Middleware for Next-Generation Urban Traffic Control

Raymond Cunningham, Anthony Harrington and Vinny Cahill
Department of Computer Science, Trinity College, Dublin, Ireland

Abstract
Next generation Urban Traffic Control (UTC) systems should be flexible in their ability to handle new and improved sensor information that is beginning to become available from cameras, GPS devices, and other sources. They should equally be open to supporting a variety of different traffic management policies, informed by this increasingly available sensor data, to optimise the flow of traffic in an urban environment.

The focus of the UTC-NG project in Trinity College Dublin is on exploring the design of next-generation UTC systems. With the expected increase in the deployment of a variety of different sensor technologies, key to such a design is that it should address this diversity. A consequence of supporting a variety of sensor types, is that the design also needs to accommodate a variety of ways of processing the information provided by these sensors to form an accurate view of the current state of the UTC system and ultimately to implement different traffic control policies. Moreover, as the variety and accuracy of sensor data improves and new sensor fusion techniques are developed, it should be possible to easily exploit this new information. In a similar way, different policies to optimise the movement of traffic through the system may need to be incorporated over time.

This paper describes how the UTC-NG project is addressing these problems through the use of middleware. Middleware is a collection of services that can be used by application developers (in this case the developers of specific UTC systems) to build a particular UTC application. In the case of middleware for next-generation UTC, the middleware will provide services for sensor fusion, reasoning about the system state, and communication embodied in an appropriate programming model and should support a "plug and play" approach to choosing the right services for different systems. A stigmergic model is being used to address the complex interaction between collections of junctions and consequently to allow optimisation of the overall traffic flow in a UTC system. In such a stigmergic model, a junction will follow simple rules, taking in appropriate information from fixed and mobile sensors as well as from neighbouring junctions and in turn informing neighbouring junctions of changes it is making to traffic light phases under its control.

Currently, a prototype implementation of a simulator for the UTC-NG project is being undertaken. The goal of this simulator is to exercise the middleware platform and provide valuable insight into how a number of sensors and sensor fusion techniques can be supported and also aid in the adaptation of the set of simple rules, which enables optimisation of traffic flow, to a particular traffic management policy.

Introduction

The UTC-NG project in Trinity College Dublin is predicated on the possible increase in the type and volume of sensor data that might be available to future UTC systems to inform their decision making. In particular, the UTC-NG project foresees a world in which wireless communication can be used to communicate sensor data, such as their locations obtained from on-board Global Positioning System (GPS) receivers, from individual vehicles to the local UTC system. It is foreseen that the availability of such information should lead to more optimal urban traffic control if it can be exploited by the designs of UTC systems in the future. Moreover, it should be possible to implement a wider range of both local (per-junction) and global (system-wide) policies than currently. For example, the availability of sensor data describing the stress levels of individual drivers, might allow the instantiation of a UTC system that attempts to minimise driver-stress levels across a city. Key to the latter example, is that the design of such a system mirrors the design of a system that attempts to optimise, for example, travel time based on GPS data from vehicles to the extent that it is concerned with applying some optimisation function to the values of sensor readings obtained from vehicles. Both systems might be based on a common design if that design accommodates the use of different sensors and different optimisation functions in different instantiations.

Unfortunately, existing UTC systems (Lowrie 1982; Robertson 1987; Mauro 1991) are designed to utilise a relatively limited range of sensors, such as inductive loops and traffic cameras, to enable the management of traffic in an urban environment. The sensors employed are usually part of a fixed infrastructure that is configured and deployed by the local traffic authority. Moreover, the policies implemented in these systems tend to focus on goals such as maximising the number of vehicles moving through the system or minimising average journey times. Next generation UTC systems should be more flexible in their ability to handle new and improved sensor information that is beginning to become available from cameras, GPS devices, and other sources. They should equally be open to a variety of different sorts of policy informed by the variety of sensor data available.

Providing such a flexible and open design for next-generation UTC systems is clearly a challenging problem. Current approaches to designing UTC systems typically make incremental modifications to existing systems. The UTC-NG project has both the advantage and disadvantage of having to start from the absence of any existing code base. The major advantage is that no restrictions are placed on the design in terms of having to support legacy functionality. The disadvantage is that no existing software infrastructure is available to bootstrap the system.

