Project IST-2000-26031

CO-operating Real-time sentient objects: Architecture and Experimental evaluation

CORTEX

Final Report

CORTEX deliverable D14

Version 1.0

December 31, 2004

Revisions

Rev.	Date	Comment
0.1	15/12/2004	First draft
1.0	31/21/2004	Final version

Editor

António Casimiro, University of Lisboa

Contributors

All partners

Address

Faculdade de Ciências da Universidade de Lisboa

Bloco C6, Campo Grande

1749-016 Lisboa

Portugal

Executive Summary

This deliverable constitutes the final report of the CORTEX project. The key objective of CORTEX was to explore the fundamental theoretical and engineering issues necessary to support the use of sentient objects to construct large-scale proactive applications and thereby to validate the use of sentient objects as a viable approach to the construction of such applications. CORTEX defined a programming model to support the development of applications constructed from mobile sentient objects, and proposed an interaction model for co-operating sentient objects. It also defined an architecture reflecting the heterogeneous structure and performance of the networks used to support the programming model and developed a demonstrator that allowed to assess the proposed solutions. This report introduces the project and its partners, summarizes the key results of CORTEX, and provides a brief discussion of the plans for dissemination and exploitation of these results. The detailed description of the work developed in CORTEX is presented in its technical deliverables, all of which are available from the CORTEX web site at `cortex.di.fc.ul.pt`.

Table of contents

Chapter 1: Introduction

A new class of application that operates independently of direct human control is starting to emerge. Key characteristics of applications of this class include sentience, autonomy, large scale, time and safety criticality, geographical dispersion, mobility and evolution. It is our belief that the development of such applications is highlighting the shortcomings of current communication architectures and middleware infrastructures. In particular, they do not adequately support advanced dynamic interaction models, e.g., in the field of autonomous agents, distributed AI, and mobile co-operating entities.

The goal of the CORTEX project was to bridge the gap between the requirements being put on system support by these advances, and the shortcomings of current architectures and middleware models. Therefore, CORTEX explored fundamental theoretical and engineering issues necessary to support the use of sentient objects as mean to achieve this goal. One of the major contributions of CORTEX was the comprehensive study of the sentient object paradigm as a fundamental paradigm to construct proactive and cooperative applications with real-time and reliability requirements. The work in CORTEX has progressed along three main directions:

- the definition of the CORTEX programming model, taking into account the (sentient) nature and the (non-functional) QoS requirements of the envisaged applications;
- the design of an interaction model centred around an anonymous event-based communication abstraction, having the notion of event channels as a basic middleware abstraction;
- the definition of the CORTEX architecture, comprising the underlying network structure and the essential services and protocols needed to support the CORTEX interaction model.

The final part of the project was fundamentally devoted to the construction of a CORTEX demonstrator, based on a co-operating sentient vehicles scenario in which it was possible to highlight several CORTEX achievements and assess the applicability of the proposed approaches.

Chapter 2: Description of consortium and roles of participants

The CORTEX consortium brought together significant expertise from the distributed computing, mobile computing, middleware, fault tolerance and real-time communities. Although the full set of CORTEX partners had not worked together before, the existence of previous collaborations and mutual knowledge between various of the partners was important to the establishment of good working relationships, to the prosecution of joint work and to the success of the project.

Lisboa: The Navigators group at the Faculty of Sciences of the University of Lisboa (FCUL) was the coordinator contracting for CORTEX. In the project, the group addressed the architecture and middleware development, namely the aspects concerning timeliness and reliability in large-scale environments. The group has a vast experience with communication protocols in distributed systems, in particular with the group communication paradigm. Previous work with relevance to CORTEX included the investigation of how timeliness and predictability requirements can be handled in environments with uncertain synchrony properties and the development of models of intermediate synchrony with support for dynamic adaptation techniques. In this context, tolerance to timing faults has been a key structuring factor in the work that has been developed.

Dublin: Research in the Distributed Systems Group in the Department of Computer Science primarily addresses language and middleware support for distributed computing, especially distributed object computing. Previous research experience of relevance to CORTEX included work on the use of reflection to support dynamic adaptation of software, the development of an event-based communication model oriented towards development of large-scale distributed applications and support for mobile computing. Within CORTEX, TCD contributed to the development of the paradigms and supporting middleware for communication and co-operation between sentient objects building on its experience with event-based programming and adaptive middleware. TCD also contributed to the design of real-time reliable communication protocols suitable for use in wireless networks.

Lancaster: The Distributed Multimedia Research Group at Lancaster has a strong record in distributed systems dating back to 1983. In the recent years, the research of the group has grown to encompass distributed systems for mobile environments and context-sensitive computing. DMRG has an established track record of engineering distributed systems, reflective and adaptive middleware, systems software and applications to target challenging problem domains such as the Utilities Industry, Emergency Services and the Mountain Rescue service. Within CORTEX, DMRG contributed to the development of the sentient object platform and interaction models. Most significantly however, DMRG led the development of the CORTEX demonstrator, employing the technologies developed by the project.

