Introducing Aspect-oriented Space Containers for Efficient Publish/Subscribe Scenarios in Intelligent Transportation Systems

Eva Kühn, Richard Mordinyi,
Laszlo Keszthelyi, Christian Schreiber
Space-based Computing Group
Vienna University of Technology
Favoritenstr. 9-11, 1040 Vienna, Austria
{ek,rm,lk,cs}@complang.tuwien.ac.at

Sandford Bessler, Slobodanka Tomic
Telecommunications Research Centre Vienna (FTW)
Donau-City 1, 1210 Vienna, Austria
{bessler, tomic}@ftw.at

Abstract

The publish/subscribe paradigm is a common concept for delivering events from information producers to consumers in a decoupled manner. Some approaches allow durable subscriptions or the transportation of events even to mobile subscribers in a dynamic network infrastructure. However, in the safety-critical telematics durable delivery of events is not sufficient enough. Short network connectivity time and small bandwidth limit the number and size of events to be transmitted hence relevant information needed for safety-critical decision making may not be timely delivered.

In this paper we propose the integration of publish/ subscribe systems and Aspect-oriented Space Containers (ASC) distributed via Distributed Hash Tables (DHT) in the network. The approach allows storage, manipulation, pre-processing, and prioritization of messages sent to mobile peers during bursts of connectivity.

The benefits of the proposed approach are a) less complex application logic due to the processing capabilities of Space Containers, and b) increased efficiency due to delivery of essential messages only aggregated and processed while mobile peers are not connected. We describe the architecture of the proposed approach and explain its benefits by means of an industry use case.

1 Introduction

The publish/subscribe paradigm [8] is a common and largely recognized concept for delivering messages in an anonymous decoupled fashion from publishers to subscribers. There are also concepts trying to ensure durability or the correct order of events [13], even in case of mobile peers or a completely dynamic network [10]. However, there are application domains, like safety-critical telematics, in which the durable delivery of subscribed messages, in other words the guaranteed delivery with "exactly once" semantics" may be considered a precondition for operation, due to jurisdictional reasons, but is not adequate at all. Among others, a durable notification service has to store all events a peer has subscribed for while the subscriber is offline. Once the peer is reachable again, the saved events have to be delivered to the associated subscriber. This means that the peer would receive a large amount of data that it has to process locally in order to extract relevant information. However, in scenarios from Intelligent Transportation Systems (ITS), mobile peers (vehicles) have only a few seconds of connectivity and very limited bandwidth [17]. This may cause several problems: the reconnecting peer should receive all stored events which may have very different importance for the user or be even stale, but due to the limited bandwidth and connectivity window only a very few messages can be forwarded to the peer creating a kind of back-pressure in the system. Furthermore, due to the small connectivity window, there is a possibility that essential information, such as safety-critical ghost driver warnings, cannot be transmitted to the peer. If such messages are not forwarded to the peer on time humans lives may be jeopardized. Therefore, the safety risk grows with the amount of irrelevant or even incorrect information delivered instead of important life-saving information.

In this paper[1] we propose the concept of Aspect-oriented Space Containers (ASC) for linking pub/sub systems and mobile peers with short connectivity time. The so called Space Container [12] is a customizable storage component for efficient storage and retrieval of structured, spatial-temporal data distributed in a fault-tolerant manner [11]

[1] This work has been supported by the Complex Systems Design & Engineering Lab, and by the Austrian Government and the City of Vienna within the competence center program COMET.

978-1-4244-4985-9/09/$25.00 ©2009 IEEE

by means of Distributed Hash Tables (DHT) [15]. Aspects are components with customizable application logic executed either before or after the operation on the Space Container for events processing. The combination of a pub/sub medium and ASC allow the mobile peer to inset a Space Container in the network which then acts as an intermediate-subscriber for events in the pub/sub medium, and processes the delivered events on behalf of that mobile peer.

