (and sub-goals) which it wants to achieve. In order to reach these goals, the agent has a library of plans at its disposal. Triggered by events the agent decides which plans to execute to achieve the adopted goals. Plans can also change the beliefs, generate events and adopt new goals, which makes it possible for the agent to function autonomously in order to reach the goals.

3. System design

As shown in figure 1, the PITA system has two user interfaces: a management interface and a handheld interface. The management interface is used for viewing and modifying the system configuration. The handheld interface is used by travellers using a handheld device to communicate with the system. A central role is played by the distributed agent platform, which runs personal agents for each individual

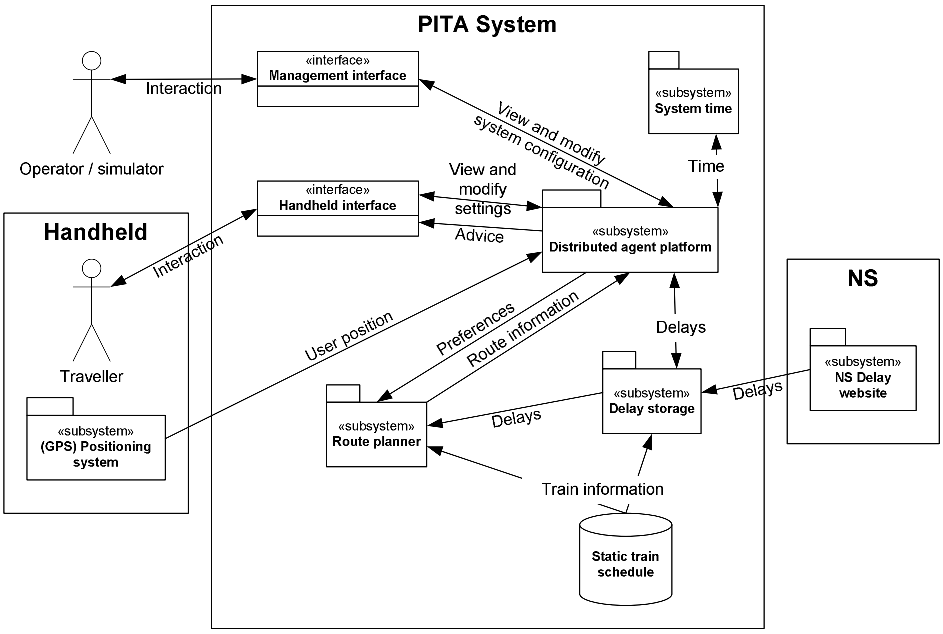

Figure 1: The PITA system design

traveller. In order to give an optimal advice to the traveller, the personal agents receive information about the best route to the destination, given the current delay situation and occurring delays during the trip. Using the user position it is possible to determine whether the user is experiencing a delay. This information about delays is also communicated back to the system itself, to actualise the stored delay information. Next to that the NS delay website ('actuele vertrektijden') is used to fill the delay system. The route planner uses the static train schedule combined with the actual delay information to search an optimal route for the traveller, based on the preferences of the individual traveller.

3.1 The handheld interface

Three different configurations for distributing the application over handhelds and a distributed server backend were considered. In the first configuration, the handheld interface, the complete application runs on the distrbuted backend and the handheld device provides the user with only an interface which directly connects to the backend. A clear disadvantage of this configuration is that whenever the connection between the handheld and the backend is lost, the user does not get an updated advice anymore and is lost. In order to solve this problem, more application functionality should be running on the handheld device.

3.2 The handheld application with distributed backend

The configuration as shown in figure 2, running the synchronisation of the delay information on a distributed backend and a personal agent on the handheld device was selected for the implementation.

As the personal agent runs on the handheld device, it also becomes possible to exchange information over an ad-hoc wireless network between individual handheld devices, without having to use a mobile internet connection. This means the personal agents on different handheld devices can exchange route information and information about delays in order to optimally get to each individual destination. Figure 2 also shows how a handheld connected only by an ad-hoc wireless connection can be connected to the system.

