Communication and metrics in agent convoy organization

Vincent Baines, Julian Padget
Dept. of Computer Science,
University of Bath
{v.f.baines,j.a.padget}@bath.ac.uk

ABSTRACT

Convoy formation, maintenance and dissolution is a multifaceted problem, with different domains creating a range of constraints, some of which can be helpful, but equally that can complicate the situation. The scenario under consideration here is, in the long-term, a mix of human- and agent-controlled vehicles in a public, transportation setting. However, the focus here is on agent-controlled vehicles and the problem of "knowledge fusion", or more precisely (i) how much/how little, and (ii) what kind of inter-vehicle communication is sufficient to enable adequate individual and group situational awareness to permit the effective operation of a convoy. This can be viewed as a global problem, but it is also a local problem, as each convoy participant must weigh up the costs and benefits arising from (i) the loss of autonomy – being subject to the governance of the convoy – and, (ii) the loss of privacy – needing to communicate data and intentions to some other convoy participants. We report on the first steps, examining communication issues and strategies, in realising this scenario by means of Belief-Desire-Intention agent controllers that operate vehicles in a 3D virtual environment.

Keywords

multiagent systems, intelligent transportation systems, convoy management

1. INTRODUCTION

Constructing an Artificial Intelligence that can enable vehicles to navigate under autonomous control has been an area of research for a number of years, with output from initiatives such as the US Defense Advanced Research Projects Agency (DARPA) "Grand Challenge" (e.g. 2005 winning entry [20]) showing the promise of real vehicles fitted with arrays of sensors being able to navigate through difficult terrain.

Given the number of research efforts (e.g. [12, 16]) that are demonstrating autonomous vehicles operating on roads, a potential progression of such work is automating vehicle convoys. Indeed, the 'SAfe Road TRains for the Environment' (SARTRE) project [2] is investigating the use of autonomous vehicle control to enable "vehicle platoons" and has produced outputs identifying convoy functionality required and what may be communicated within such a platoon. However, such work is in early stages, and there does not seem to be consideration of potential benefits in communicating higher levels of SA awareness (as aiding understanding of the situation). At the same time, Vehicle-to-Vehicle (V2V) communication is an active research area (e.g. [18]). However, the focus in V2V research tends to be on the hardware and network protocol layers, whereas our concern is what data should be communicated in order to allow vehicles to cooperate as part of a larger collective and to achieve common goals with measurable benefits e.g. improved fuel consumption, reduced journey time, etc.. We believe an open architecture is required that can enable the vehicles to pursue goals and manage their action selection through varying communication strategies. In order to produce an assessment of benefits in various approaches to V2V communication, such an architecture has been constructed.

An essential idea that has inspired our approach is that of *situational awareness* (SA) [10] and how its principles may be transferred to the domain of autonomous vehicles. The desire is to be able to capture sufficient information, at various levels of detail, about the environment, coupled with additional data pertaining to the vehicle itself, in order to be able to complement sensor-derived perceptions with higher level comprehensions about situation of the vehicle and its context.

The Belief-Desire-Intention [5] model has been adopted as an effective means to meet these requirements. BDI provides an agent based software architecture with a store of beliefs and available plans to achieve goals. These provide a loose mapping to core SA concepts (perception, comprehension, and projection), and facilitate the communication of a vehicle's current beliefs and future intentions to other vehicles. The aim is that this should augment the autonomous decision making process of other vehicles, as they will be informed of potential future events (e.g. an emergency stop occurring, and the reason for that stop) much more rapidly, rather than relying on physical sensors and beliefs inferred from that sensor feed.

The successful application of a BDI approach to convoy coordination has been demonstrated in earlier research [19] where focus was on varying convoy coordination methodologies (e.g. centralized, de-centralized, multi-agent team) and the impact on convoy split/merge activities.

We outline a range of scenarios that have been constructed to explore the impact of various V2V communication strategies, in place of or as a complement, to pure sensor approaches. The motivation here is that there are some issues which may be best perceived at a low physical sensor level, such as that the car in front is closer than some threshold. However, some, (such as the vehicle in front speeding up in order to bring the convoy speed closer to the optimum for fuel efficiency, or the vehicle behind is leaving the convoy because it is turning off at the next junction), are clearly at the level of information, not data. The same communication strategy is not necessarily appropriate for all three of these. In the first instance, we report on the use of two approaches: data push and data pull. Push is the basic and most obvious, where vehicles publish their position data to other vehicles (without it being requested), which may even avoid the need to depend on physical sensors in some circumstances. This permits convoy members to remain informed of the position of other vehicles in the convoy, and manage their own movement based on this. Pull demands a request-response protocol, but may reduce the overall level of communication, whereby a vehicle requests information if, say, they have not received updates in a required time window, and vehicles might share their current plans and future intentions.

The remainder of the paper is set out as follows: in the next section we survey some of the large amount of related work. In section 3 we outline the simulation testbed, comprising of the Tankcoders 3D environment and the agent driving team. We have identified several scenarios that we set out in section 4, before presenting some preliminary results in section 5. We finish with some issues for future work (section 6) and conclusions (section 7).

2. RESEARCH BACKGROUND

This work attempts to combine a number of research areas in order to tackle the problem of autonomous vehicle convoys. We focus first on situational awareness (SA) in order to inform the design approach to the simulation so that a vehicle's perceptions, comprehension, and projections are accessible and observable. A number of metrics have been proposed in an attempt to measure SA, and although it is a challenging unit to quantify, the effects of incorrect or lack of SA are dominant features in many accident investigations. Hourizi [14] relates Endsley's components of SA [11] failures in understanding the current state of an aircraft, given as:

1. Failure to perceive important elements in the environment;
2. Failure to comprehend the elements that have been perceived;
3. Failure to predict the future status of those components.

