Event Channels, an Integration Concept for Predictable Interaction in Embedded Heterogeneous Networks

Jörg Kaiser
University of Ulm, Germany
kaiser@informatik.uni-ulm.de

Abstract

The paper describes an event-based system to enable seamless communication over multiple heterogeneous networks. On the one side, it is desirable to have an addressing and routing mechanism that provides transparent network access to all connected nodes, on the other hand, when dealing with predictable systems, the different quality of service in different networks cannot be hidden. The approach introduces events and event channels to allow the uniform access to information throughout the network but also enables a programmer to explicitly define and control these communication relations.

1. Introduction

One of the challenging problems in the area of embedded systems is the integration of components that are connected to a collection of heterogeneous networks. This will be the standard situation in future applications. Heterogeneity already is a problem in a single car, where designers now have to provide interoperability between TTP [11], FlexRay [2], CAN [3], TT-CAN [7], LIN [LIN00], and others automotive networks. In the classical field for predictable communication, the industrial automation, a considerable number of field-busses and backbones have to exchange information. Recently, mobile and wireless communication is emerging adding another building block to heterogeneity in this field. In mobile applications we have to support interoperability between internal sensor networks embodied in a robot, a vehicle or in wearable components and an infrastructure or the direct interaction with other similar components. The exploitation of floating car data or team robotics are examples of such scenarios.

There are two important dimensions of an integration concept. The first is related to the definition of information which is moved around in such an environment. Usually, the semantics of a message is defined implicitly by the programs which send and receive it. This requires an a priori knowledge and agreement about how to interpret the message content. The message passing system does include any information about this content or the context in which the message has been generated. In point-to-point communication relations or in fixed address-based groups this works fine because it is easy to infer the content, at least the structural information, of a message from the sender's address. In a heterogeneous environment with different addressing structures where messages are routed via gateways, this is not obvious. Some networks like CAN or LIN don't provide sender or receiver addresses. Instead, the type of message is carried in the message identifier. Therefore, we would have to provide address information in the payload of the message. Another problem arises with the context, in which a message was generated. Often it is more important to know where a message was generated than which entity generated this message. It may be difficult in a large dynamic network with mobile entities always to relate a sender address implicitly with a location. Therefore, a higher level abstraction is desirable to provide some form of self description for messages which comprise the type of content and the context of creation.

The second dimension of integration is related to the quality of dissemination. Let us consider that we attached an end-to-end deadline to a message. How can this deadline be guaranteed in a heterogeneous set of networks? To provide a single QoS-mechanism for all networks may be as unrealistic as it is undesirable. If it could be done at all, it only would be possible with severe bandwidth penalties or the least predictable network would determine the QoS in the entire system. Moreover, if we consider the safety critical part of the internal network of a robot or a car, we want to achieve a predictability and responsiveness which is many orders of magnitudes larger than in the non-critical parts or in the wireless communication which allows cooperation between vehicles. To cope with these various requirements, integration needs a higher abstraction level to express the individual demands of a communication relation. Secondly, to guarantee a certain end-to-end quality of dissemination, we need to reserve the adequate resources statically or dynamically before communication can be started.

To cope with the problems of integration in heterogeneous networks under reliability and timeliness requirements, we introduced an event-based system architecture and implemented the concept in the COSMIC (COoperating SMart devICes) middleware [9].

2. Abstractions in the COSMIC Middleware

The COSMIC (COoperating SMart devICes) middleware supports the abstractions of events and event channels. An event describes an individual occurrence and may be triggered by an observation of the environment or generated by the system itself. Such an observation can be characterized by a location and a time of occurrence. Additionally, because the value of the observed event may change over time, we have to model this as a time-value entity [12]. The time-value relation provides the necessary basis to decide about the temporal validity of an event. Events may be spontaneously or periodically disseminated containing a current state variable or a previous change of state. Thus, there is no synchrony property directly associated with an event and also nothing is said about the way how state is updated by an event as it is proposed in the terms event-triggered and time-triggered. Events are disseminated in a publisher/subscriber style [17], [6], which is particularly suitable because it supports spontaneous, content- or subject-based communication and does not create any artificial control dependencies between producers of information and the consumers [10]. These properties are also known as space-decoupling and flow-decoupling [6]. Differently from simple messages, an event includes the context in which it has been generated and quality attributes defining requirements for the dissemination. This is particularly important in an open environment where an event may travel over multiple networks and will be used dynamically by varying applications. An event instance is specified as:

event := <subject, context_attributeList,
quality_attributeList, contents> .