By sharing information over an ad-hoc network, it becomes possible to use the system without having a mobile Internet connection per device. The following information can be shared over the ad-hoc network:

- **Position information**
  If several travellers are on the same train, then only a single user needs to have a position device, like for instance a GPS receiver. By sharing this information across the wireless network, all other users are updated of their locations as well.
- **Route information**
  If travellers (partly) follow the same route to a destination, routes can be shared and it is not necessary to set up a mobile Internet connection with the distributed backend.
- **Delay information**
  By sharing information about train delays, this information needs to be communicated only once over the mobile

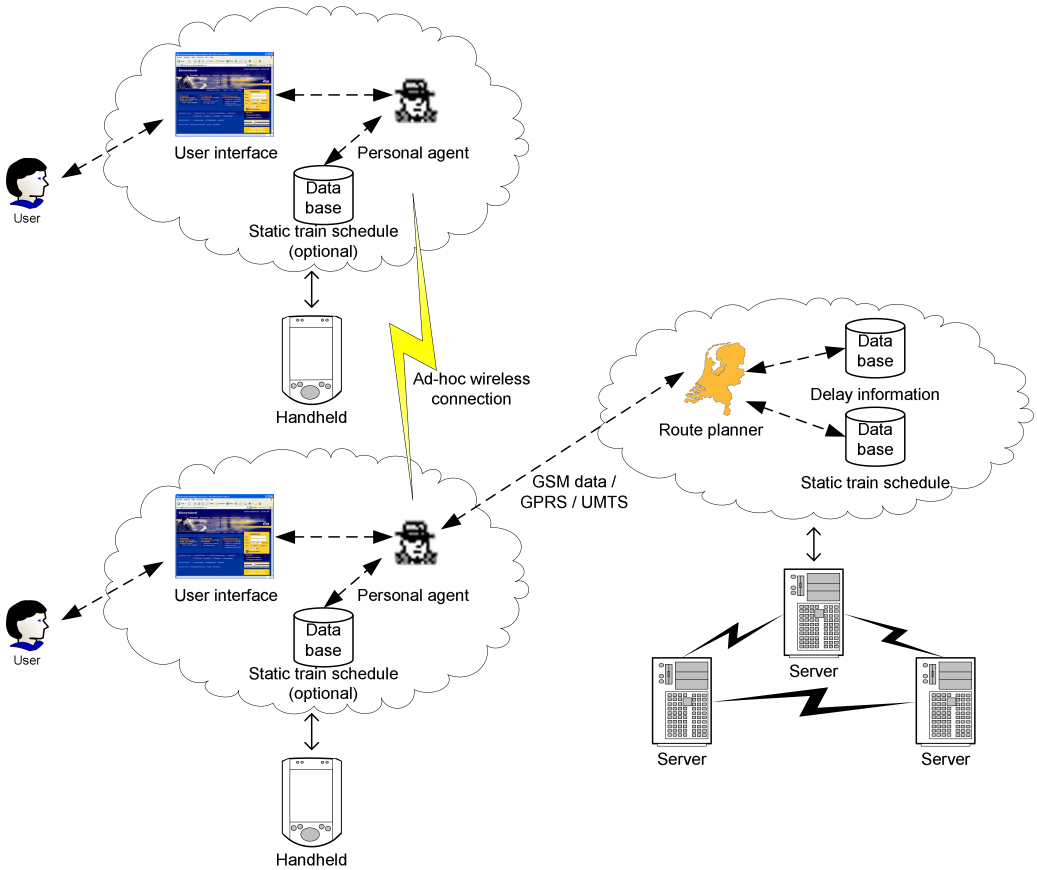

Figure 2: The distribution of the application over handhelds and a distributed backend

Internet connection and can then cost effectively be shared across the wireless ad-hoc network between different handhelds.