Because SA can be likened to human understanding of the environment, and it is this which informs human decision making, then it follows that an incorrect or lack of SA can be the cause of incorrect actions and decisions being taken. There is little (human) self awareness as to whether (one's) SA is accurate, complete, inaccurate, or incomplete – we have very limited awareness of what we do not know. Furthermore, belief in SA is measured by the self as well, and so liable to be fallible to the (self) human in the decision making loop. If, in contrast, we place the derivation of SA within an AI context, especially within a multiagent collective, where members can contribute to others' SA, this raises the possibility that some of the weakness identified above may be addressed. Revisiting the previous excerpt of Hourizi's work with this in mind:

1. Failure to perceive important elements in the environment;

   > *Perceptions (e.g. obstacle detected) are passed between members of the agent collective in order to negate this issue*

2. Failure to comprehend the elements that have been perceived;

   > *Other vehicles contribute their understanding of events; complex non-understood perceptions are referred to some other entity for resolution*

3. Failure to predict the future status of those components.

   > *Entities within the simulation exchange their intentions and goals, adding information as to how events are likely to unfold.*

The intention of combining the SA constructs with a BDI model is to address these issues and thus improve the ability of an automated system to control a vehicle, and furthermore with potential advantages over solely human control. Specifically, for the convoy scenarios being explored, members of the convoy are dependent on data exchange between their members. The aspiration is that allowing vehicles to exchange a range of, but especially higher level, data/information pertaining to their understanding of the current situation will aid members of the convoy in their action selection, improving the efficiency of the convoy as well as its ability to deal with unexpected events in the simulation.

That said, the issue of how much information should be passed between distributed agents also needs consideration. In [6], the approach proposed is to communicate only information that is needed and beneficial to other agents but it is not clear that the sender is capable of establishing these criteria. The motivation for addressing the quantity of communication is due to the cost of such communications and potential bandwidth limitations in the given scenario.

However, there is also concern regarding security and privacy: how much information should be revealed, as even some may be too much.There is a further advantage of an automated system handling such information exchange: that humans would find such communication tedious, even invasive, as well as distracting. Furthermore, humans most likely could not make good use of the information because the driving task is fairly routine. Recent [17] reports highlight that vehicle communication could lead to significant benefits in reducing motorway pile-ups.

The concept of self driving cars has been gathering increased momentum, with efforts by Google [16] to produce a self driving car, along with significant interest in developing such a concept from vehicle manufacturers. Earlier efforts took place during the series of DARPA funded challenges, and focusing on the 2007 entry of Tartan Racing, the "Perception and World Modelling" component [21], is of relevance here.

It performs "Situation Assessment" on received sensor data of tracked objects, integrates this with other knowledge of the world, and attempts to estimate the intention of this object. In relation to this SA-like concept, it is reported that the system struggles to perform well when approaching intersections and projecting future events (e.g. will a vehicle leave or join at that intersection). This provides an example of how communication of BDI constructs between vehicles could prove useful; rather than having to rely on some visual cue (e.g. an indicator light), vehicles would have been informed as to what was likely to happen at that intersection based on other vehicles belief and intention set. Allowing vehicles to communicate their planned events would potentially have the benefit that excessive braking and acceleration would be reduced, as vehicles are able to take account of expected future events rather than relying on last minute reactions.

Other work [19] has demonstrated the application of BDI to vehicle convoys in Collaborative Driving Systems (CDS), though the intention here is to explore what information exchange best supports SA (both individual and group) generation amongst the vehicles in order to improve road travel (safety, energy consumption, etc).

With this motivation and design selection in place, we turn to the construction of a suitable test bed which can support the assessment of different communication strategies, and their impact on convoy performance.

3. SIMULATION TESTBED

In order to assess the affect of various vehicle communication implementations, a testbed has been developed where a number of scenarios can be explored. To reduce the number of technological challenges faced from the outset, a simulation based approach has been selected as it offers the ability to assess performance of the system in a more controlled fashion. As the BDI component is software based, it can be tested using a simulated vehicle, allowing a base set of functionality to be established using some test scenarios.

There are a number of BDI implementations available, from which we have chosen to use Jason [3], because of its ease of extension using Java, an active support community, and the existing integration with the TankCoders virtual environment we are using for visualization of driving.

The TankCoders project [13] aimed to support research into Jason agent teams working cooperatively in a virtual environment, which it achieves by integrating Jason with a tank simulation based on the jMonkeyEngine 3D engine.

This has been revised as work has progressed, however the intention is to maintain it as being non-vehicle and non-domain specific with the aim of retaining applicability beyond the current vehicle scenario (e.g. unmanned aerial vehicles). This abstraction not only enables alternative vehicle types to be deployed within the TankCoders simulation, but decouples the high level call made from a Jason agent (e.g. `moveToXZ`) and the lower level implementation determined by the target platform (e.g. `turnWheel`, `applyTorque`, etc). This supports another objective of this work, which is to demonstrate the relevance of this research to real physical platforms as well simulated entities.