Ulm: The real-time group at the Faculty of Computer Science of the University of Ulm has been active in the area of object-oriented real-time systems and real-time communication. The main focus of work addresses the conflict between the substantial resource constraints often found in embedded applications and the demand for high-level computational models which are needed to enable the cost effective design of such systems. Under this aspect, the group had experience in the development of time-bounded reliable communication protocols for controller area networks. Additionally, it had experience in building and integrating special purpose hardware components in a testbed for intelligent sensors and actuators. The main contributions of the group in CORTEX were in the definition of the interaction model, the architecture and the development of middleware to support the programming of applications based on sentient objects.

Chapter 3: Key scientific results and achievements

This chapter summarizes the main results and achievements of CORTEX, organized according to the three main lines of work and corresponding work-packages. The CORTEX deliverables contain full technical details of all this work, and only a brief summary is made here.

3.1 Programming model

- Definition of the most important properties that are expected to be exhibited by the applications addressed by CORTEX and introduction of a scheme based on three vectors – autonomy, consistency and cooperation – which serve to classifying candidate applications considered in CORTEX.
- Detailed description and analysis of a number of application scenarios, including a feasibility discussion with respect to building demonstrator, outlining the scope of each scenario, how it has been classified with respect to the other scenarios and, of course, the specific requirements that it places on the CORTEX technology.
- Definition of an event model that allows in general to specify functional and non-functional attributes of an event. The focus has been on the definition of temporal attributes for events that can be used to control the dissemination. The programming interface is integrated in the publisher/subscriber model of communication.
- Definition of APIs to be incorporated in the programming model, supporting different real-time classes for event channels and supporting dependable adaptation of timeliness constraints.
- Definition of an architecture that provides a way to structure applications around a component-based object model, allowing object composition to be influenced or constrained by the component's physical structure. The architecture postulates a generic events model easing composition and structural (body-environment) awareness, further enriching the basic CORTEX object-oriented programming model based on anonymous event-based communication.
- Definition of the final CORTEX programming model, addressing sentient object and event-based programming model, QoS specification, context based reasoning, and sensor fusion for reducing uncertainty in context acquisition.

3.2 Interaction model

- Definition of an appropriate interaction model, based on the examination of existing schemes to compare their suitability with respect to the specific requirements in CORTEX, i.e. a large number of interacting entities, a high degree of cooperation and a dynamically evolving system. It consists on a distributed publisher/subscriber model based on the notion of event channels. Event channels support an event-based interaction model and enforce anonymity, an important property to enable dynamic extensibility of the system. Communication abstractions and interaction model are defined in a suitable way for the CORTEX WAN-of-CANs network architecture. The contradicting requirements between anonymity of communication on the one side, and predictability on the other side is tackled by elaborating the concept of predictable event channels.

- Investigation into approaches for context and environment awareness for CORTEX, focussing on context based reasoning and a probabilistic sensor fusion architecture.

- Definition of an event-based communication middleware for mobile, ad-hoc, wireless networks and the definition of a model for real-time, event-based communication in mobile, ad-hoc, wireless networks. Most previous work on real-time event-based communication has assumed infrastructure-based networks with stationary components. In contrast, ad-hoc wireless networks comprise sets of mobile nodes connected by wireless links that form arbitrary wireless network topologies without the use of any centralized access point. Such highly mobile, dynamic networks do not satisfy the design assumptions for previous real-time event-based communication. The conceptual model for real-time event-based communication in mobile ad hoc wireless networks was the first to address the issue of achieving timeliness and reliability for real-time event-based communication in dynamic mobile ad hoc wireless networks.

- Application of event channels in a mobile ad-hoc environment, addressing the problem of mapping event channels to different underlying wired and wireless networks and defining means to constrain dissemination and increase predictability of operation.

- Design of a filter mechanism, which evaluates the event attributes. In a system composed from performance constraint devices, it constitutes a good compromise between expressiveness and predictability. Solutions to constraint event dissemination are based on subject, attribute and content filtering, which allow, in particular, filtering on tiny smart devices with substantial resource constraints. The concepts have been realized on PCs and currently are ported to smart components powered by 8-Bit micro-controllers.

- Investigations in adequate temporal models to describe the interaction via event channels. Particularly, integration of event interaction through physical channels of the environment has been explored. This has been discussed in the context of a compositional sentient object approach which particularly takes into account smart components as mechanical/hardware/software constructs.