In order for the ad-hoc networking to function in an optimal way, it is necessary to have enough travellers which are equipped with capable devices, having built-in GPS receivers and mobile Internet connection (preferably GPRS or UMTS). This is necessary to have a user base which forms a minimal coverage for the PITA system across the stations and trains in The Netherlands. To realise such a minimal coverage, NS could equip all conductors with a capable PITA device. Another solution would be to give travellers who buy an annual railpass the option to buy such a device for reduced charges.

Another problem is that nobody wants to use his mobile Internet connection all the time to help other users, but also wants to profit from the system from time to time. A solution for this would be to register credits on each handheld device. As the mobile Internet connection of the device is used, the handheld gets credits. If there are more devices with a mobile Internet connection available in an ad-hoc network, the mobile Internet connection of the device with the least credits is used to provide the network with information.

This way the handhelds will negotiate what connection is used and which information is shared, leading to ad-hoc created networks and autonomous personal agents negotiating and sharing information.

3.3 The distributed handheld application

Another possible configuration, which was not used for the implementation of the PITA system, but shows several interesting aspects which can be of interest for further research, is shown in figure 3. The route planner is also running on the mobile device in this configuration. Besides this, the handheld device also stores all delay information locally. This means a central server synchronising the delay information is no longer needed.

In this configuration the delay and route data is generated and shared by personal agents in ad-hoc networks. Each personal agent is responsible for finding relevant data from nearby other agents. This leads to autonomous functioning personal agents, exchanging and storing information in a non-deterministic way, based on available information in the nearby ad-hoc networks. The delay data will be distributed in a way that suits the demand for

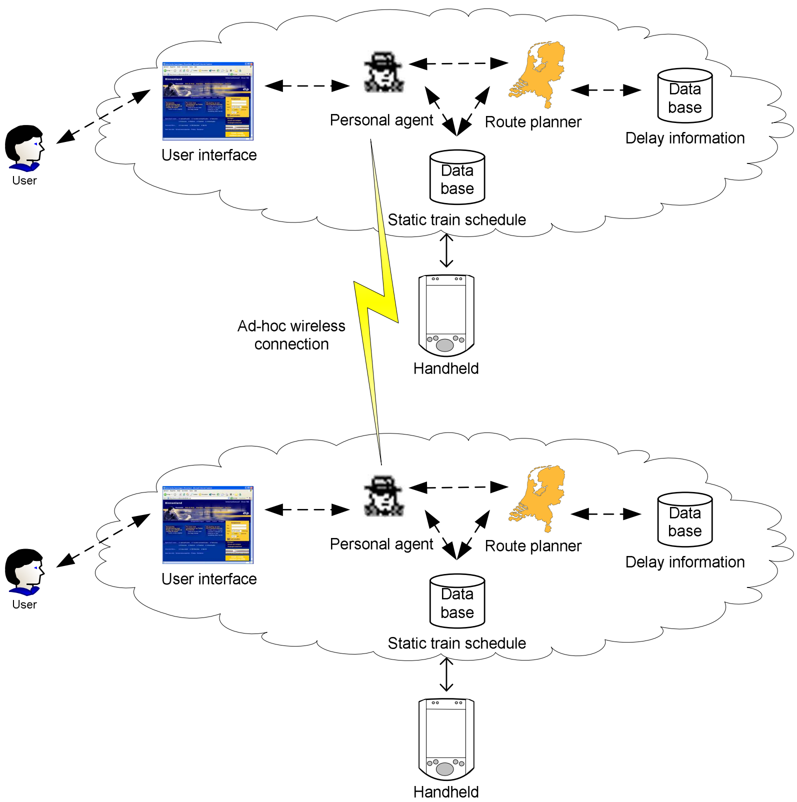

Figure 3: The distributed handheld application

information from the users of the system. Emerging from the autonomous functioning agents is a distributed application with a very interesting distribution model for the data.

4. Implementation

The prototype was implemented on a distributed, fault tolerant, configuration of a Jade platform using Main-Container replication for the server backend.

Besides this, each handheld runs its own individual Jade platform with the personal agent and its interface. These personal agents communicate with the rest of the system to generate an optimal advice for the user as shown in figure 4.