There is also the matter of building up an enhanced set of behaviours which could be considered as fundamental to the safe operation of the vehicle, an example of which is the emergency stop condition. The process of that behaviour itself resides in the agent and is not especially complicated (e.g. apply brakes, come to a complete stop, do no further actions), however by having that available, we can then consider situations in which it might be invoked. A 'bottom-line' approach to safety for any vehicle proceeding in a given direction is that, if there is some obstacle in that direction, which would be struck in the near future, then do not proceed any further in that direction. In other words, a collision avoidance behaviour. Such a behaviour also relates back to the earlier situational awareness notion, as it is a higher level inferred projection based on: (i) perceptions: obstacle detected and vehicle's current speed, (ii) comprehension: obstacle is of a significant size and will damage vehicle and (iii) projection: given current speed a collision will occur in n seconds.

With this in mind, we have developed a (naive) initial solution to the problem whereby each vehicle is placed within a collision volume that is constructed according to the vehicle's current orientation, speed, and a given time interval n. This represents the physical space that the vehicle will pass through for the next n seconds. If any object is detected within this volume, then the vehicle performs an emergency stop. Jason is able to construct this volume in the simulated environment and dynamically update as the vehicle moves, which demonstrates both a benefit of working in simulation (that the scenario can be augmented with data derived by the Jason agent, even going so far as to explain some of its SA), and the strengths of that agent in being able to derive additional data and use it to inform action selection.

Behaviours such as this can be thought of as providing primitive building blocks to allow much complex composite behaviours to be constructed. In the convoy scenario, vehicles may well follow closely behind each other, and this collision detection could get triggered if they follow too closely for some reason. In that case, the more primitive (and important) behaviour of avoiding a collision should take precedence, but the set of conditions which led to it occurring also need to be investigated in case there is a fault in how the convoy is behaving.

3.1 Agent capability

The simulation framework (elements of which are described above) provides one side of the story. The agents that are responsible for controlling the vehicles are the other. Thus, it was also necessary to develop agents that are capable of exhibiting convoy behaviour. Firstly, we have discarded the notion of a single controlling agent and replaced with a driving team, which allows us to break away from centralized control and to enable vehicles to be dynamically extended with additional capabilities as required, for example, only needing to instantiate one agent to handle convoy behaviour when the vehicle joins a convoy. The first agent used to supplement the coordinator agent is a *driver* agent, which has the primary responsibility for the vehicle's navigation. This separates off some responsibility, because the driver agent determines a speed and asks the central agent to achieve this, and introduces flexibility, because some other agent may request the coordinator agent to reduce speed, for example, and introduces robustness, for example the ability to encapsulate alternative solutions via the BDI plan failure mechanisms. More discussion of the driving team approach appears in the next section.

Agents are able to interact with each other via Jason's communication mechanism. The infrastructure supports two communication languages (KQML and FIPA-ACL), and at present we use KQML following from the TankCoders project. The specific implementation is not the subject of interest, but rather the capability which it provides. Whilst supporting intra-vehicle agent communication (for example the driver agent asks the coordinator agent to achieve some speed), it also provide an inter-vehicle communication capability, allowing not just information related to beliefs, but also plans and goals to be sent between vehicles. Vehicles can use such information to better improve the projection element of their SA, and perhaps also modify their own plans based on the plans of other vehicles. At present, only a small set of performatives are in use – primarily `askOne`, `achieve` and `tell`. This enables vehicles to add data to other vehicles knowledge bases (e.g. inform vehicles of obstacles which they may not have detected via physical sensors), enquire as to another vehicle's beliefs (e.g. what is your position), and ask another vehicle to achieve some goal (e.g. move to a given position). These are considered as fundamental to allow vehicle groups (in this case a specific convoy) to achieve a collective goal and handle self organisation. It is the effects of varying this communication behaviour which is the subject of investigation in the convoy scenarios and which is now presented, with the specifics of the convoy agent approach presented in the results section.

Key extensions to the agents' available plans are detailed below (details regarding beliefs have been largely omitted for the sake of brevity).

Coordinator agent:

- `+!chosenSpeed(V)` – set the vehicle speed via its API; update driver agent belief with new value.
- `+!requestTurnToAngle(A)` – call vehicle API to achieve an orientation.
- `+!updateColPred(X1,Y1,Z1,X2,Y2,Z2)` – update coordinates of collision prediction volume.

Driver agent:

- `+!emergencyStop`, `+!arrivedAtDestination` – ask coordinator to achieve zero speed and to unachieve `requestTurnToAngle(_)`, abolish own desired XZ and drop desire `moveToKnownPosition`.
- `+!cruise` – ask coordinator agent to achieve `+!chosenSpeed(V)`.
- `+!applyBrakes`, `+!standardSpeed`, `+!speedUp`, `+!slowDown`, – adjust vehicle speed away from default value.
- `+!moveToKnownPosition` – using `desiredXZ(X,Z)`, if arrived then `+!arrivedAtDestination`, otherwise determine direction A to X,Z; ask coordinator agent to achieve `requestTurnToAngle(A)`, then `cruise`.

Convoy member agent:

- `+vehAheadInfo(X,Y,Z,_,_,_,_)` (Data push scenario) Using X,Y,Z of vehicle ahead, determine distance to that vehicle and ask driver agent to achieve `standardSpeed`, `speedUp` or `slowDown` to maintain convoy gap. Tell driver agent to update its belief to `desiredXZ(X,Z)` and then to achieve `moveToKnownPosition`.
- `+!convoyMgmtPlan` (Data pull scenario) At three second intervals, ask the vehicle ahead `convoyMemberInfo(X,Y,Z,_,_,_,_)`. Tell driver agent to update its belief to `desiredXZ(X,Z)` and then to achieve `moveToKnownPosition`.