A subject defines the type of event. It supports anonymous communication and is used to route an event. In a system requiring dynamic interaction between components which are not known a priori, a content-based characterization of information is crucial. The identifier has to specify what kind of information the event carries rather than which component is the sender or receiver. What is important is that in addition to a content related addressing scheme, the spatial and temporal context of events must be known. This information is defined in the event attributes and is complementary to the event contents. An example of such an event description, specifying a distance sensor is given below:

distance_event:= <UID, rel_pos., timestamp,
max_rate, temp_validity, distance>

The context describe the environment in which the event has been generated, e.g. a location, an operational mode or a time of occurrence. In the examples above this is the relative position of the sensor inside a vehicle (rel_pos) and a timestamp. The quality attributes specify the temporal properties in terms of an validity interval (temp_validity). The validity interval defines the point in time after which an event becomes temporally inconsistent [20]. The temporal validity in many respects is similar to an end-to-end deadline. However, usually a deadline is an engineering artefact which is used for scheduling while the temporal validity is a general property of a time-value entity. In an environment where a deadline cannot be enforced, a consumer of an event eventually must decide whether the event still is temporally consistent, i.e. represents a valid time-value entity. Consider a distance event class which is used in obstacle avoidance and by a navigational system to build a map. Obviously, for the obstacle avoidance component, the temporal validity is much shorter than for the navigational system which may benefit from a reading even when the vehicle already crashed into some obstacle. It is also important, that the temporal validity in this case is related to an error bound which is dependent on the speed of the vehicle. Thus assuming a certain error bound, the temporal validity may change for every individual event occurrence.

The second problem is to disseminate an event in time. The system has to provide the necessary resources to ensure timely delivery. The dissemination properties are specified in the form of event channels. Publishers and subscribers interact via an unidirectional event channel by pushing events in the channel and receiving notifications from a channel. Event channels provide a certain guaranteed degree of synchrony and reliable delivery. When a publisher wants to send an event, it first has to announce this publication and specify the properties of the event channel. With the announcement, it provides the subject, the class of the real-time event channel and some maximum rates or periods at which an event will be published. COSMIC then sets up the necessary local data structures and resolves the UID in the channel specification. When successfully set up, it is guaranteed that the necessary local resources are available to provide the channel properties. An event channel is defined by:

event_channel := <subject, quality_attributeList, handlers >.

The subject determines the event types which may be issued to the channel. In contrast to the attributes of an event which describe the properties of a single individual occurrence of an event, the attributes of the event channel abstract the properties of the underlying communication network and dissemination scheme. These attributes include latency specifications, dissemination constraints and reliability parameters. Currently, we support quality attributes of event channels in a CAN-Bus environment represented by three explicit synchrony classes. The quality attributes of an individual event are used to define the specific settings of the quality parameters of the channel, within the defined synchrony class of the channel.

Hard real-time event channels (HRTEC) provide rigorous timeliness guarantees, guaranteed maximum latency and low period- and latency jitter. Additionally, it is possible to define an omission degree to increase robustness against transient faults. HRTEC are synchronous channels based on a TDMA reservation scheme. They require dedicated operating system and communication resources to prevent any interference with lower real-time classes. An event disseminated through a HRTC always is delivered at the deadline. Thus, all jitter during transmission is masked out by the middleware. This is an important difference to most schemes which try to avoid jitter on the network layer [1, 2, 7, 11] and therefore exhibit less flexibility to reuse allocated but unused bandwidth. An example of the specification of a hard real-time distance channel specification is given below:

distance_channel:= <UID, hard real-time,
reaction_time, omission degree, exc_h>

In addition to the subject and the channel class, the reaction time defines the interval in which TDMA slots have to be allocated. The omission degree specifies the number of retransmissions. An exeption handler entry is provided to handle unanticipated fault situations, e.g. if a message has not successfully be sent in a certain time slot. It allows the application specific, local last moment reaction to a safety critical situation. Soft real-time channels (SRTC) are scheduled according to EDF. The deadline is derived form the temporal validity specification in an event.