The personal agent uses the route planner to find the best route to the destination. Next to that the personal agent stores the preferences of the user and keeps track of the position of the user to detect whether the user is experiencing a delay. As soon as a delay is detected, this is updated in the delay database and a new route to the destination is determined. After updating the delay information, other personal agents which have planned to take this train are notified of the delay and can also determine a new route to their destination.

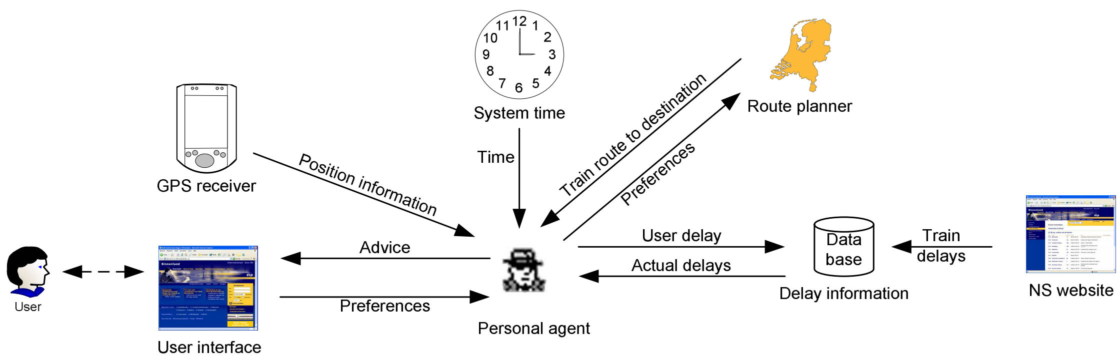

Figure 4: The personal agent, communcating with the rest of the system

For the distribution of the workload on the server backend, the servers divide the responsibility for different regions of The Netherlands. Agents currently travelling in a region will contact the server responsible for that region. The region division is also used by the synchronisation algorithm of the available delay information. Regional delays are distinguished from interregional delays. Interregional delays are synchronised across all servers while regional delays are only known on the server responsible for that region. As a traveller moves from one region to the next region, the server responsible for the other region is contacted to update the advice.

Next to this, the DYNET algorithm (see (Tulp91)) was implemented to search routes. The algorithm consists of two passes: a forward pass and a backward pass. The forward pass finds a route departing as soon as possible and reaching the destination station at the earliest possible arrival time. The backward pass optimises the route found in the forward pass by finding the route having the last possible departure time which reaches the destination station at the same time.

The DYNET algorithm considers a change_value parameter, which is the extra time a traveller is prepared to travel to avoid one train change. Furthermore the CON macro, specifying the amount of time needed for a specific train change can be implemented user dependend (i.e. giving a larger value for elderly persons). This was done in order to search routes which optimally fit the preferences of the user.

To use the available dynamic information about train delays, the implementation of the DYNET algorithm, as used in the PITA system, adds the delay of a train on a certain track e to the *end(e)* attribute of the edge *e* and to the *begin*(e_j) and *end*(e_j) attributes of all consequent edges e_j having $id(e_j) = id(e)$ and for which $begin(e_j) \geq end(e)$. The CON macro was implemented as a function, taking as parameters a minimum change time and a change time per platform that has to be crossed to reach the departure platform of the next train, as follows:

$CON(e_{j-1}, e_j) = 0$ if $id(e_{j-1}) = id(e_j)$, or
= train_change_time + (extra_change_time_per_platform x
| getarrivalplatformnumber(e_{j-1}) - getdepartureplatformnumber(e_j) |)

To simplify the parameter configuration of the search algorithm, we defined four profiles among which the user can select. The profiles depend on whether the user is currently travelling or planning to travel in some time. Travelling users use the single (first) pass algorithm while planning users use both passes. The other distinction is made on whether the user want a comfortable route or the fastest route. The change_value parameter and the needed change time (the CON function) are configured accordingly. Table 1 gives an overview of the four user profiles and the search algorithm parameters.