Middleware based approach

As highlighted in the previous section, the focus of the UTC-NG project is on exploring the design of next-generation UTC systems. With the expected increase in the deployment of a variety of different sensor technologies, key to such a design is that it should address this diversity. A consequence of supporting a variety of sensor types, is that the design also needs

to accommodate a variety of ways of processing the information provided by these sensors to form an accurate view of the current state of the UTC system and ultimately to implement different traffic control policies. Depending on what is being sensed (e.g., cars passing through junctions with inductive loops, the position of every vehicle in the system with on-board GPS, or the stress levels of drivers with on-board sensors) and our confidence in the accuracy and timeliness of the sensed values, different means of fusing these sensor readings together will be appropriate (Abidi and Gonzalez 1992; Klein 2001). Moreover, as the variety and accuracy of sensor data improves and new sensor fusion techniques are developed, it should be possible to exploit them. In a similar way, different policies may need to be incorporated over time.

In addition to the above requirements, it is important to note that systems of the scale of a UTC system are never exactly the same and the design must accommodate a variety of deployment options including the use of different communications technologies.

Technically, these requirements on the design of a next-generation UTC system can be captured as a requirement to support a family of UTC systems. In his seminal work, Parnas (Parnas 1976) defines program families as "sets of programs whose common properties are so extensive that it is advantageous to study the common properties of the programs before analysing individual members". Members of this family may differ in a number of ways, such as the types of sensor and communications technologies supported, the way in which they fuse this sensor data or reason about their resulting view of the system. However, the key observation is that they share a common design that should be configurable to the requirements of specific UTC systems to be developed in the future.

Providing a design for such a family of UTC systems, requires expertise from a number of different disciplines such as computer science, electronic engineering and civil engineering. The focus of the remainder of this report is on the software design.

The software design of a family of UTC systems can be captured as a middleware framework (Cahill 1996). By middleware is meant a collection of services that can be used by application developers (in this case the developers of specific UTC systems) to build a specific application according to a programming model embodied in the middleware. For example, commercially
available middleware (Pitt and McNiff 2001; OMG 2002; Platt 2002) supporting the distributed object programming model provides the services needed to develop applications as collections of software objects distributed over multiple computers. In the case of middleware for next-generation UTC, the middleware will provide services for sensor fusion, reasoning about the system state, and communication embodied in an appropriate programming model and should support a "plug and play" approach to choosing the right services for different systems.

Such middleware will be a key enabler allowing UTC developers to build next-generation UTC systems. Thus, it is important to understand that the intended "users" of the results of this project are the developers of future UTC systems and not, for example, traffic authorities who

may wish to deploy advanced UTC systems in the future. The results of the project should serve both to inform their own design decisions and also to make available a software infrastructure that can be used to build specific UTC systems. Using such middleware should help to address the requirements highlighted in the previous paragraphs such as supporting diverse sensor technologies and enabling different traffic control policies. Two specific research challenges underlie this approach:

- the design of an appropriate programming model for the development of UTC systems to be supported by the middleware; and
- the design of the middleware to allow the inter-operation and interchange of a variety of services providing different approaches to sensor fusion, reasoning and communication.

In the next section we describe a candidate programming model.

Sentient Objects and Stigmergy

The programming model proposed to underpin the middleware framework discussed in the previous section is the Sentient Object model (Biegel and Cahill 2004) since it is designed specifically for the construction of large-scale, proactive applications driven by sensor data of which next-generation UTC systems are representative. A sentient object, illustrated in Figure 1, is an encapsulated component, with its interfaces being sensors and actuators.