4. VEHICLE CONVOY SCENARIOS

The vehicle convoy domain has been selected for a number of reasons. This topic has been attracting attention recently, with the potential benefits of vehicle platoons being reported [2]: up to twenty percent reduction in fuel consumption, ten percent reduction in fatalities, and improved driver convenience (for passenger-drivers in the vehicles where control has been ceded to the platoon). Benefits have also been claimed recently [9] with improved traffic efficiency as a key goal. In relation to environmental considerations, [15] shows that the total trip time for journeys can be significantly improved through vehicle to vehicle communication. This study also shows that if navigation systems share traffic information, then journey times can be shortened, highlighting the potential benefit that sharing simple beliefs of BDI constructs may bring.

However, the application of this research does not reside purely in vehicle convoys; rather this has been selected as a key area where vehicle behaviours (such as information sharing and common goals) lend themselves to explore the benefits of SA-like knowledge exchange in a challenging but relevant context. As such, there are a number of limitations to the scenarios in use. Firstly, there is no road model; vehicles are bounded only by physics such as collisions with other objects and terrain. Secondly, there is no traffic model; at this stage we are only considering how vehicles communicate at an intra-convoy level. This will be extended once the scenarios become more broad as we wish to build up a larger SA picture into the convoy performance (e.g. external convoy members informing of obstacles ahead, and the convoy deciding on a course of action based on this). This will add further understanding to the question being posed, as if we are addressing how much intra-vehicle communication is required, it certainly follows that non-convoy member communication (e.g. position updates from other vehicles) requires consideration.

We have chosen to modularize the structure of the driving system by developing substantial new behaviours as separate agents, rather than as additional behaviours of an existing agent. This is already clear in the initial structure where there is a coordinator agent and a driver agent. The motivation is that new behaviours can be added (or removed) by the introduction (or removal) of an agent, rather than the modification of an existing agent: it only requires that the central coordinating agent is informed of new functionality (or its loss), while the collection of agents function as individual self-interested entities under the governance of the common objective of getting the vehicle to its destination (for example).

Two further motivations for this behavioural decoupling are: (i) to keep individual agent behaviour specifications "small enough" to be maintainable and to minimise the potential impact of hard-to-identify bugs arising from the aggregation of behaviour within a single BDI reasoning engine, and (ii) to keep constituent agent reasoning cycles short enough that response times might potentially be adequately controlled for close enough to real-time behaviour. It remains to be seen how well each of these is borne out in practice.

One such additional behaviour is the agent responsible for vehicle behaviour in the convoy collective. At its most basic, on instantiation this agent is informed of the vehicles in front and behind in the convoy of which it is a member, and on receipt of the vehicle in front position data, it seeks to move to that position. Both the driver agent and convoy member agent are currently generic, so the same agent capability is embedded across all the vehicles in a convoy.

As mentioned previously, KQML is used as the communication language in this testbed. KQML is used at both intra- and inter-vehicle communication. For example to allow a driver agent to request a speed from the coordinator agent, or to allow a coordinator agent to update a driver agent on current position within simulation. At inter-vehicle level, it allows the convoy agent of vehicle 1 to ask the convoy agent of vehicle 2 for the current location of vehicle 2.

It is this communication mechanism, coupled with the intrinsic BDI data constructs, which is the topic of interest regarding the impact it has on the success and efficiency of convoy behaviour(s) in the scenario(s). The main communication strategies can be broken down in line with the BDI paradigm, that is, inter-vehicle sharing of beliefs, desires and intentions.

Our first step has been an investigation of the benefits of sharing beliefs and two convoy scenarios are presented in section 5 based on this. The two approaches differ in how the data is transmitted; the first requires all vehicles to inform other vehicles of their position at every tick of the simulation, while the second implements a request approach where each vehicle determines when to ask some other vehicle for its current position details.

At present, we are focussing on belief sharing and this has produced some initial statistics and observations on (convoy) behaviour, depending on whether the data is pushed (i.e. sent out at some tick interval to n agents), or pulled (agents request information from other agents at their chosen interval). The first implementation of a convoy has been based on each convoy member knowing the identity of the convoy member behind it, and at each simulation tick using a KQML performative `send(vehicleBehind, tell, vehicleInfrontPosition(X,Z))` to advise its position to the vehicle following. Upon receipt of a `vehicleInfrontPosition(X,Z)` belief, a convoy member establishes the goal of `moveToXZ(X,Z)` thus following the path of the vehicle in front.

5. RESULTS

The experiments presented at this point are based around five scenarios. The first three scenarios are baseline assessments of the operation of the framework, and use no convoy member agents. The fourth scenario implements the data push communication strategy, in which, data is pushed at a regular interval between convoy members, where each member passes its position to the vehicle behind. The fifth scenario implements the data pull strategy, in which data is pulled by request from a specified agent to the requestor. Precise details are given in the following section.

5.1 Convoy scenarios

The detail of each scenario, and the intention of what it should assess, is as follows:

Scenario 1: Four vehicles, with a driver agent but no destination to achieve, no convoy member agent. Assess: Baseline of physics simulation and rendering of four vehicles.

Scenario 2: Four vehicles, with a driver agent and given a destination, no convoy member agent. Assess: Impact of using the driver agent on the initial baseline.

Scenario 3: Two vehicles, with a driver agent and given destination, no convoy member agent. Assess: Impact workload of half as many vehicles and driver agents.

Scenario 4: Four vehicles, with a driver agent, where lead vehicle's driver agent is given a destination, and each vehicle has a convoy member agent based on convoy strategy 1. Assess: Initial convoy strategy dependent on high communication traffic between convoy agents.

Scenario 5: Four vehicles, with a driver agent, where lead vehicle's driver agent is given a destination, and each vehicle has a convoy member agent based on convoy strategy 2. Assess: Impact of reduced communication between convoy agents.