- Research on coordination of sentient objects through temporally consistent real-time images. The issue of temporal consistency plays a fundamental role when dealing with time-value entities, in particular when considering interactions between sentient objects and the environment (or through the environment). This work was done along with the investigations in adequate temporal models to describe the interaction via event channels.

- Proposal of stigmergy, as a possible coordination mechanism for large networks of real time sentient objects.

3.3 Architecture and protocols

- Definition of a prototype of a WAN-of-CAN structure based on CAN-bus for the CAN level and on a TCP/IP network for the WAN level. This structure is aimed to support the publisher/subscriber protocol defined in the CORTEX interaction model. Furthermore, the concept of an event channel has been adopted as the basic communication model on which the architecture is based.
- Definition of a new protocol, named TBMAC (for Time-Bounded Medium Access Control), specifically designed for wireless ad hoc networks.
- Definition of a Timely Computing Base, which can be seen as a special architectural component serving the whole system and providing crucial time related services. This TCB component will play an important role in the definition of the communication services and protocols, to be done in the next period.
- Definition of basic services and protocols that materialize the CORTEX node architecture, which are necessary to support the operation of sentient applications.
- Definition of proof-of-concept prototypes aimed at demonstrating some specific achievements and partners' contributions, which were later integrated and used as input for the construction of the final demonstrator. In particular, the following proof-of-concept prototypes were defined and presented:
  - Cooperating Autonomous Robots Demo
  - Adaptation and Fail-safety in Cooperating Cars Demo
  - Sentient Vehicle Demo
  - Sentient Room Demo
  - Demo of Framework for Testing Safety-Critical Sentient Applications
- Definition of a time-bounded medium access protocol (TBMAC) for multi-hop wireless ad hoc networks. More specifically, studies of the inaccessibility of TBMAC and a detailed description of the messages used in this protocol have been presented.
- Definition of a QoS framework supporting real-time event-based communication in mobile environments.
- Definition of a resource management framework for the CORTEX architecture. The framework encompasses both a task model and a resource model.
- Definition of protocols to implement TCB services, in particular to implement the distributed timing failure detection service and the duration measurement service.
- Definition of the ATES framework (Adaptable Timed Event Service) based on CORTEX event-channels, to allow event-based communication with adaptable and timed events.
- Development of a reliable multicast protocol for ad hoc environments with high mobility. More specifically, an adaptive flooding algorithm based on retransmission probabilities has been defined. The algorithm tries to maximise the reliability of the multicast without sacrificing too many resources.

- Definition of the layered COSMIC architecture (CO-operating SMart devICes), which provides different real-time event channel classes and supports an API for a real-time publisher/subscriber interaction. COSMIC includes the definition of an abstract network layer over CAN, which allows specifying different real-time requirements for message transmission. The abstract network layer also supports automatic configuration of devices when connected. A prototype system was implemented, includes nodes based on PCs running under Linux/RT-Linux and various micro-controller families. Furthermore, the full real-time functionality of COSMIC event channels has been realized.

- Development of an infrastructure to support interoperability between CANs and a wireless network (802.11) based on the publisher/subscriber architecture.

- Development of a protocol CCPR (Content- and Cell-Based Predictable Protocol) for content-based routing and predictable access over a wireless network. The protocol is based on assumptions derived from TBMAC.

- Integration of the COSMIC middleware with the architectural framework of GEAR. The architectural construct of body and environment has been exploited as a means to describe encapsulation and recursive system construction. For the event-based communication this results in a clear identification of intrinsic knowledge about the system which can be exploited to reduce the context information which an event has to carry in its attributes. Additionally, it allows specifying QoS complementary to particular network zones.

Chapter 4: Characteristics of CORTEX Applications

With advances in sensor-based computing and mobile communication, researchers have started to explore ubiquitous computing (UbiComp) systems that aim to have computing devices embedded literally everywhere, while making them disappear into the physical environment (e.g., in our cars, buildings, soft furnishings, appliances, clothing etc.). Novel applications are possible in these environments, but many of the scenarios we can envision require elements to operate independently of direct human control. Among the most popular examples of these kinds of applications are based around intelligent vehicles, traffic management systems, and smart buildings or working/living environments. We believe that there are a number of special characteristics that differentiate these classes of application from traditional computing applications, such as:

- **Sentience**: These applications are context-aware, i.e. have the ability to perceive the state of the surrounding environment, through the fusion and interpretation of information from possibly diverse sensors. Sentience is realised by building these applications using sentient object paradigm: sentient objects can receive software events via a variety of software sensors or other sentient objects to construct their view of the world by using the sensor fusion module inside sentient object.
- **Autonomy**: Components of these applications will be capable of acting in a decentralised fashion, based solely on the acquisition of information from the environment and on their own knowledge. Autonomy is achieved by using the sentient object paradigm: the inference engine inside sentient objects give them a certain level of "intelligence" to allow them to act autonomously based upon the acquisition of information from the environment.
- **Proactivity**: These applications are able to act in anticipation of future goals or problems without direct human intervention. They should have a certain degree of intelligence, and be able to decide what action to take from gathered sensor data. Proactivity is achieved by using the sentient object paradigm, particularly the inference engine inside the sentient object.
- **Cooperation**: The constituent sentient objects of an application are able to interact between each other to achieve common goals. Different sentient objects communicate with each other via some anonymous and asynchronous event channels, which suits well for the mobile ad-hoc network environments.
- **Decentralisation**: There is no single central server that does intensive computation for the clients. Typical applications consist of components that might be scattered across geographical regions, e.g., street, buildings, cities, countries, and continents.
- **Large scale**: These applications operate in a pervasive environment whereby a large number of hardware and software components are typically involved. The sentient vehicle demonstrator by TCD shows a large number of sentient objects cooperating with each other to reach some common goals.
- **Adaptivity**: These applications will have to cope with changing conditions during their lifetimes. Not only must the applications be designed to evolve, but their underlying support must be adaptable as well. This is achieved by using the component oriented approach to engineer the sentient objects. By using the reflective componentised middleware called OpenCOM from Lancaster University, we can dynamically change the configuration of the sentient objects, e.g., plug-out the current ultrasonic sensor fusion component and plug-in another one.
- **Time and Safety Criticality**: These applications interact with physical environments and are required to provide real-time services to human users. It is important to

provide real-time guarantees and dependability assurance through some system or middleware modules, e.g., resource management and configuration, timing failure detection and Quality of Service (QoS) management. Timely Based Computing (TCB) from University of Lisbon has been integrated into the final demonstrator to achieve time and safety criticality, e.g., the sentient vehicles can do emergency stop if the network coverage stability (reported by TCB) is below a certain threshold. The resource management module creates Virtual Task Machines (VTMs) to handle different types of events from the event channel, so that it is possible to process more important events first than the unimportant ones.

- **Mobility**: The components of these applications often need to move between different geographical locations, which might involve roaming across different networks. It is important to remain continuous operation while roaming.
- **Evolution**: Regards the case when the growth of the application scales well. In addition, support is provided for both application extensibility and the inclusion of technology advances. By using the component oriented approach for engineering the sentient object paradigm, this property can be achieved by the ability of replacing or adding new software components.

These characteristics make it extremely challenging for application designers and system engineers to design and implement such systems and applications. Some key research challenges we aimed to address in the CORTEX project were as follows:

- **Context-awareness**: We follow the sentient object paradigm to handle inputs from diverse sources, e.g., different sensors or other sentient objects. Uncertainty is a major problem in sensing the environment due to the inherent limitations of sensors with respect to accuracy and precision. This has led to a crucial requirement for our middleware that it provides uncertainty management for software components whose actions are based on environmental perception. Additionally, "intelligent" software components that reason based on context are required in order to make sentient objects autonomous and proactive.
- **QoS management and fail safety**: Due to the real-time nature of the CORTEX applications, the middleware needs to take into account the provision of incremental real-time and reliability guarantees. QoS properties need to be expressed as a metric of predictability in terms of timeliness and reliability. For distributed objects coordinating in uncertain environments, the timing bounds for distributed actions could be violated because of the timing failures. This requires a reliable timing failure detection service for distributed operations.
- **Communication model**: Traditional communication models, such as client-server and the RPC paradigm, are not well suited to mobile ad-hoc environment, because there is no fixed infrastructure to host centralised services. Since disconnections are common in the wireless communication environment, the communication paradigm should be decoupled and asynchronous. Moreover, in novel applications with mobile or context-based elements, the scope of information dissemination is dynamically determined by spatial parameters. For example, in the cooperating cars scenario, one might wish to limit dissemination to those vehicles directly affected by an obstacle on the road, and the information is only valid in a restricted geographical area.
- **Routing in mobile ad-hoc environment**: In mobile ad-hoc networks, the senders and receivers move constantly so that the network topology frequently changes. This poses a challenge for routing packets in such dynamic environments. Multicast protocols based on proactive and reactive ad-hoc routing, using shared state kept in the forms of routes and adjacent information, is useful in environments with low node mobility. However, this shared state and topology information can quickly become

outdated in the highly mobile environments. Hence, it requires a new type of routing protocol in highly dynamic ad-hoc network environments.

In the context of the CORTEX project, we explored the fundamental issues relating to the support of such applications, including defining programming and interaction model based on sentient object paradigm, specifying the system architecture of CORTEX applications, and engineering a componentised CORTEX middleware for this domain as well as a cooperative sentient vehicle demonstrator (which has been delivered on the final review in Lisbon).