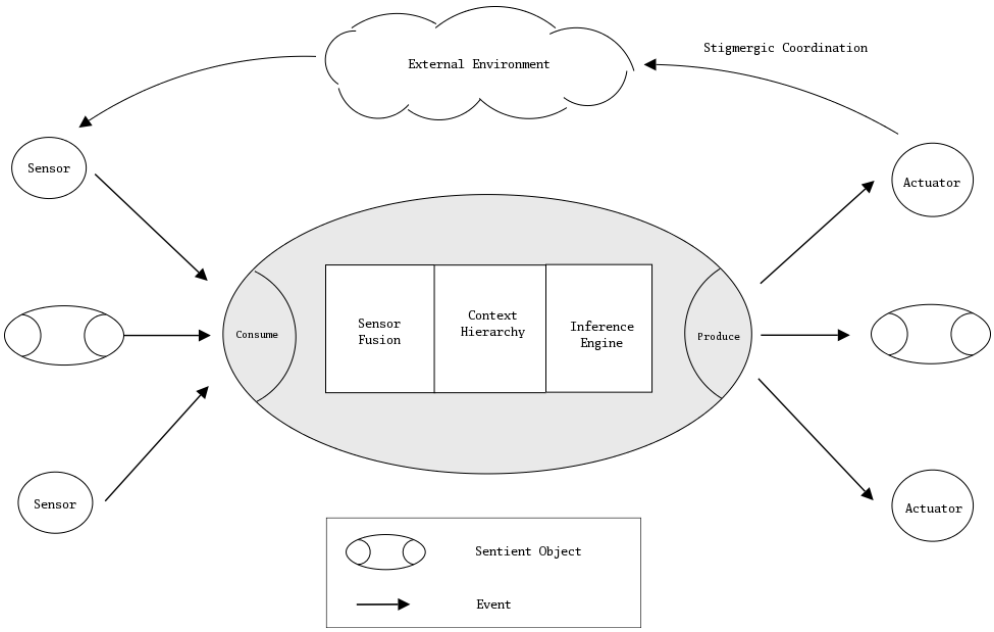


Figure 1 Sentient Object

Sensors in the sentient object model are defined as components that produce software events in reaction to a stimulus detected by some device. These devices include hardware devices,

such as GPS receivers and inductive loop sensors, as well as software devices such as personal information managers (PIMs). Thus, a sensor encapsulates both a physical device, as well as software that provides a higher-level symbolic interpretation of the output of the physical sensor or software component.

An actuator in the sentient object model is defined as a component that consumes software events and reacts by attempting to change the state of the external environment, such as modifying the phases of a traffic light. Actuators in the sentient object model are software abstractions of physical devices or software components that receive events and perform some action.

A sentient object is then defined as a software component that can both consume and produce software events, and lies in a control path between at least one sensor and one actuator. Sentient objects can be either autonomous or cooperative. Co-operating sentient objects communicate with each other and with actuators via an anonymous event-based communication paradigm (Meier and Cahill 2003). Sensors also communicate with sentient objects using this event-based communication paradigm.

As illustrated at the top of Figure 1, there is a feedback path from actuators to sensors through the environment. As actuators change the environment in some way, the effects of this change will, in turn, be detected by sensors sensing the environment. This communication through the environment from actuators to sensors is called stigmergic communication and is inspired by the communication achieved by social insects such as ants (Bonabeau, Dorigo et al. 1999; Kennedy, Eberhart et al. 2001). For example, ants lay down pheromone trails to inform other ants of a path to food with higher concentrations of pheromones indicating a larger number of ants using a particular path. By communicating through the environment and using simple rules, ant colonies in particular and insects in general are capable of complex adaptive behaviour (Bonabeau, Dorigo et al. 1999; Kennedy, Eberhart et al. 2001).

Sensor data is used to inform the particular application about its current context. Based on this current context, the sentient objects then change their environment using appropriate actuators. With respect to the sentient object model, context is defined as any information sensed from the environment that may be used to describe the state of a sentient object. A sentient object using contextual information to fulfill its goals is then said to be context-aware.

As illustrated in Figure 1, a sentient object is composed of three main internal components responsible for sensor fusion, context reasoning, and inference. The sensor fusion component of a sentient object performs sensor fusion in order to manage uncertainty of sensor data and derive higher level context information from a number of data sources. Our current middleware supports only one approach to sensor fusion based on Bayesian Networks. A variety of different techniques, such as Kalman filtering, fuzzy logic etc, already exist to fuse data from a number of sensors. However, making a collection of these techniques available as components that can be used by a sentient object without necessarily affecting other components is a difficult problem. The difficulty arises due to each fusion technique making different underlying assumptions about the particular sensors and their configuration.

As illustrated in Figure 2, the set of possible contexts of a sentient object are represented as a hierarchy, based on the Context-Based Reasoning (CxBR) paradigm (Gonzalez and Ahlers 1999). A context in the context hierarchy represents a particular state of the sentient object. In the context of traffic control, the mission context might represent the overall policy that a sentient object, located at a junction, is trying to implement such as optimising the number of vehicles passing through the junction. Major contexts might represent different periods of operation such as peak and off-peak times or different perceived system states such as free flowing or congested. Sub contexts might represent different junction states such as the presence of heavy traffic on particular approaches.

