Cooperating Sentient Vehicles for Next Generation Automobiles

Thirunavukkarasu Sivaharan, Gordon Blair, Adrian Friday, Maomao Wu,
Hector Duran-Limon, Paul Okanda, Carl-Fredrik Sørensen
Computing Department, Lancaster University, Lancaster, LA1 4YR, U.K.
Tel: +44 (0) 1524 593315
{ t.sivaharan, gordon, adrian, maomao, duranlim, okanda } @comp.lancs.ac.uk
carlfrs@idi.ntnu.no

ABSTRACT

It is becoming clear that location-aware intelligent transportation systems will be one of the most promising upcoming applications for next generation vehicles. The driving force behind this is the introduction of pervasive high performance wireless networks and location sensing technologies, such as GPS and roadside detection systems. Intelligent transportation systems utilise inter-vehicle cooperation without human assistance to provide autonomous vehicle navigation from a given source to a pre-determined destination. The resultant sentient vehicles are 'context-aware' autonomous cars that form cooperative `flotillas' of peers using mobile ad hoc network environments (MANETs). In this paper we report on our experiences of building a component framework based middleware architecture designed to meet the challenges of such environments. We show how such a framework can be used to engineer a proof of concept cooperating sentient vehicle application, and highlight the research challenges raised.

1. INTRODUCTION

The cooperating sentient vehicle application is a key demonstrator of the technology developed by the EU Framework V IST funded CORTEX project[1] [7].The sentient vehicles are 'context-aware' cooperating autonomous cars, which autonomously navigate to a given destination. CORTEX is concerned with developing middleware support for constructing 'proactive applications' based on a paradigm we call real-time sentient objects. CORTEX proposes a sentient object model based on anonymous event-based communication. Generally speaking, the systems consist of an environment and a set of sentient objects that are capable of independently sensing this environment, deriving context and inferring autonomous action. Applications built from sentient objects may communicate using event channels to establish higher-level contexts and thus cooperate with each other.

We have chosen to focus on a particular proactive application, the 'cooperating sentient vehicles'. This application has been made possible with the recent technological advances including, wireless networking such as 802.11b capable of operating in ad hoc mode with a bandwidth of up to 11Mbps, improved location accuracy provided by GPS which is capable of 3m-5m relative positional accuracy, ultrasonic sensors for obstacle sensing, magnetic digital compasses and Pocket PCs. However, there are key research issues that need to be addressed to realise such applications. We have identified the challenges posed by cooperating sentient vehicle application, and built a resultant Component Framework (CF) based middleware architecture. The middleware architecture addresses the many challenges raised by mobile context-aware applications operating in MANETs. We have implemented this middleware platform and used it to build a prototype of the cooperating sentient vehicle application.

The remainder of the paper is structured as follows. First, section 2 briefly describes the application scenario. Section 3 discusses the challenges raised by the cooperating sentient vehicle application and outlines our approaches to address the challenges using component framework based middleware technology. Section 4 briefly describes our sentient vehicle test bed. Finally, section 5 provides our conclusions.

2. COOPERATING SENTIENT VEHICLE APPICATION SCENARIO

The demonstrator application is divided into two sub-problems: 1) Cooperative behaviour without human control, and 2) Autonomous vehicle navigation from a given source to pre-determined destination. The autonomous vehicles have the objective of travelling along a given path, defined by a set of GPS waypoints (a 'virtual' circuit). Every vehicle that travels along the path cooperates with other vehicles by inter-vehicle communication. Each vehicle needs to build a real-time perception (an RTImage) of its surrounding environment within some bounded error to make informed decisions regarding its next move. The cooperation between vehicles is critical to avoid collisions, to follow a leading vehicle and to travel safely. The vehicles must obey external traffic signals and give way to pedestrians who cross the road (by sensing their presence). The intelligent vehicles can be deployed in any outdoor arena. Vehicles need to travel from one location to another with minimum driver assistance. Before a journey, vehicles are notified about the virtual circuit waypoint information and bearings.