Table 1: the four user profiles

Location:	User type:	Parameters:
Travelling	Comfortable	first pass algorithm, change_value = 10
Travelling	Performance	first pass algorithm, change_value = 0
Planning	Comfortable	two pass algorithm, change_value = 10
Planning	Performance	two pass algorithm, change_value = 0

The prototype was implemented on a Sharp SL-C860 Zaurus handheld, which runs blackdown Java 1.3.1 on embedix Linux. In order to use speech synthesis, Festival Lite with an American male voice was installed on the device, with a wrapper developed in C, which opens a serversocket to which the text that has to be read out loud can be outputted from Java.

The prototype can be used as a real-time system, importing actual delays directly from the NS website, or as a simulation environment. In order to simulate a real system, random travellers can be started and monitored and the system time can be altered.

5. Tests

Using the simulation environment different tests were performed. First of all tests were performed on the functioning of the system. Figure 5 shows a screenshot of the route plan of a route starting in Delft at 10:15 to Groningen Noord. After introducing a interregional delay of Intercity 537 from station Zwolle, using the server management interface shown in figure 6, the route plan is updated as shown in figure 7. Figure 8 shows a visualisation of the route in which the cross represents the current position of the traveller and the line the route to the destination.

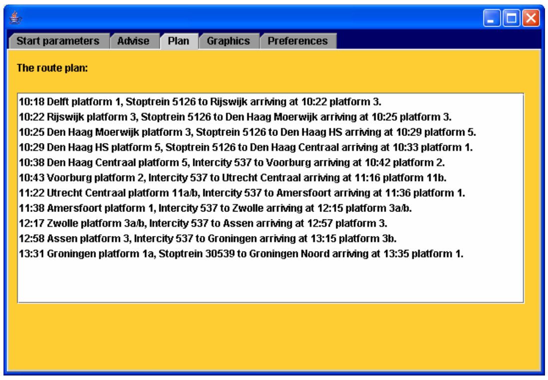

Figure 5: The route plan to Groningen Noord

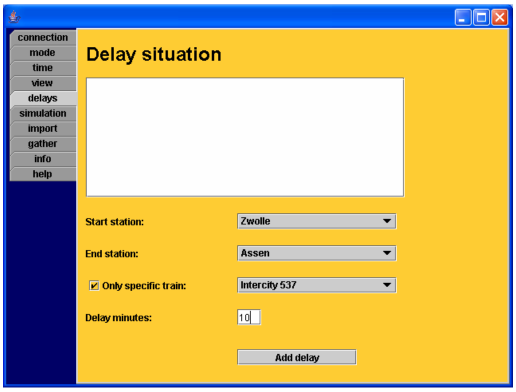

Figure 6: Introducing a delay

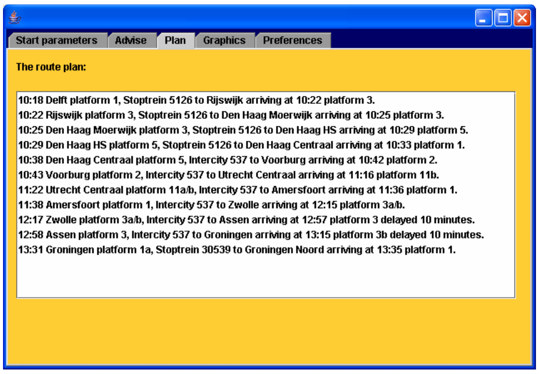

Figure 7: The delayed route plan

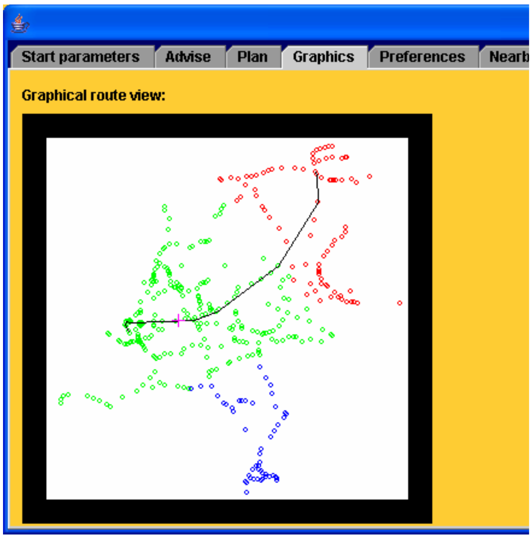

Figure 8: The route visualisation

Secondly the simulation environment offers the possibility to do tests with guidance of travellers based on incomplete information on delays. As in the case of the distributed handheld application, shown in figure 3, it is very important to guide users based on fragmented, incomplete information, the simulation environment can be used to test the effects in such an environment.

Lastly the performance of the route searching implementation was evaluated. By tuning the parameters of the search algorithm, the same results could be found as the route planner of the NS website generates. A minor issue is the performance of the searchalgorithm implementation, which is non linear, as can be seen in table 2.

Table 2: searchtimes of the DYNET implementation

From:	To:	Stops in solution:	Algorithm:	Searchtime:
Uitgeest	Haarlem	5	P1	50 msec
Uitgeest	Haarlem	8	C1	60 msec
Maastricht	Groningen	31	P1	591 msec
Maastricht	Groningen	17	C1	1432 msec