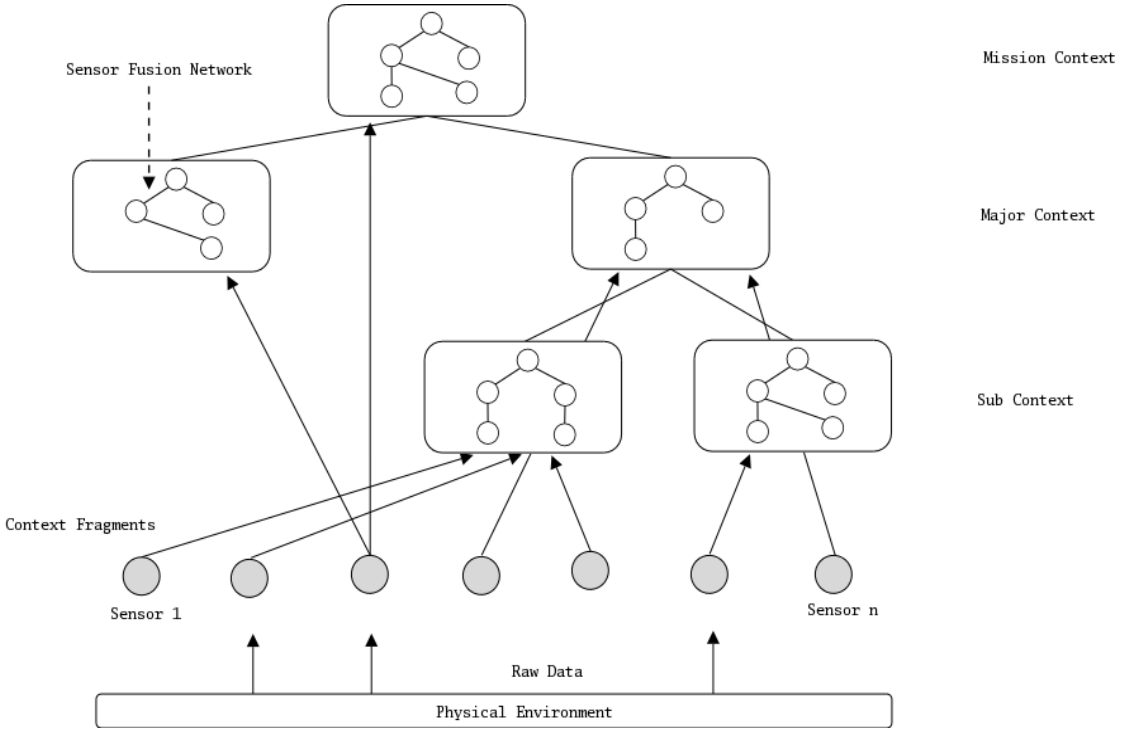


Figure 2 Context Hierarchy

By defining a hierarchy of contexts in which a sentient object may exist and by associating specific actions to be undertaken in each context, a sentient object's behaviour is influenced by its context. In particular, context-aware behaviour in a sentient object is achieved by the sentient object using an inference engine to activate different sets of rules to drive its behaviour in different contexts in order to implement its optimisation strategy. Contexts in the context hierarchy act as filters on both the set of sensors that are relevant and the set of rules that may be applicable at a given time and therefore help to reduce both the sensor fusion and inference problems. As in the case of the sensor fusion component, different inferencing techniques may be needed in different circumstances to implement different optimisation strategies such as reinforcement learning (Barto and Sutton 1998) or particle swarm optimization (Kennedy, Eberhart et al. 2001).

Chosen approach

Most existing UTC systems attempt to optimise the flow of traffic under the control of the particular UTC system. The goal of this optimisation can vary depending on the particular policy of the UTC system. Some UTC systems may want to optimise traffic volumes, travel times, or queuing time or optimise for public transportation versus private transportation.

Tackling the optimisation of traffic flow in a generic way is a difficult task. Existing UTC systems, such as SCATS, typically break down the area under the control of the UTC system into a number of sub-areas (or subsystems) and then optimise the traffic flow in each of these sub-areas. A subsystem typically consists of a collection of junctions. However, the complex interaction between traffic flowing into and out of each of these subsystems may have a detrimental impact on the overall global traffic flow.

Following the sentient object model, the approach being pursued in the UTC-NG project to address this complex interaction between collections of junctions and consequently to allow optimisation of the overall traffic flow in the UTC system is to follow a stigmergic model. In such a stigmergic model, junctions will follow simple rules, taking in appropriate information from fixed and mobile (i.e., vehicle-mounted) sensors as well as from neighbouring junctions and in turn informing neighbouring junctions of changes it is making to traffic light phases under its control.

By each junction following these simple rules, the hypothesis is that global traffic flow can be optimised in a way similar to how ant colonies seek to optimise their search for food. In addition, the approach accommodates local optimisations, such as minimising queuing time when there is no opposing traffic, in a straightforward manner and also allows different rules to be implemented at different junctions.