Soft real-time messages are queued, transmitted and delivered in a best effort way. In normal load situations, deadlines are met. If inter-arrival times of messages are known, a load analysis can be performed but is not included in COSMIC. However, SRTCs provide awareness if an event is late. SRTCs are in between the fully predictable HRTCs and non real-time channels where no temporal properties are available to define any priority. Soft real-time events are delivered as fast as possible when they are received. Therefore, latency may vary due to load conditions. An analysis of this behaviour can be found in [9].

The concept of event channels is not new, rather it already found its way to standards like the CORBA notification service [16]. A real-time variant first has been introduced by Harrison, Levi and Schmidt [8]. However, these event channels hardly reflect the quality properties of the underlying network structure. Rather, a central notification server is used to manage event dissemination to realize the respective decoupling between publishers and subscribers. The COSMIC middleware is completely distributed and provides awareness about the properties of the network connection. Our goal is to handle the temporal specifications of an event channel eventually as a <bound, coverage> pair [4] orthogonal to the more technical questions of how to achieve a certain synchrony property of the dissemination infrastructure.

3. Confining the dissemination of events in a heterogeneous network

In a network that relies on subject-based routing, it is an important question how to describe events by attributes and how to constrain event dissemination. It is clear that the event layer should be universal, i.e. potentially every component can talk to every other component using events [19]. However, when striving for a form of self-describing events by providing context attributes, an overhead is inevitable. The amount of information that is needed is dependent on the knowledge of the environment which we can assume for an application. E.g. when disseminating information inside a robot we can assume that less context information is needed compared to disseminate this information to a team of robots. Therefore, we need to structure the network not only according to the various quality of service demands and properties but also according to zones in which we can assume an intrinsic common knowledge about the events which we do not have to include during dissemination.

We propose a set of structuring concepts which meet the requirements of different abstraction levels. These abstractions have been developed during the CORTEX project [18]. On the highest level, such an abstraction is provided by the notion of body and environment [19]. On the network level, the notions of zones and the concept of a WAN-of-CANs structure have been introduced to model diversity. A zone is on the level of an abstract network, describing a certain interaction mechanism and a QoS while a CAN is related to the physical network layer. The notion of body and environment is derived from the recursively defined component-based object model [5, 19]. A body is similar to a cell membrane and represents a collection of components in a well defined operational context as well as a quality of service container for the components inside. On the network level, it may be associated with a certain quality zone [14] or a CAN [18]. A zone defines the dissemination quality which can be expected by the cooperating objects. In the above example, a vehicle may be the body composed from the respective lower level components (sensors and actuators) which are connected by the internal network. Correspondingly, a platoon of vehicles can be seen as a body including a collection of cooperating vehicles.

The notion of body and environment is more high level than the zone or the WAN-of-CAN model and allows to reason not only about the technical problems of a bridge or a gateway between different networks. It defines the amount of information which is necessary to interpret an event properly. Usually, the program defines the semantics of a message. There is an a priori understanding between the sender and the receiver of what a certain message contains and how it is handled. In the COSMIC event-based system, the basic semantics of an event is defined by its type represented as the subject and thus carried with the event itself. Hence, the meaning of the event is associated on the system level instead on the application level only. This helps to provide more dynamic interactions. However, the type of the event alone is not enough. Consider a team of robots which uses the distance sensors of all robots to build a map of the environment including obstacles. Events of the type “distance” are disseminated which carry the readings of the distance sensors of an autonomous vehicle. It is necessary to know more about the context in which this event has been generated, e.g. where the sensor is mounted in the vehicle (this actually is more important than to know the sensor ID). While inside the vehicle the relative position of the sensor is sufficient, additional information is needed if such an event is used outside the vehicle. Then the identity of the vehicle or the geographic coordinates may be necessary to exploit this type of information. If we need to include the geographical position of the event generation, a reasonably precise GPS coordinate would require about 6-8 bytes, the entire payload of a CAN message. The same is true if absolute (e.g. NTP [13]) time stamps are included. The notion of body and environment allows to describe these relations. Inside a body, a certain context and also common failure modes or a certain quality of event dissemination can be assumed and exploited by the respective applications. There is no need to assign the position of a vehicle to internal events. Likewise, the reliability and the timeliness of the event transmission can be assumed to be similar for all events inside the body. Thus, the context and quality attributes can be adapted to the precise needs omitting an unnecessary over-specification of event attributes. The ability to specify body/environment relations recursively allows to define the respective gateways and filters. For a smart sensor, the vehicle represents the environment, for the vehicle, the platoon represents the environment and so on.