In the following push and pull strategies, there is an assumption that a 'convoy join' behaviour has already taken place, resulting in three vehicles following the lead vehicle. Part of this behaviour would involve determining whether the convoy is heading (at least partly) in the direction required. On joining the convoy, there is an abdication of route planning responsibility as part of the ceding of individual autonomy to the collective convoy, and instead navigation involves following the trail of the vehicle in front.

5.1.1 Data push strategy

Convoy strategy one implements the following approach:

- Only the lead vehicle knows the final destination.
- The lead vehicle's coordinator agent starts the movement by sending a message to its driver agent regarding the desired location:
  `send(driverAgent, tell, desiredXZ (500,2500))`
  followed by a message to achieve a movement to that destination:
  `send(driverAgent, achieve, moveToKnownPosition)`.
- Each vehicle in the convoy starts a convoy member agent.

- Each convoy member agent is told the vehicle's driver agent name (in order to be able to send messages regarding updated positions to move to) and the name of the convoy member agent of the vehicle behind (in order to push data to the correct vehicle).
- On every simulation update cycle, the coordinator agent tells its convoy member agent and driver agent the vehicle's new position.
- When a driver agent receives a position update, it uses this to calculate the distance remaining to the `desiredXZ` and perform any necessary actions (e.g. course corrections, or stop if at that location).
- When a convoy member agent receives a position update from its coordinator agent, it performs the data push of telling the following convoy member agent this new position.
- When a convoy member agent receives a position updates from the vehicle ahead, it tells its driver agent as a new `desiredXZ` followed by the request to achieve `moveToKnownPosition`.

5.1.2 Data pull strategy

Convoy strategy two implements the following approach:

1. Only the lead vehicle knows the final destination.
2. The lead vehicle's coordinator agent starts the movement by sending a message to its driver agent regarding the desired location:
   `send(driverAgent, tell, desiredXZ (500,2500))`
   followed by a message to achieve a movement to that destination:
   `send(driverAgent, achieve, moveToKnownPosition)`.
3. Each vehicle in the convoy starts a convoy member agent.
4. Each convoy member agent is told the vehicle's driver agent name (in order to be able to send messages regarding updated positions to move to) and the name of the convoy member agent for the vehicle ahead (in order to pull data from the correct vehicle).
5. Each convoy member agent starts a convoy management plan, in order to handle the data pull aspect. At present, every 3 seconds this plan uses the KQML performative `askOne` to ask the vehicle ahead's current position.
6. When a convoy member receives a reply containing the position of the vehicle ahead, it sends this to its driver agent as a new `desiredXZ` followed by the request to achieve `moveToKnownPosition`.

5.2 Scenario results

The focus of the discussion here is on scenarios four and five, as these demonstrate the affect of varying the convoy communication strategy. Scenarios one, two and three demonstrate consistent behaviour and sufficient performance of the simulation to have confidence in the output from scenarios four and five.

Figure 1 plots the position reports of each convoy member, as the convoy forms and moves from starting positions (approximately 0,2000) to the fixed destination given to the lead vehicle (500,2500). Figure 1 shows the convoy positions during the fourth scenario, i.e. convoy strategy one (data push). By comparison, in Figure 2 the position of each vehicle is shown for the fifth scenario, i.e convoy strategy two (data pull). In this simple situation where the destination of the lead vehicle is not changing, it is evident that there is little difference between the two approaches in terms of the route taken by the convoy and its member vehicles. However, it can be observed that, during the transition from start conditions to the steady state behaviour when moving towards the (non-changing) destination, vehicle 4 (starting at the farthest left in the figures) does differ between the two scenarios. This shows the impact of increasing the gap between position updates: between updates from the vehicle in front vehicle 4 has diverged slightly from the convoy direction. In this particular scenario (i.e. with a fixed destination for the lead convoy vehicle) there is no real impact as, with the next position update, the vehicle realigns to the convoy. However, in a case where the lead convoy vehicle changes route more frequently (i.e. navigation through a congested city with many intermediate destinations or waypoints) this could result in greater divergence from the convoy grouping. This also suggests where benefits may arise from obtaining higher level information from the convoy member in front (e.g. desired final location) rather than low level details (current position), and this will be the subject of investigated in future scenarios.

During the scenarios involving these two convoy strategies data was collected to capture the effect of the different approaches on the communication volume, specifically the number of percepts and the number of messages. In Figure 3 the marked contrast can be seen between strategy one (data push) and strategy two (data pull). The approach of a data push has resulted in approximately five times as many percepts being registered by vehicles two, three and four compared to the same vehicles using a data pull approach. This result is expected, as the same data (of vehicle position) is being communicated in both scenarios, however in the data pull strategy the communication frequency is lower (as the responsibility for when to ask for this data resides with the receiving vehicle and it does so every few seconds compared to at every opportunity) and as such fewer percept updates are received. In Figure 4 the number of messages transmitted for the four vehicles is shown, and a similar profile to that of Figure 3 can be seen. Compared to Figure 3 however, data points do not begin until approximately 20 seconds have elapsed. This is due to how the simulation starts; as soon as the environment is instantiated (and the vehicle agents created) they begin receiving percepts. However, until the vehicles begin moving and using the convoy strategy there will be no exchange of messages, and this only occurs after the simulation has been fully initialised (approximately 20 seconds into the data capture).