Related work

Three generations of interconnected traffic signal control systems are identified in (Klein 2001). First generation traffic signal control systems operated by selecting timing plans from pre-engineered and pre-stored plans. These signal plans were developed using classical traffic engineering theory such as (Webster and Cobbe 1966) and produced using off-line software such as TRANSYT (Klein 2001). Second generation UTC systems attempted real-time optimization of traffic flows by fine-tuning phase split times and offsets between intersections. Third generation or adaptive control systems support the online generation and implementation of signal timing parameters derived from real-time sensor data, prediction and optimization techniques.

The advent of adaptive control systems was enabled by the greater use of sensors for vehicle and traffic flow detection. Two popular adaptive UTC systems are the Split, Cycle and Offset Optimization Technique (SCOOT) and the Sydney Coordinated Adaptive Traffic System (SCATS).

SCOOT was developed by the Transport Research Laboratory in the United Kingdom and is used in over 170 towns and cities worldwide (Hunt, Robertson et al. 1981). SCOOT relies on sensor data to identify platoons of vehicles in the street network and, based on a road network model it estimates the arrival time of the platoon at downstream junctions. These platoons are then processed through the linked signal controllers. There are three optimization techniques used in SCOOT (Robertson and Hunt 1983):

1. The Split Optimization technique analyses the current red and green timings to determine whether the stage time should be advanced, reduced or retained. It alters split settings by 4 seconds.
2. The Offset Optimization technique analyses current traffic flows at each node using cyclic flow profiles for each of the links with upstream or downstream nodes. It then assesses whether the existing action time should be advanced, reduced or retained. It alters offset timings by 4 seconds.
3. The Cycle Time Optimization technique operates on a region basis once every five minutes, It identifies the critical node within the region and adjusts the cycle time on this node to maintain a 90% link saturation on each state. Cycle lengths are altered in 4, 8 or 16 second increments.

The SCATS UTC system was developed by the New South Wales Roads and Traffic Authority (Sims 1979). It operates by measuring real-time traffic volumes and flows at intersections and uses this information to adjust cycle times, splits and offsets. When demand is less than system capacity SCATS operates to minimize overall stops and delay. The optimization technique employed maximises the degree of saturation on approaches to junctions. This value represents the ratio of the effectively used green time to the total available green time. When demand approaches system capacity SCATS maximizes throughput and controls queue formation (Sims 1979). SCATS also supports the linking of neighbouring junctions together to create a green-wave to optimize key corridors in a transport network.

Novel control techniques such as evolutionary computation and Artifical Intelligence strategies are being employed to tackle the issue of large scale complex systems control. Hoar et. Al (Hoar, Penner et al. 2002), have investigated the application of swarm intelligence to cooperative signal setting optimization. Their approach is based on the stigmergic model of environment mediated communication to enable inter-vehicle and vehicle-signal controller cooperation. The fitness function evaluated seeks to minimize the average waiting time of all vehicles in the network. Initial tests indicate promising results and increased adaptivity to high rates of change in traffic flow.

Abdulhai et. al (Abdulhai, Pringle et al. 2003), have applied reinforcement learning techniques to the UTC domain. Using an unsupervised learning technique they have demonstrated that an AI approach can at least match the performance of expert-knowledge based pre-timed signal plans without the requirement of a network model or traffic engineering expertise. The system attempts to minimize the total delay incurred by vehicles in all queues on a junction approach.

However, the evaluation of this technique only relate to the operation of a single intersection controller.

Findler (Findler 1999), has applied a learning technique known as harmonization to distributed real-time adaptive traffic control. This work focuses on facilitating co-ordinated omni-directional signal progression across intersections. A theoretical foundation for the harmonization technique is provided and an inductive proof is offered that benefits seen from using harmonization in limited regions extends to global optimization.

The research work summarised above illustrates that new control and communication techniques are being applied to the UTC application area. However, the issue of sensor technology requirements and accuracy assumptions is not directly addressed by any of the authors listed above who assume it is possible to have detailed and accurate knowledge about the application environment. Next generation UTC systems will need to incorporate more detailed models of sensors and their accuracy and support a number of different sensor fusion techniques.

Acknowledgements

The work described in this paper was funded under the Programme for Research in Third Level Institutions (PRTLI) administered by the Higher Education Authority (HEA) of Ireland.

References

Abdulhai, B., R. Pringle, et al. (2003). Reinforcement Learning for True Adaptive Traffic Signal Control. Transportation Engineering. **129**.

Abidi, M. and R. Gonzalez (1992). Data Fusion in Robotics and Machine Intelligence, Academic Press.

Barto, A. G. and R. S. Sutton (1998). Reinforcement Learning, MIT Press.