From the viewpoint of QoS, the interactions across the body/environment boundaries and over multiple zones show that the approach to define a single mechanism to provide quality measures for the interaction is not appropriate. Instead, a high level construct for interaction across boundaries is needed which allows to specify the quality of dissemination and exploits the knowledge about body and environment to assess the feasibility of quality constraints. The notion of an event channel represents this construct in our architecture. It disseminates events and allows the network independent specification of quality attributes. These attributes must be mapped to the respective properties of the underlying network structure.

4. Mapping event channels to a heterogeneous network

Mapping the abstractions of the event layer directly to the underlying network is a tough challenge because the usual abstractions on the basic network layer are low-level frames. Hence, this layer does not match the requirements like group communication, subject-based addressing or the QoS specifications defined for channels. layer. Therefore, an abstract network layer is introduced enriching the properties of the raw with additional properties and communication services such as reliable broadcast, group communication, and temporal guarantees for message dissemination. This separation of concerns supports modularity and allows for an easier adaptation of the event layer to networks with widely differing characteristics. Protocols which create abstract network properties which support temporal constraints on CAN are e.g. FTT-CAN [1], TT-CAN [6], or the server-based network scheduling approach of Nolte et al. [15]. They allow to map the different event channel classes available in COSMIC to their respective QoS mechanisms. The same applies to TTP and FlexRay which also support hard real-time and more relaxed real-time conditions. In COSMIC we provided a mapping to he CAN-Bus [9] and also to wireless networks [14]. Of course, the properties of the channel is closely related to the respective abstract network layer capabilities.The QoS properties of the event layer in general depend on what the abstract network layer can provide. Thus, it may not always be possible to support highly safety critical hard real-time event channels if the abstract network layer cannot provide the required guarantees. Therefore, the event channels supported by the event system are dependent on the zones in which the respective protocols can be provided.

5. Conclusion

Integrating multiple heterogeneous networks in a large scale application scenario becomes an important issue of future embedded systems. Events and event channels support the fine grained description of communication spanning multiple physical networks. Events are related to a type of information by their subjects. Event attributes allow to describe the context of event production and the temporal validity of events. To avoid unnecessary overhead in the attributes, the notion of body and environment is introduced. Assumptions about common knowledge can be exploited inside a body and thus need not to be included in the event's context attributes.

Event channels represent an abstraction of the underlying communication network. They represent the date structure to specify individual channels attributes like synchrony class, latencies and omission degree. Additionally, event channels are created before communication, thus are used for resource reservations. The functional separation of the event layer and the abstract network layer allows to map the event layer on various abstract network protocols. Currently, COSMIC specifically supports periodic hard real-time channels, sporadic and aperiodic soft real-time channels and best effort non real-time channels..

From the application programmers point of view, the event-based approach greatly eases application design and integration of new hard- and software components in a heterogeneous network environment. All interactions can be defined in terms of events which are disseminated via event channels. Currently COSMIC is used in a team robotic application in which networked smart sensors and actuators embodied in autonomous vehicles cooperate with the sensors and actuators in similar vehicles communication via event channels bridging the local CANs and a wireless network

6. Acknowledgements

The work described in this paper was partly supported by the FET programme of the Commission of the EU under research contract IST-2000-26031 (CORTEX).

7. References

[1] L. Almeida, J. A. Fonseca, and P. Fonseca, "A Flexible Time-Triggered Communication System Based on the Controller Area Network: Experimental Results", *Proceedings of the International Conference on Fieldbus Technology (FeT'99)*, Magdeburg, Germany, 1999.

[2] Belschner, R., J. Berwanger, et al., “FlexRay requirements specification Version 2.0.2“, April 2002.