[1] This work was supported by EC, through project IST-FET-2000-26031 (The CORTEX Project), http://cortex.di.fc.ul.pt/

3. CHALLENGES AND MIDDLEWARE ARCHITECTURE

The key research challenges, that need to be addressed, to enable the application scenarios described in section 2, are: communication model, routing protocol, context-awareness, end-to-end QoS (Quality-of-Service) and fail-safety. We address these challenges and provide our solutions as different component frameworks (CF). The middleware platform consists of, Publish-Subscribe CF, Group communication CF and Context CF. Component frameworks (CF) enforce the functional and non-functional properties of the system, and keep consistency across adaptations triggered by applications. A particular instantiation of middleware components for the MANETs is shown in figure 1.

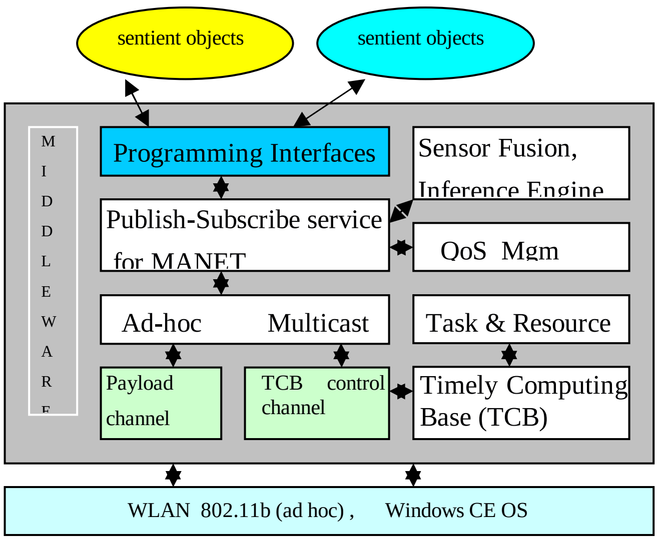

Figure 1 – Middleware Platform for MANET

Communication model: A key challenge that needs to be addressed by the cooperating sentient vehicles application is the suitable communication model. The client-server or RPC-based paradigms supported by the state-of-the-art object oriented middleware such as CORBA and DCOM are well suited for fixed infrastructure based wired networks, but they are not well suited for MANETs. In mobile ad hoc networks any centralised infrastructure based highly coupled, synchronous communication model is not well suited [5]. Since there is no fixed infrastructure to host centralised services, disconnection is the norm and communicating nodes are generally anonymous. To address this, a loosely coupled, asynchronous, anonymous and a fully decentralised communication model is required. The Publish-Subscribe communication model has the aforementioned properties. However, most of the state of the art publish-subscribe or event based middleware are based on centralised event brokers. Our middleware takes the approach of having a public-subscribe (event based) communication model, with all the aforementioned required properties and was especially designed for MANET. The design of Publish-Subscribe CF was inspired by STEAM [3]. Events are transported via selectable ad-hoc multicast protocol described below.

Routing Protocol: Routing in mobile ad-hoc networks is a challenging issue because of frequent topological changes in networks. Publishers and subscribers move frequently, posing a challenge for routing of events in wireless ad-hoc networks. Multicast routing based on proactive and reactive ad-hoc routing, using shared state kept in the form of routes and adjacent information, is useful in environments with low node mobility. However, in scenarios with high node mobility such protocols are unsuitable as shared state and topology information can quickly become outdated. For this reason as part of the Group communication CF, we have built a Probabilistic Multicast Protocol for Wireless Ad Hoc Networks. The protocol specifically targets ad hoc environments with high node mobility and a frequently changing topology of group members. The design of the probabilistic multicast protocol was inspired by previous research on multicast algorithms (both proactive and reactive) for ad-hoc networks, which shows that most existing algorithms (AMRoute, CAMP, MCEDAR, AODV, etc.) [6] perform inadequately when high node mobility is present in the environment. Our multicast routing protocol is based on a probabilistic flooding algorithm with damping, which does not maintain shared state in nodes. The Group communication CF also includes IP multicast protocol. Publish-subscribe CF can dynamically reconfigure to utilise a multicast protocol from the Group communication CF.