Biegel, G. and V. Cahill (2004). A Framework for Developing Mobile, Context-aware Applications. Second IEEE International Conference on Pervasive Computing and Communications, Orlando, Florida, IEEE Computer Society.

Bonabeau, E., M. Dorigo, et al. (1999). Swarm Intelligence: From Natural to Artificial Systems, Oxford University Press.

Cahill, V. (1996). On The Architecture of a Family of Object Support Operating Systems. Computer Science. Dublin, Trinity College Dublin. **Ph.D**.

Findler, N. (1999). Harmonization for Omnidirectional Progression in Urban Traffic Control. Computer-Aided Civil and Infrastructure Engineering, Honolulu, Hawaii.

Gonzalez, A. J. and R. Ahlers (1999). "Context-Based Representation of Intelligent Behaviour in Training Simulations." Transactions of the Society for Computer Simulation **15**(4).

Hoar, R., J. Penner, et al. (2002). Evolutionary Swarm Traffic: If Ant Roads had Traffic Lights. Proceedings of the IEEE Conference on Evolutionary Computation, CEC'02, Honolulu, Hawaii.

Hunt, P., R. Robertson, et al. (1981). SCOOT- A Traffic Responsive Method of Coordinating Signals, TRRL Report 1014, HM Stationery Office.

Kennedy, J., R. C. Eberhart, et al. (2001). Swarm Intelligence, Morgan Kaufmann.

Klein, L. A. (2001). Sensor Technologies and Data Requirements for ITS.

Lowrie, P. R. (1982). The Sydney Co-ordinated Adaptive Traffic System - principles, methodology, algorithms. Proceedings of the IEEE International Conference on Road Traffic Signalling.

Mauro, V. (1991). Road Network Control.

Meier, R. and V. Cahill (2003). Exploiting Proximity in Event-Based Middleware for Collaborative Mobile Applications. Proceedings of the 4th IFIP International Conference on Distributed Applications and Interoperable Systems (DAIS'03), Paris, France.

OMG (2002). The Common Object Request Broker: Architecture and Specification, V3.0.

Parnas, D. L. (1976). "On the Design and Development of Program Families." Transactions on Software Engineering **SE-2**(1): 1--9.

Pitt, E. and K. McNiff (2001). Java.RMI: The Remote Method Invocation Guide, Addison Wesley.

Platt, D. S. (2002). Introducing Microsoft.NET, Microsoft Press.

Robertson, D. and P. Hunt (1983). Estimating the Benefits of Signal Co-ordination using TRANSYT or SCOOT. Proceedings of the 53rd Institute of Transportation Engineers (ITE'83), London, United Kingdom.

Robertson, G. D. (1987). Handling Congestion with SCOOT. Traffic Engineering and Control.

Sims, A. (1979). The Sydney Coordinated Adaptive Traffic System. Proceedings of the ASCE Engineering Foundations Conference on Research Priorities in Computer Control of Urban Traffic Systems '79.

Webster, F. and B. Cobbe (1966). Ministry of Transport Road Research Technical Paper No. 56 Traffic Signals, HM Stationery Office.

Middleware for Next Generation Urban Traffic Control

Raymond Cunningham

DSG DISTRIBUTED SYSTEMS GROUP
TRINITY COLLEGE DUBLIN

Overview

- Existing Urban Traffic Control (UTC) approaches
- UTC-NG (Next Generation UTC)
- Middleware
- Status

Existing UTC systems

- Reactive
  - User controlled
- Limited number of sensors used
  - Inductive loops
- Periods of operation
  - Morning peak, evening peak, off peak, etc
- Local optimisation
  - Small collections of junctions are optimised

Next Generation UTC

- Provide flexibility
  - Support different types of sensors
  - Support different ways of reasoning about the current state of the system
  - Support different management policies to optimise the flow of traffic
- Make the incorporation of new technologies easier.
  - New sensors
  - New policies