[3] Bosch GmbH, “CAN Specification version 2..0”, September 1991.

[4] A. Casimiro and P. Veríssimo, “Using the Timely Computing Base for dependable QoS adaptation”, *20th IEEE Symposium on Reliable Distributed Systems*, New Orleans, USA, 2001.

[5] Antonio Casimiro, Jörg Kaiser, Paulo Verissimo: "An Architectural Framework and a Middleware for Cooperating Smart Components", *Proc. Of the 2004 Computer Frontiers Conference*, April 14-16, Iscia, Italy, 2004.

[6] P. Eugster, P. Felber, R. Guerraoui, and A.-M. Kermarrec, “The many faces of publish/subscribe,” EPFL, Lausanne, Switzerland, *Tech. Rep. DSC ID:200104*, 2001,

[7] Führer, T., Müller, B., Dieterle, W., Hartwich, F., Hugel, R., Walther, M., “Time triggered communication on CAN”, http://www.cancia.org/can/ttcan/fuehrer.pdf , 2000.

[8] T. Harrison, D. Levine, and D. Schmidt, "The Design and Performance of a Real-Time CORBA Event Service," in *Proceedings of the 1997 Conference on Object- Oriented Programming Systems, Languages and Applications (OOPSLA'97)*. Atlanta, Georgia, USA: ACM Press, 1997, pp. 184-200.

[9] J. Kaiser, C. Bruna, C. Mitidieri, 2004. "Implementing Real-Time Event Channels on CAN-Bus*",Proc. of the 2004 IEEE International Workshop on Factory Communication Systems*, Vienna, Austria, 2004

[10] J. Kaiser and M. Mock*,* 1999*,* Implementing the Real-Time Publisher/Subscriber Model on the Controller Area Network (CAN)," *in Proceedings of the 2nd International Symposium on Object-oriented Real-time distributed Computing (ISORC99).* Saint-Malo, France

[11] Kopetz, H., Grünsteidl, G., 1992. *TTP - A time-triggered protocol for fault-tolerant real-time systems.* Tech. Rep. 12/92, Inst. für Techn. Informatik, Techn. University of Vienna.

[12] H. Kopetz and P. Verissimo, Real-time and Dependability Concepts. In S. J. Mullender, editor, *Distributed Systems, 2nd Edition*, ACM-Press, 1993

[13] D. Mills : „Internet Time Synchronization “The Network Time Protocol”*, IEEE TC on Communications*, 39(10), , 1991.

[14] René Meier, Jörg Kaiser, Barbara Hughes, Cristiano Brudna, Vinny Cahill, 2004, An Event Model for Real-Time Systems in Mobile Environments, *WSTFEUS, Workshop on Software Technologies for Future Embedded & Ubiquitous Computing Systems, Vienna, held in in conjuction with the: 7th IEEE International Symposium on Object-oriented Real-time distributed Computing, ISORC04)*

[15] Nolte T., Nolin M., Hansson, H. Server-Based Real-Time Communication on CAN. *Proceedings of 11th IFAC Symposium on Information Control Problems in Manufacturing,* Salvador, Brazil, 2004.

[16] OMG. 2002, Corbaservices: Common object services specification - notification service specification, version 1.0.1. *Object Management Group,* August 2002.

[17] B. Oki, M. Pfluegl, A. Seigel, D. Skeen. 1993, The Information Bus® An Architecture for Extensible Distributed Systems. 14th ACM Symposium on Operating System Principles, Ashville, NC, USA

[18] Veríssimo, P., Cahill, V., Casimiro, A., Cheverst, K., Friday, A., Kaiser, J., Cortex: Towards supporting autonomous and cooperating sentient entities. *In: European Wireless Conference.* Florence, Italy, 2002.

[19] Veríssimo, P., Casimiro, A., Jan. 2003. Event-driven support of real-time sentient objects. In: Proceedings of the 8th IEEE International Workshop on Objectoriented Real-time Dependable Systems. Guadalajara, Mexico.

[20] Paulo Veríssimo, Luís Rodrigues, “Distributed Systems for System Architects*”, Kluwer Academic Publishers,* Boston, January 2001