The P1 and C1 abbreviations stand for the performance, single (first) pass algorithm and the comfortable first pass algorithm. As can be seen, the search time increases dramatically with the complexity of the problem instance.

6. Conclusions

As part of this project a prototype and a simulation environment of the PITA system were implemented. The simulation environment offers the possibility to simulate guidance of travellers on the basis of incomplete information. This can be used to implement the fully distributed handheld application, using ad-hoc networking and by moving the different components running on the distributed backend to the handheld devices. However in this project the functionality of the PITA system was divided over handhelds and a distributed backend of servers as shown in figure 2.

The distributed Jade agent platform in combination with the Jadex extention were a good choice for implementing the system. The Jade platform forms a fault tolerant platform, while Jadex makes it possible to model agents after the BDI agent model. The integration of the current application with the COUGAAR architecture (KLT03) looks very promising, since the partitioned distributed blackboard of the COUGAAR architecture could be a good solution to sharing the delay and route information in a distributed handheld application.

The used DYNET algorithm has proven to be a good choice as the algorithm fits the discrete nature of the train schedules and also adapts to different user preferences very well. A disavantage of the algorithm is that the implementation does not use a linear amount of extra time for a more complex problem.

7. References

(BCPR03) F. Bellifemine, G. Caire, A. Poggi, G. Rimassa, *JADE – A white paper*, in:

EXP - In Search of Innovation (Special Issue on JADE), volume 3, number 3, Telecom Italia Lab, Turin, Italy, September 2003, pages 6-19.

(Egge01) G. Eggenkamp, *Dynamic Multi Modal Route Planning, An Artificial Intelligence Approach*, MSc thesis, TU Delft, July 2001.

(KLT03) K. Kleinmann, R. Lazarus, R. Tomlinson, *An Infrastructure for Adaptive Control of Multi-Agent Systems*, IEEE KIMAS'03 Conference Paper, october 2003.

(PBL03) A. Pokahr, L. Braubach, W. Lamersdorf, *Jadex: Implementing a BDI - Infrastructure for JADE Agents*, in: EXP - In Search of Innovation (Special Issue on JADE), Vol 3, Nr. 3, Telecom Italia Lab, Turin, Italy, September 2003, pages 76-85.

(RN95) S. J. Russell, P. Norvig, *Artificial Intelligence, A Modern Approach*, Prentice Hall, International, Inc., 1995.

(TRAIL) The Netherlands Research School for TRAnsport, Infrastructure and Logistics.

(Tulp91) E. Tulp, *Searching Time-table Networks*, PhD thesis, Vrije Universiteit van Amsterdam, Drukkerij Elinkwijk BV, Utrecht, oktober 31, 1991.

(Wool00) M. Wooldridge, *Reasoning about Rational Agents*, The MIT Press, Cambridge, Massachusetts, 2000.