Context-awareness: Another challenge is context-awareness in highly dynamic physical environments. The sentient vehicle requires higher level context information such as relative location, orientation and status of the entities in its proximity, in real time, to autonomously decide its speed and steering control actuations. The fundamental challenge is that it is not possible to construct an exact 'image' (perception) of the surrounding environment. Therefore, there is a risk of wrong decisions being made based on inaccurate information. We follow the sentient object paradigm [7] to model our context aware system. In broad terms, sentient objects are objects that consume events from variety of different sources including sensors and event channels, fuse them to derive higher level contexts, reason about using a expert logic (a C Language Integrated Production System-CLIPS inference engine)[4], and produces output events whereby they actuate on the environment or interact with other objects. The Context CF provides the facility for supporting a range of inference engines and sensor fusion algorithms (that may be selected at runtime). For example, one fusion component provides algorithms to fuse sensor data from GPS, ultrasonic, compass and context events received via event channels and derive higher-level contexts. The fusion component algorithms include Gaussian modelling and dead-reckoning, together with home grown algorithms to fuse noisy sensor data and to help build a more accurate real time 'image' of the environment.

End-to-End QoS Management and fail safety: In cooperating sentient vehicle applications, timely event delivery and awareness of the QoS of the event channels used for inter-vehicle communication are crucial for fail-safety. Dealing with highly dynamic interactions and continuously changing environments, at the same time, with needs of predictable operations is a major challenge. The key issue in operating in uncertain environments is that timing bounds for distributed actions may be violated because of timing failures. Therefore when executing in uncertain environments, distributed operations with timeliness requirements must be able to deal with timing failures. The problem is solvable, if we could assume a reliable and timely propagation of events through the event channels. We assume in our architecture, that we can model the uncertainty of wireless communication using a dependable timing failure detection service for distributed operations. In our middleware this is provided by University of Lisboa's Timely Computing Base (TCB) [2]. The TCB provides the facility to monitor timeliness of event delivery on distributed event channels, thus providing estimations and awareness of timing failure probability for a given required coverage. We achieve fail-safety with timing failures by switching the vehicles to a fail-safe state as soon as a critical timing failure occurs.

Finally, we believe, adequate middleware support also encourages the widespread development of promising applications such as cooperating sentient vehicle. Without a middleware platform, the application programmer is faced with the immense task of dealing with low level sensors, fusion of sensor data, distributed computing, wireless networking, and adapting to changing QoS. Previous experiences in construction of reflective middleware has build the basis to use reflection, component technology and component frameworks (CF) based approach to create aforementioned middleware platform, see [1] for more details on our reflective middleware approach.

4. SENTIENT VEHICLE TEST BED

A sentient vehicle used in the demonstrator application is shown in figure 2.