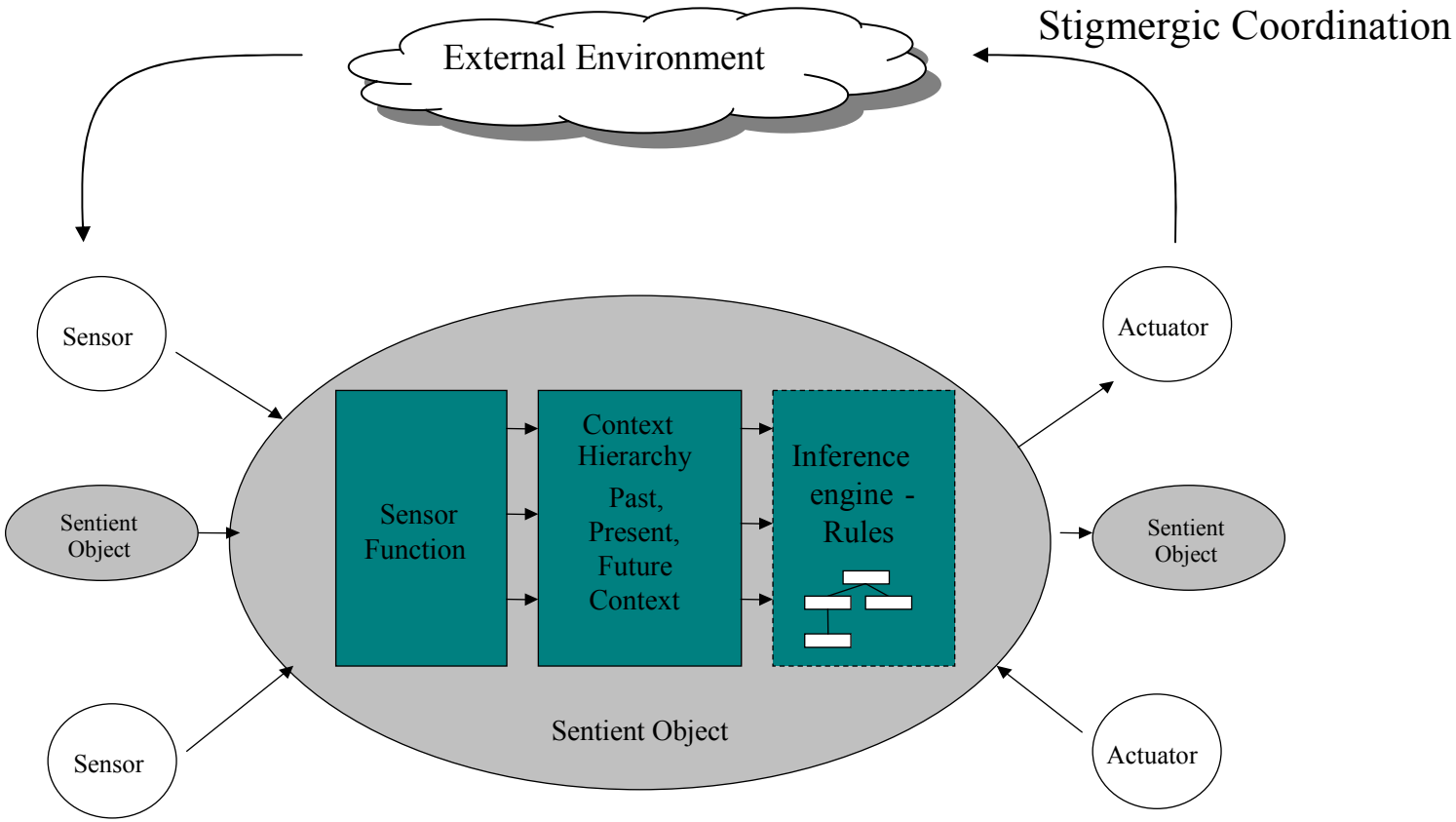
Middleware

- What is it?
  - In general, middleware is a collection of (software) services that can be used by application developers to help them design and build applications.
- Middleware for Next Generation UTC
  - Support a number of sensors & sensor fusion techniques
  - Support a number of optimisation techniques
  - Support a number of different traffic management policies

Middleware for UTC-NG

- Global optimisation
  - Emergent behaviour
    - Simple rules leading to complex adaptive behaviour
  - Stigmergy / ant colony optimisation
- Next generation UTC systems should learn from past experience
  - Use historical data
- There is no silver bullet!!
  - Different techniques should be inter-changeable