Chapter 5: Lessons learned

In this chapter, we briefly review the key findings of the evaluation conducted in the final part of the project and the engineering lessons learnt from our experience. This text was extracted from the evaluation report of cortex (Deliverable D13), which can be found in the CORTEX web site.

5.1 CORTEX Paradigm

5.1.1 Sentient Object Programming Model

The sentient object programming model provides a systematic approach to application development, reducing the complexity of the design task. By developing small proof-of concept applications and the final demonstrator, we argue that the sentient object model as a useful paradigm for developing dependable, real-time applications, because:

- At the design stage, software components in such applications can be nicely modeled as software sensors, actuators, or sentient object by following the CORTEX programming model.
- Cooperation between components or application can be easily achieved by making use of the anonymous and asynchronous event channel specified in the programming model;
- Certainty and accuracy of the sensor data can be increased by employing the sensor fusion inside the sentient object, which receives sensor data through some event channel, perform fusion algorithms, and derive higher level context information from multi-modal data sources.
- Inference engine inside the sentient object enables the application to make autonomous/proactive decisions based on the environmental parameters getting from the sensors and their domain knowledge specified in rulesets.

5.1.2 Generic Event Architecture

We have developed the Generic Event Architecture GEAR which provides a uniform model integrating interactions through both the environment and the network. We have demonstrated through the implementation of two proof-of-concept demonstrators (COSMIC and STEAM), that:

- Sensors and actuators can be viewed consistently as sentient objects producing and consuming (generic) events.
- That the model substantially eases application design, by offering an abstraction layer which hides lower level network details, allowing transparent communication between different networks.
- That dynamic integration of new devices in possible at the event-layer, with reconfiguration being handled automatically, enabling the plug-and-play of smart devices.
- That such a system is suitable for tiny systems as low cost micro-controllers with severe performance and memory constraints.
- That bounded propagation of event notifications using proximity-based filtering can be used to reduce the number of events delivered to an application.

5.2 Engineering Lessons

5.2.1 Sensor fusion

By engineering our proof of concept demonstrators, we have also found that:

- Using Bayesian networks to fuse data from multiple sensors overcomes the scalability problems when increasing the number of variables in the system, by exploiting conditional independencies in the data.
- Also, that Bayesian networks allow us to fuse multi-modal sensor data, i.e. data originating from a variety of sensors, in different formats and at different frequencies.
- However, we have also found that large amounts of data must be gathered before such networks may be constructed, and that networks can be extremely subjective, with no guarantee that correct causal dependencies are identified and correctly encoded in the network.

5.2.2 Resource management

We were able to demonstrate that the resource model and task model offer adequate support for resource management over both fine and coarse-grained interactions. However, further work is required in investigating how such techniques might expand across platoons of sentient objects, especially in spontaneous environments featuring ad-hoc interactions.

5.2.3 Integrative Demonstrator

In integrating the contributions from different CORTEX partners into the sentient vehicle demonstrator, we can summarize some of our key experiences as the follows:

- The sentient object model has proved to be a good programming abstraction for the development of real-time, cooperative, context-aware applications, particularly because of its intrinsic support for decoupled event channel communication and context-awareness. Significantly, the demonstrator shows it can be used to handle both context adaptation, e.g., responding to the traffic light signals, and QoS adaptation, e.g., reacting when network coverage stability is low, in a self-consistent and uniform way.
- Our component model and CFs offered us benefits in terms of flexible configuration and reconfiguration at run-time, supporting the both context and QoS adaptive applications. For example, we can dynamically change the “event filtering” component’s underlying communication mechanism to flexibly select between a probabilistic ad-hoc multicast protocol, an IP multicast based protocol and an optimized local (shared memory based) group communication protocol.
- The use of an inference engine and rule based programming makes it easy to do high level adaptation, and react in a uniform way to changes in context and QoS. For example, we can perform actuations, to swap steering rules depending on the current road condition, or adapt the configuration of the components within the demonstrator or supporting middleware.
- The sentient object model also simplifies the management of non-functional concerns such as timeliness and reliability. For example, the data from TCB is distributed using the anonymous communication paradigm, and it in turn goes through the inference engine to derive high-level network coverage stability and trigger actuation events being published to the car speed control channel. This uniform approach allows QoS to become part of context descriptions.

Appendix 1: Deliverables

The following list of CORTEX technical deliverables is organized by work-package. All of these deliverables are publicly available from the CORTEX web site, cortex.di.fc.ul.pt.