In both figures it can be seen that vehicle 1 follows the same profile across both scenarios. This provides an expected correlation, as in both convoy strategies the lead vehicle is performing a role where its communication is significantly different to the rest of the convoy. The lead vehicle receives no external position updates as it is not following any other vehicle, instead only sending data to the vehicle behind. This would seem to confirm that there are no other sources of percept generation (e.g. mass broadcast of position data to all convoy members rather than to the specified target vehicle), which confirms the observed results are indeed due to

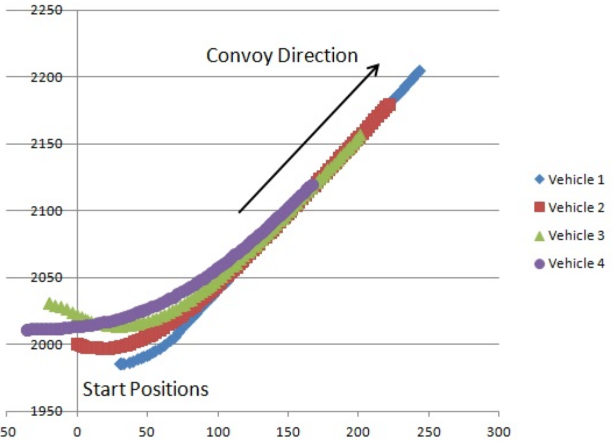

Figure 1: Vehicle positions with convoy strategy one

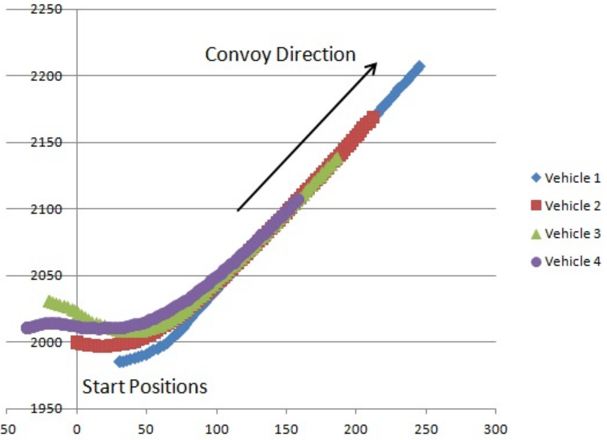

Figure 2: Vehicle positions with convoy strategy two

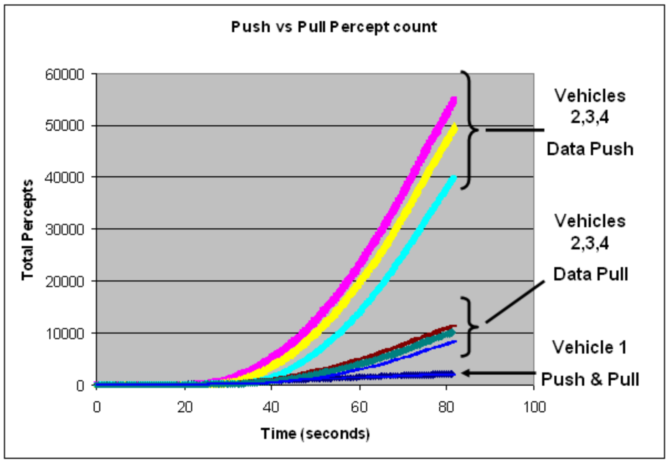

Figure 3: Vehicle percept updates

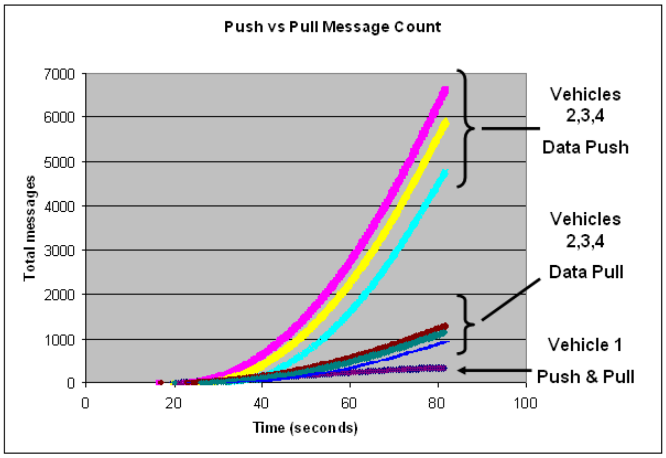

Figure 4: Vehicle message counts

the variation in convoy strategy rather than other factors.

Although no specific performance metrics have been developed yet, there was a notable impact on the simulation during scenario five, as the frames per second rate dropped from approximately 19fps to 12fps. The system performance (measured by frame rate) during these two scenarios is shown in Figure 5, where this difference in performance can be seen. With the performance of the simulation dropping to such levels, we conclude that the resources consumed by communication are impacting the ability of the system to carry out computation. Video capture of the two convoy strategies is available in mp4 format, for scenario 4 (data push) at `http://people.bath.ac.uk/vb216/dataPush.mp4` and for scenario 5 (data pull) at `http://people.bath.ac.uk/vb216/dataPull.mp4`.

In both videos it can be seen that the frame rate differs from that shown in Figure 5, due to the increased load on the system of capturing the video stream. However, it can be seen that there is still a performance difference between the two scenarios, with data-pull outperforming data-push.

If the simulation performance drops much further, it has been observed that unexpected and unpredictable agent behaviour occurs and convoy behaviour breaks down. This issue is one of the motivations for the decoupling of software components discussed in the next section.