Sentient Objects for UTC-NG



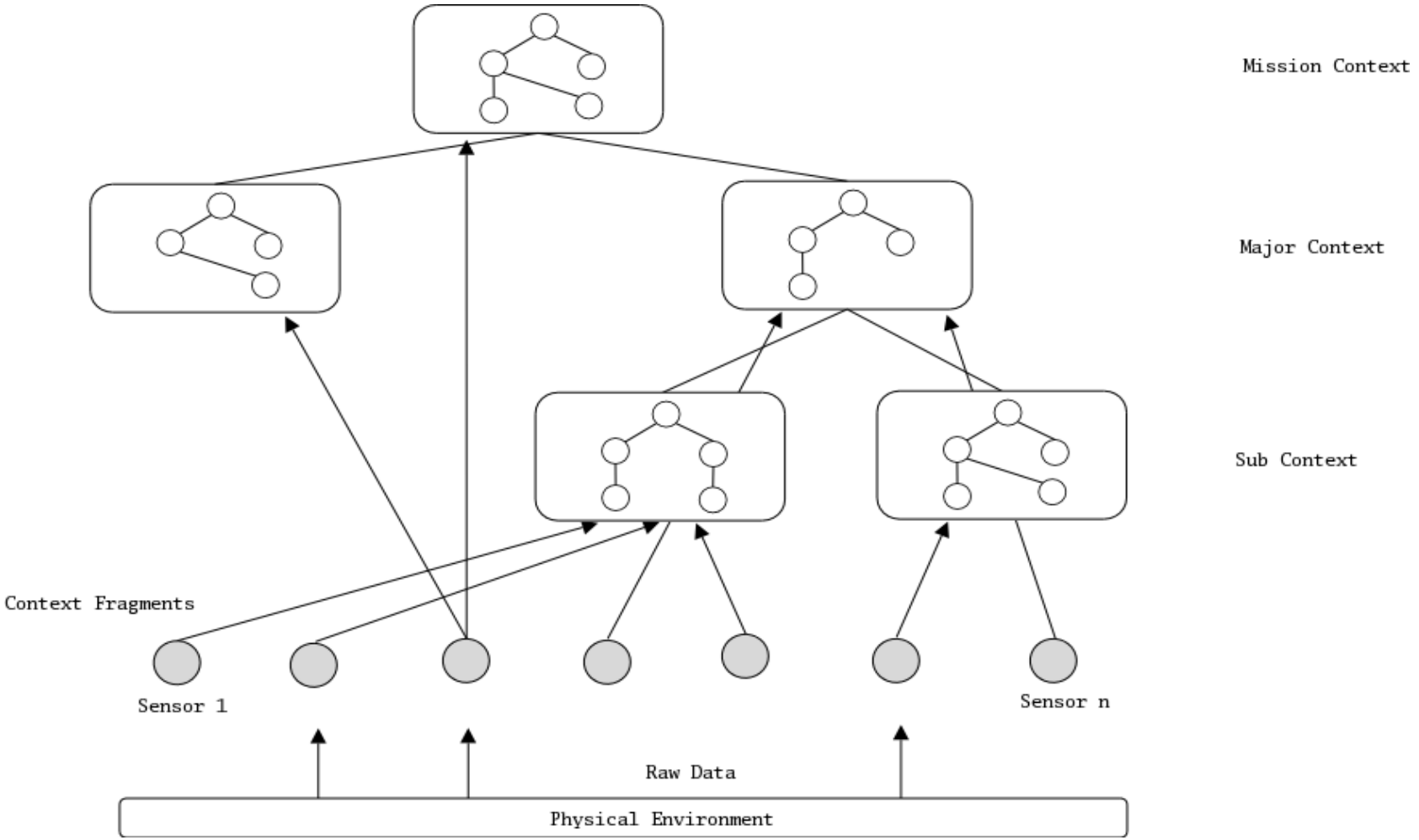
Sentient Objects

- Each traffic light controller is modeled as a sentient object
- Sentient objects communicate using an anonymous event based communication paradigm.
- This allows a variety of different sensors to communicate with a sentient object
  - By using the same type of events
- New types of sensors can be added easily without the need to change the internals of a sentient object.

Sensor Fusion

- Different types of sensors
  - Infra-red, cameras, inductive loops, GPS
- Different types of sensor fusion techniques
  - Kalman filtering
  - Bayesian networks
  - Fuzzy logic
- Sensors providing the same information should be inter-changeable
  - E.g. Inductive loops vs GPS + telemetry sensors

Context Hierarchy



Inference Engine

- Use present and past context to ensure informed decision making
- Inference techniques
  - Belief theory
  - Neural networks
  - Reinforcement learning
- Rules are then executed to act on the environment
  - Change traffic light phases
  - Give priority to public transport

Status

- Initial design for tackling UTC-NG, supporting:
  - Different sensors
  - Different fusion techniques
  - Different transport policy goals
- Prototype implementation of sentient object model in a traffic simulator.
  - Model of road network consists of 248 traffic lights, 1000 junctions, 3000 links
    - Dublin city
  - Model of realistic traffic volumes

Acknowledgements

- This work is supported by TRIP, a multidisciplinary transport research centre in Trinity College Dublin funded by the Programme for Research in Third Level Institutions (PRTLI) administered by the Higher Education Authority (HEA) of Ireland.