WP1 - Programming Model

Definition of application scenarios

Preliminary definition of CORTEX programming model

Final definition of CORTEX programming model

WP2 - Interaction Model

Preliminary definition of CORTEX interaction model

Final definition of CORTEX interaction model

WP3 - System Architecture

Preliminary definition of CORTEX system architecture

Preliminary specification of basic services and protocols

Proof-of-concept prototypes

Final definition of CORTEX system architecture

Final specification of basic services and protocols

WP4 - Demonstrator

Analysis and design of application scenarios

Demonstration of application scenarios

Evaluation report

Appendix 2: Publications

The following list of CORTEX-related publications is organized according to publication type (conference papers and journal papers) and year of publication.

Conference and workshop papers

2004

J. Kaiser and E. Nett, "**Resolving the Trade-Offs in Designing QoS Communication Services for Control Applications on CAN**", Workshop on Architectures for Cooperative Embedded Real-Time Systems (WACERTS'04), in conjunction with the 25th IEEE International Real-Time Systems Symposium (RTSS04), December 5-8, 2004, Lisbon, Portugal.

P. Martins, A. Casimiro and P. Veríssimo, "**A Generic Temporal Consistency Model for Distributed Control Systems**", Workshop on Architectures for Cooperative Embedded Real-Time Systems (WACERTS'04), in conjunction with the 25th IEEE International Real-Time Systems Symposium (RTSS04), December 5-8, 2004, Lisbon, Portugal.

J. Kaiser, "**Event Channels, an Integration Concept for Predictable Interaction in Embedded Heterogeneous Networks**", International Workshop on Dependable Embedded Systems, October 17, 2004, in conjunction with the 23rd Symposium on Reliable Distributed Systems (SRDS 2004), Florianopolis, Brazil.

C.-F. Sørensen, M. Wu, T. Sivaharan, G. Blair, P. Okanda, A. Friday and H. Duran-Limon, "**Context-Aware Middleware for Applications in Mobile Ad Hoc Environments**", ACM/IFIP/USENIX International Middleware conference 2nd Workshop on Middleware for Pervasive and Ad-Hoc Computing, Toronto, Canada, 18th-22nd October, 2004.

J. Kaiser, C. Brudna, and C. Mitidieri, "**Implementing Real-Time Event Channels on CAN-Bus**", 5th IEEE International Workshop on Factory Communication Systems, Vienna, Austria, September 22nd-24th, 2004.

T. Sivaharan, G. Blair, A. Friday, M. Wu, H. Duran-Limon, P. Okanda and C.-F. Sørensen, "**Cooperating Sentient Vehicles for Next Generation Automobiles**", ACM/USENIX MobiSys 2004 International Workshop on Applications of Mobile Embedded Systems, Boston, USA, June 6, 2004.

M. Wu, A. Friday, G. Blair, T. Sivaharan, P. Okanda, H. Duran-Limon, C.-F. Sørensen, G. Biegel, and R. Meier, "**Novel Component Middleware for Building Dependable Sentient Computing Applications**", ECOOP'04 workshop on Component-oriented approaches to context-aware systems, Oslo, Norway, June 14, 2004. Affiliated with 18th European Conference on Object-Oriented Programming (ECOOP 2004), 14-18 June 2004, Oslo.

P. Martins, P. Sousa, A. Casimiro, P. Veríssimo, "**Dependable Adaptive Real-Time Applications in Wormhole-based Systems**". In Proceedings of the International Conference of Dependable Systems and Networks (ICDSN04), Florence, Italy, June 2004.

R. Meier, J. Kaiser, B. Hughes, and V. Cahill, "**An Event Model for Real-Time Systems in Mobile Environments**", in Proceedings of the Second IEEE Workshop on Software Technologies for Future Embedded and Ubiquitous Systems (ISORC/WSTFEUS '04). Vienna, Austria: IEEE Computer Society, May 2004, pp. 29-34.

C. Mitidieri, J. Kaiser and C.E. Pereira, "**An event filtering scheme for distributed real-time embedded systems**", Proc. of the 11th IFAC Symposium on Information Control Problems in Manufacturing (INCOM'04). Salvador, Brazil. April, 2004

A. Casimiro, J. Kaiser, P. Veríssimo, "**An Architectural Framework and a Middleware for Cooperating Smart Components**", Proc. Of the 2004 Computer Frontiers Conference, Iscia, Italy, April 14-16, 2004

G. Biegel and V. Cahill, "**A Framework for Developing Mobile, Context-aware Applications**", in Proceedings of 2nd IEEE Conference on Pervasive Computing and Communications, (Percom) 2004, Orlando, FL, March 14-17, 2004.

2003

B. Hughes and V. Cahill, "**Achieving Real-time Guarantees in Mobile Ad Hoc Wireless Networks**", Work in progress session, IEEE Real Time Systems Symposium, Cancun, Mexico, December, 2003.