6. FUTURE WORK

At present Jason is quite tightly integrated with the TankCoders platform, which is good in some respects for performance, although we have already experienced stochastic behaviour arising from tracing that has further obscured the issues we were attempting to observe. In the next phase of our work, despite some concern over the performance impact of the introduction of middleware, we wish to decouple the various components for four reasons:

1. We seek to avoid a repeat of the problem cited above, that monitoring perturbs the system further.
2. Experience elsewhere has taught us that large numbers of agents on a single Jason instance can be problematic, so we would like to be able to connect multiple Jason instances to a single TankCoders environment. In addition, this would permit driver teams to be lo-

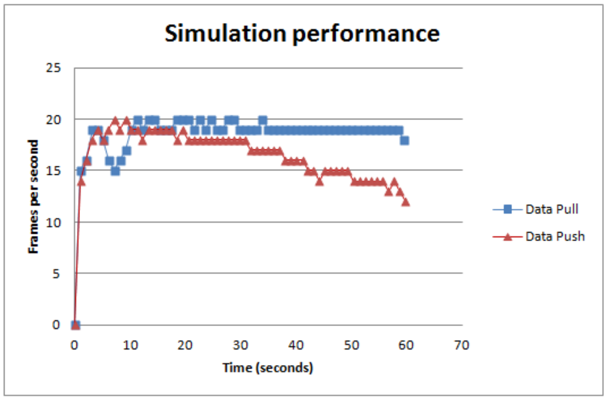

Figure 5: System performance variation

cated anywhere on the Internet, not just on the same machine as the virtual environment.

3. A critical feature of the next phase is the introduction of normative framework [8] to capture the rules of the convoy in the form of an externally reference-able entity that governs the behaviour of individual teams, as well as subsequently exploring interaction between convoy instances [7] to handle operations such as merging, splitting and passing through one another. Previous experience [1] of its integration, encourages us to decouple the normative framework from the agent platform.
4. Finally, useful though working with the TankCoders environment is, the harshest environment is the physical world and so we wish to substitute physics models of vehicles with simple robot vehicles, in this case LEGO Mindstorms platforms carrying android mobile phones as communication devices.

In pursuit of these goals, we are currently developing the means for the various components identified above to communicate using the Extensible Messaging and Presence Protocol (XMPP). XMPP is in widespread use underpinning internet messaging systems, but it is equally applicable for inter-program communication and for the collection of sensor data (our initial application). Thus, by taking an event-oriented view of the world and treating each of the above components as event consumers and producers in conversations enabled by XMPP, it is relatively straightforward to achieve the desired decoupling.

Further work is also necessary on the simulation system itself. Some refinement is required on the generation of system level metrics, such as those presented in Figures 3 and 4, in order to improve the efficiency of data collection and ensure there is minimal impact on system performance. An extension is also planned to provide a breakdown of the message and percept types being communicated, in order to understand further what is being exchanged between agents. As the scenarios grow more complex, this is expected to be essential in order to follow the interactions occurring between the vehicles and their agents.

The scenarios also need to be extended to add both realism and challenge to the vehicle agents. A more complex convoy route is required to more clearly demonstrate merits between differing convoy strategies, and also to identify strengths and weaknesses of varying communicated data (e.g. beliefs vs intentions). As the complexity increases, so too will the likelihood of calling upon the ability of Jason to handle plan failures, as unexpected situations and interactions occur.

With this in place, more advanced metrics measuring the impact of varying convoy strategies are needed. Two already under development are fuel management and convoy route deviation measures. The first involves the integration of a simplified engine model into the simulation, such that inefficiencies (e.g. high engine revs, excessive acceleration and braking) in the drive of the vehicle will be reflected in the fuel consumption. Such work will also introduce the ability to explore competing objectives between agents (e.g. fuel management requiring a slow speed to conserve fuel, convoy agent requiring a high speed to maintain convoy position). The second, convoy route deviation measure, is to extend the results being produced which produced Figure 1 and Figure 2. This will produce a metric indicating how well the convoy is performing in geographic cohesiveness and highlight deviations from its route.

A major future development is to utilise a normative framework within the system and to capture a reasonable set of both legal governance and societal convention into this architecture. The design and implementation of the normative solution will require significant effort. Work has been presented in [1] demonstrating a methodology for the utilisation of institutional models of governance in open systems. This raises a number of questions which will need consideration, such as whether an individual agent should refer its action selection to a normative control, or does a normative agent model actions at an individual vehicle level, how will the convoy be regulated, and are certain actions allowable but involve a punishment mechanism? The work of [1] also demonstrates the feasibility of integrating BDI (and specifically Jason) with institutional models. The work of Bradshaw [4] also touches on the notion of potential actions vs permitted actions, raises the question of how some adjustable autonomy will be managed (e.g. action selection when in convoy vs action selection when driving as individual). Some larger scenarios are likely to be required to investigate these questions, and the effort of both this and the normative framework itself positions this work in a more ready state for transition into a real-world domain.

As discussed earlier, another aspect of development is the decoupling of this study from the TankCoders-jMonkeyEngine simulation in order to connect it with a real sensor-actuator capability. This process is underway, with integration to an XMPP based sensor framework in early stages. This will allow the simulation of a vehicle instance from TankCoders to be replaced with a real vehicle, passing geographical data back to Jason and responding to Jason agent requests. Work is progressing to formalise the specifics of message exchange format, and this will then form the basis of a ratification of the V2V communication strategy by introducing real world limitations, e.g. latency, bandwidth.

We plan to experiment with an Android device coupled with a real world platform (a remote control car with an IOIO breakout board) that provides a sensor suite from the android device (orientation, position) coupled with an appropriate actuator. In addition, work is in progress to couple this system to the LEGO Mindstorms platform, with a fuller set of functional XMPP message being developed.