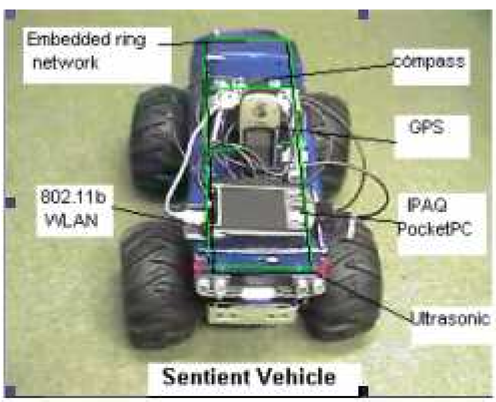

Figure 2 –A top view of sentient vehicle

The sentient vehicle is a modified Juggernaut 2 remote control (RC) Car. The RC module has been removed and the car is controlled by an HP iPAQ Pocket PC via a RS232 connector. The car is augmented with a GPS module for location sensing, an electronic compass for sensing its orientation, 8 units of ultrasonic sensors (with 3cm to 3m range) to sense the presence of neighbouring physical objects. The Pocket PC has 2 WLAN cards (configured for ad-hoc mode use). The more powerful WLAN card is for the exclusive use of TCB control channel (which requires a lightly loaded network) and the other WLAN card is for the event channels (payload) used for inter-vehicle communications. The two WLAN cards operate in non-overlapping 802.11b channels. An on-board CAN is used on the sentient vehicle to broker data between the sensors, actuators and iPAQ. The on-board network is a bespoke ring topology with single break failure resilience. Importantly, the design enables addition of further devices on to the ring in a plug and play fashion making it extensible. The test bed contains a small number of sentient vehicles and laptop acting as a sentient traffic light. The pocket PC mounted on the sentient vehicles has the Windows CE version of the middleware platform, which was implemented using embedded Visual C++.

5. CONCLUSIONS

This paper has described an interesting and challenging application domain involving the co-ordination of autonomous vehicles in a MANET environment. The scenario described in section 2 is now fully operational. Based on the experience of constructing this application, the key results are: i) sentient object model has proved to be an excellent programming abstraction for the development of such applications, particularly because of their intrinsic support for context awareness; ii) there is a real need for middleware in this area to ease the burden on the application developer and also to provide support for the management of non-functional concerns such as timeliness properties; iii) the properties of configurability and re-configurability inherent in our approach are highly suited to this domain, for example to encourage the construction of adaptable or autonomic systems. In ongoing work, we are particularly keen to investigate the generality of the approach to other domains such as environmental monitoring, smart buildings, mobile systems for disabled or elderly people and retail systems.

6. ACKNOWLEDGMENTS

We would like to thank all the CORTEX partners for their contributions, Angie Chandler and Joe Finney for the construction of the sentient vehicles, George Samartzidis for building a multicast protocol for ad hoc networks and Trimble technologies Europe for GPS Kits.

7. REFERENCES

[1] Coulson, G, Blair, G.S, Clark, M, Parlavantzas, N, "The Design of a Highly configurable and reconfigurable Middleware platform" ,ACM Distributed Computing Journal, Vol 15,No 2, pp109-126, April 2002.

[2] Antonio Casimiro, Paulo Verissimo. Using the Timely Computing Base for Dependable QoS Adaptation. In Pro of the 20th IEEE Symposium on Reliable Distributed Systems, pages 208–217. IEEE Computer Society Press, 2001.

[3] Rene Meier, Vinny Cahill. STEAM: Event-based Middleware for Wireless Ad Hoc Networks. In Proc. of the International Workshop on Distributed Event-Based Systems (ICDSC/DEBS02), pages 639–644, 2002.

[4] NASA CLIPS: A Tool for Building Expert Systems http://www.ghg.net/clips/CLIPS.html

[5] C. Mascolo, L. Capra, Wolfgang Emmerich. "Middleware for Mobile Computing (A Survey)". In Advanced Lectures on Networking - Networking 2002 Tutorials, Pisa, Italy. vol 2497 of LNCS, pages 20-58, Springer Verlag. May 2002.

[6] Royer, E. M., Toh, C-K., A Review of Current Routing Protocols for Ad-Hoc Mobile Wireless Networks, IEEE Personal Communications Magazine, pp 46-55, April 1999.

[7] P.Verissimo, V.Cahil, A.Casimiro, K.Cheverst, A.Friday and J.Kaiser. Cortex:Towards supporting autonomous and cooperating sentient entities.In Proc. of European Wireless 2002,Florence,Italy,Feb 2002.