P. Veríssimo, J. Kaiser, A. Casimiro, "**An architecture to support interaction via generic events**", Work-in-Progress session, IEEE Real-Time Systems Symposium. Cancun, Mexico, December 2003

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

H. Duran-Limon, G. Blair, A. Friday, T. Sivaharan, and G. Samartzidis, "**A Resource and QoS Management Framework for a Real-Time Event System in Mobile Ad Hoc Environments**", Ninth IEEE International Workshop on Object-oriented Real-time Dependable Systems (WORDS 2003), Capri Island, Italy. October 2003.

J. Kaiser, "**An Event Model for the Predictable Interaction of Smart Devices**", Workshop on Dependable Embedded Systems, in conjunction with the IEEE 22nd International Symposium on Reliable Distributed Systems, Florence Italy, October 6-8, 2003

J. Kaiser, C. Liu, "**Content and Cell based Predictive Routing (CCPR) protocol for Mobile ad hoc networks**", 5th International Workshop for Advanced Parallel Processing Technologies (APPT'03), Xiamen, China, Sept. 2003. Lecture Notes in Computer Science 2834: Advanced Parallel Processing Technologies, Springer, pp. 488-495

J. Kaiser, C. Mitidieri, C. Bruna, C.E. Pereira, "**COSMIC: A middleware for event-based interaction on CAN**", ETFA, Emerging Technologies and Factory Automation, Lisbon, Portugal, September 2003

B. Hughes and V. Cahill, "**Towards Real-time Event-based Communication in Mobile Ad Hoc Wireless Networks**" in Proceedings of 2nd International Workshop on Real-Time LANS in the Internet Age 2003 (ECRTS/RTLIA03), Porto, Portugal, July 2003, pp. 77-80.

C. Mitidieri, J. Kaiser, "**Attribute based filtering for embedded systems**", 2nd International Workshop on Distributed Event-Based Systems (DEBS'03), San Diego, California, June 2003

J. Kaiser, C. Brudna, and C. Mitidieri, "**A Real-Time Event Channel Model for the CAN-Bus**," Workshop on Parallel and Distributed Real-Time Systems, held in conjunction with the International Parallel and Distributed Processing Symposium IPDPS, 2003, April 22-26, 2003 in Nice, France.

V. Cahill, and G. Biegel, "**Sentient Objects for Context Aware Business Process Management**", in Proceedings of the SAP Innovation Congress. Miami, Florida, USA, 2003.

R. Meier and V. Cahill, "**Exploiting Proximity in Event-Based Middleware for Collaborative Mobile Applications**", in Proceedings of the 4th IFIP International Conference on Distributed Applications and Interoperable Systems (DAIS'03), LNCS 2893**.** Paris, France: Springer-Verlag, 2003, pp. 285-296.

P. Veríssimo and A. Casimiro, "**Event-Driven Support of Real-Time Sentient Objects**", in Proceedings of the 8th IEEE International Workshop on Object-oriented Real-time Dependable Systems, Guadalajara, Mexico, January 2003

2002

P. Martins and P. Veríssimo, "**The Timely Computing Base and its Future Trends**", in Proceedings of the 7th CaberNet Radicals Workshop, Bertinoro, Italy, Oct 2002

P. Sousa and P. Veríssimo, "**Towards a cooperating autonomous car**", in Proceedings of the 7th CaberNet Radicals Workshop, Bertinoro, Italy, Oct 2002

J. Kaiser and T. Fries, "**ROCON – A Virtual Construction Kit, Visualization Tool and Remote Control System for Mechatronic Devices**," IEEE International Conference on Mechatronics and Machine Vision in Practice 2002, Chiang Mai, Thailand, September10-12, 2002.

J. Kaiser and C. Brudna, "**A Publisher/Subscriber Architecture Supporting Interoperability of the CAN-Bus and the Internet**", 2002 IEEE Int. Workshop on Factory Communication Systems, Västeras, Sweden, August 28-30, 2002.

A. Casimiro and P. Veríssimo, "**Generic Timing Fault Tolerance using a Timely Computing Base**", in Proceedings of the 2002 International Conference on Dependable Systems and Networks, Washington D.C., USA, June 2002

R. Meier and V. Cahill, "**Taxonomy of Distributed Event-Based Programming Systems**," in Proceedings of the International Workshop on Distributed Event-Based Systems (ICDCS/DEBS'02). Vienna, Austria, 2002.

R. Meier and V. Cahill, "**STEAM: Event-Based Middleware for Wireless Ad Hoc Networks**," in Proceedings of the International Workshop on Distributed Event-Based Systems (ICDCS/DEBS'02). Vienna, Austria, 2002.

N. Reijers, R. Cunningham, R. Meier, B. Hughes, G. Gaertner, and V. Cahill, "**Using Group Communication to Support Mobile Augmented Reality Applications**," in Proceedings of the 5th IEEE International Symposium on Object-oriented Real-time distributed Computing (ISORC 2002). Washington DC, USA, 2002.