7. CONCLUSIONS

This work demonstrates that a usable simulation framework has been constructed, capable of supporting the next phase, which will focus on the investigation of benefits in BDI type message exchange to support of vehicle convoy behaviour. Results to date are:

1. Demonstration of a working simulation with Jason Belief-Desire-Intention agent controlled simulated vehicles.
2. An agnostic control design, where agents are not specific to a vehicle (e.g. weight, size, power), or type (e.g. locally simulated vehicle, or remote XMPP vehicle).
3. An initial convoy scenario exploring the performance of a 'data push' of vehicle positions.
4. A second convoy scenario exploring the performance of a 'data pull' of vehicle positions.
5. An initial suite of metrics to measure aspects of system and convoy performance.

Having established our foundations, we will now work towards more credible vehicle scenarios. Following this, the integration of a normative framework will be explored such that the governance of this vehicle collective is established. Finally, the applicability to real platforms will be demonstrated through the use of remote physical vehicles.

8. REFERENCES

[1] Tina Balke. *Towards the Governance of Open Distributed Systems: A Case Study in Wireless Mobile Grids*. PhD thesis, University of Bayreuth, November 2011. Available via http://opus.ub.uni-bayreuth.de/volltexte/2011/929/. Retrieved 20120109. Also available as ISBN-13: 978-1466420090, published by Createspace.

[2] C. Bergenhem, Q. Huang, A. Benmimoun, and T. Robinson. Challenges of platooning on public motorways. In *17th World Congress on Intelligent Transport Systems*, 2010.

[3] R. H. Bordini, J. F. Hübner, and M Wooldridge. *Programming multi-agent systems in AgentSpeak using Jason*. Wiley, 2007.

[4] Jeffrey M. Bradshaw, Paul J. Feltovich, Hyuckchul Jung, Shriniwas Kulkarni, William Taysom, and Andrzej Uszok. Dimensions of adjustable autonomy and mixed-initiative interaction. *Computational Autonomy*, 2969:17–39, 2004.

[5] M. E. Bratman, D. J. Israel, and M. E. Pollack. Plans and resource-bounded practical reasoning. *Computational Intelligence*, 4:349Ű355, 1988.

[6] J. Chen. Distributed coordination of autonomous agents by communicating on a need-to-know basis. Master of engineering, Massachusetts Institute of Technology, 2003.

[7] O. Cliffe, M. De Vos, and J. A. Padget. Specifying and reasoning about multiple institutions. In *Coordination, Organizations, Institutions, and Norms in Agent Systems Lecture Notes in Artificial Intelligence*. Springer, 2007.

[8] Owen Cliffe, Marina De Vos, and Julian Padget. Answer set programming for representing and reasoning about virtual institutions. In Katsumi Inoue, Ken Satoh, and Francesca Toni, editors, *CLIMA VII*, volume 4371 of *Lecture Notes in Computer Science*, pages 60–79. Springer, 2006.

[9] F. Dressler, F. Kargl, J. Ott, O. Tonguz, and L. Wischhof. 10402 abstracts collection and executive summary – inter-vehicular communication. In *Inter-Vehicular Communication*, number 10402 in Dagstuhl Seminar Proceedings, Dagstuhl, Germany, 2011. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, Germany.

[10] M. R. Endsley. Toward a theory of situation awareness in dynamic systems. *Human Factors*, 37:32–64, 1995.

[11] Mica R. Endsley. Toward a theory of situation awareness in dynamic systems. *Human Factors: The Journal of the Human Factors and Ergonomics Society*, 37(1):32–64, 1995.

[12] Freie Universitat Berlin. Autonomous car navigates the streets of Berlin. http://autonomos.inf.fu-berlin.de/news/press-release-92011, September 2011. accessed October 8th, 2011.

[13] Germano Fronza. Simulador de um ambiente virtual distribuido multiusuario para batalhas de tanques 3d com inteligencia baseada em agentes bdi. http://campeche.inf.furb.br/tccs/2008-I/2008-1-14-ap-germanofronza.pdf, July 2008.

[14] R. Hourizi. *Awareness beyond mode error*. PhD thesis, University of Bath, 1999.

[15] I. Leontiadis, G. Marfia, D. Mack, G. Pau, C. Mascolo, and M. Gerla. On the effectiveness of an opportunistic traffic management system for vehicular networks. In *IEEE Transactions on Intelligent Transportation Systems*. IEEE Intelligent Transportation Systems Society, 2011.

[16] J. Markoff. Google cars drive themselves, in traffic. http://www.nytimes.com/2010/10/10/science/10google.html, October 2010. accessed October 8th, 2011.

[17] Katia Moskvitch. 'Talking' cars could reduce motorway pile-ups. http://www.bbc.co.uk/news/technology-14125245, July 2011. accessed October 8th, 2011.

[18] M. Raya, P. Papadimitratos, and J.-P. Hubaux. Securing Vehicular Communications. *IEEE Wireless Communications*, 13:8–15, October 2006.

[19] Brahim Chaib-draa and Simon Hallé. A collaborative driving system based on multiagent modelling and simulations. *Journal of Transportation Research Part C (TRC-C): Emergent Technologies*, 13(4):320–345, 2005.

[20] Sebastian Thrun and Mike Montemerlo et al. Stanley: The robot that won the DARPA Grand Challenge. *Journal of Field Robotics*, 23(9):661–692, 2006.

[21] C. Urmson and W. Whittaker. Self-driving cars and the urban challenge. *IEEE Intelligent Systems*, 23:66–68, March 2008.