R. Cunningham and V. Cahill, "**Time Bounded Medium Access Control for Ad Hoc Networks**", in Proceedings of the Second ACM International Workshop on Principles of Mobile Computing (POMC'02). Toulouse, France: ACM Press, 2002, pp. 1-8.

A. Fitzpatrick, G. Biegel, S. Clarke, and V. Cahill, "**Towards a Sentient Object Model**", presented at Workshop on Engineering Context-Aware Object Oriented Systems and Environments (OOPSLA/ECOOSE'02), Seattle, Washington, USA, 2002.

Byun H. E. and K. Cheverst, "**Harnessing Context to Support Proactive Behaviours**", Workshop in ECAI2002 on AI in Mobile Systems, 2002, Lyon, France.

P. Veríssimo, V. Cahill, A. Casimiro, K. Cheverst, A. Friday and J. Kaiser, "**CORTEX: Towards Supporting Autonomous and Cooperating Sentient Entities**", Proceedings of European Wireless 2002, Florence, Italy, February 2002.

Hector A. Duran-Limon and G. S. Blair, "**Reconfiguration of Resources in Middleware**", Seventh IEEE International Workshop on Object-oriented Real-time Dependable Systems (WORDS 2002), San Diego, CA. January 2002.

2001

A. Casimiro, P. Martins, P. Veríssimo and L. Rodrigues, "**Measuring Distributed Durations with Stable Errors**", Proceedings of the 22nd IEEE Real-Time Systems Symposium, London, UK, December 2001.

A. Casimiro and P. Veríssimo, "**Using the Timely Computing Base for Dependable QoS Adaptation**", Proceedings of the 20th IEEE Symposium on Reliable Distributed Systems, New Orleans, USA, October 2001.

N. Gura, A. Held, J. Kaiser, "**Proactive Services in a Distributed Traffic Telematics Application**", Informatik 2001: Joint Annual Conference of the GI/OCG, Workshop on Mobile Communication over Wireless LAN: Research and Application, Sept. 25-28, 2001.

J. Kaiser, P. Schaeffer, "**ICU – A smart optical sensor for direct robot control**", IEEE International Conference on Mechatronics and Machine Vision in Practice 2001, Hong Kong, China, August 27-29, 2001.

A. Casimiro and M. Correia, "**Recent Advances on the Timely Computing Base Model**", Fast Abstract at the International Conference on Dependable Systems and Networks, Göteborg, Sweden, June 2001

C.E.Pereira, L.B. Becker, C. Villela, C. Mitidieri, J. Kaiser, "**On Evaluating Interaction and Communication Schemes for Automation Applications based on Real-Time Distributed Objects**", ISORC 2001: IEEE 4th International Symposium on Object-Oriented Real-time distributed Computing, Magdeburg, Germany, May 2001.

M. O. Killijian, R. Cunningham, R. Meier, L. Mazare, and V. Cahill, "**Towards Group Communication for Mobile Participants**", in Proceedings of Principles of Mobile Computing (POMC'2001). Newport, Rhode Island, USA, 2001, pp. 75-82.

R. Meier, M.-O. Killijian, R. Cunningham, and V. Cahill, "**Towards Proximity Group Communication**", presented at Workshop on Middleware for Mobile Computing (IFIP/ACM Middleware 2001), Heidelberg, Germany, 2001.

Journal papers

2004

J. Kaiser, C. Brudna and C. Mitidieri, "**COSMIC: a real-time event-based middleware for the CAN-Bus**", Journal of Systems and Software (JSS). To appear.

G. Gaertner and V. Cahill, "**Understanding Link Quality in 802.11 Mobile Ad Hoc Networks**", IEEE Internet Computing, Volume 8 (Number 1), 2004, pp 55-60.

Hector A. Duran-Limon, G. S. Blair, and G. Coulson, "**A Survey of Adaptive Resource Management in Middleware",** IEEE Distributed Systems Online Journal. 2004.

2003

G. Biegel and V. Cahill, "**Sentient Objects: Towards Middleware for Mobile Context-Aware Applications**", ERCIM News No. 54, July 2003.

Hector A. Duran-Limon and G. S. Blair, "**Resource Reconfiguration in Adaptive Multimedia Middleware**", submitted to ACM Multimedia Systems. 2003.

2002

P. Veríssimo and A. Casimiro, "**The Timely Computing Base Model and Architecture**", IEEE Transactions on Computers - Special Section on Asynchronous Real-Time Systems, vol. 51, n. 8, Aug 2002

R. Meier, "**Communication Paradigms for Mobile Computing**", ACM SIGMOBILE Mobile Computing and Communications Review (MC2R), vol. 6, 2002.