2 Related Work

The pub/sub paradigm defines two types of clients: *publishers* generating events, and *subscribers* receiving notifications of events they have previously subscribed for. This type of messaging paradigm allows a strong decoupling of publishers and subscribers in time and space. Furthermore, it enables asynchronous and anonymous communication between publishers and subscribers [8]. There are two types of pub/sub systems: *topic*-based and *content*-based [8]. Hybrid pub/sub systems like Hermes [14], SIENA [5] or Rebecca [18] support both types of subscription. Furthermore, the pub/sub system architecture can be further classified into *client-server* and *peer-to-peer*. In a client-server architecture, like Gryphon [3], publishers and subscribers are clients which are connected to a network of servers. Generally, the servers temporarily store the events generated by publisher-clients and forwarded them to the subscriber-clients. If a subscribed client is not directly connected to a server, where a publisher has dispatched an event, the server has to forward the event to other servers until the event reaches the server capable of delivering the event directly to the subscribed client. In a P2P architecture each node can act as a publisher, subscriber or event-forwarder to another node.

Reliable pub/sub systems guarantee that published events are delivered to all its subscribers. Durable (fault-tolerant) pub/sub system are able to cope with unreachable subscribers and servers (due to network failures or crashed clients/servers). Some pub/sub systems like SIENA offer a best-effort delivery strategy, i.e. the system will periodically retry to deliver the message until the message was delivered successfully, a timeout expired or the maximum retry-count was reached. In mobile environments [10] client applications reside on a host that is moving and therefore accessing the network (composed of so called *event brokers*) from various locations [4]. The event brokers are responsible to guarantee the reliability and durability of the pub/sub system as described before. Furthermore, a protocol must exist which enables the update of a client's subscription as it is moving from one broker to another. During the client's movement undeliverable events have to be stored by the system and delivered as soon as the client reconnects to the system. When the client reconnects at another broker, all stored events have to be forwarded to that broker. One of the first pub/sub system that supports the reconnection of mobile clients is JEDI [7]. However, our investigations of pub/sub systems have shown that currently most of the available systems are research prototypes which concentrate primarily on scalability and reliability rather on durability in P2P environments. Furthermore, current pub/sub systems for mobile scenarios have the disadvantage that the time needed to update a subscription and to forward all messages to the peer where the re-subscription is made, may take too long. Mobile clients in ITS scenarios are fast moving and have only a few seconds for data transmission (section 3).

In order to adequately address the scaling needs of distributed applications, over the past years there has been research on pub/sub systems making use of the scalability characteristics of Distributed Hash Tables (DHTs) [15]. This has led to several implementations of DHT-based pub-/sub systems, like [6, 9, 16, 1, 2]. However, the papers aim at using DHT like for routing purposes, extended querying, efficient subscriptions, or the efficient distribution of events. In contrast to those approaches ASC focuses on the distribution of Space Containers as fault-tolerant intermediate subscribers, functioning as a scalable and efficient bridge between mobile peers with very short connectivity time and pub/sub systems. Thus, although ASC has been developed with respect to a mesh network (section 3), it does not prescribe the usage of P2P capable pub/sub systems (section 4.2).

3 Scenario

A motivating use-case to illustrate the benefits of the proposed ASC architecture is an Intelligent Transportation System (ITS) scenario [11]. The scenario consists of a highway with fast moving vehicles communicating with a fixed, geographically distributed infrastructure. Along the highway there are so called Road Site Units (RSU) responsible for either passing safety and traffic information to the vehicles or receiving information from the vehicles and pass it to the system. RSUs are installed along the road network in 2-3 km distance of each other and are connected in a meshed wired broadband network in order to assure scalability and increase fault-tolerance. Information exchanged in the system mainly concerns the road traffic itself and the messages are published by e.g. the Traffic Control Centre (TCC), the police, the road maintenance depot. Events generated may contain information about traffic restrictions and warnings, traffic density, traffic congestions, accidents, road and weather conditions. The published data is geo-located and its relevance in space and time is limited to a certain region, moving direction and period of time. A subscriber my be a vehicle driving at high speed or a road worker in a field service. Connectivity between the RSU and the passing by vehicles is characterized by a limited bandwidth, communication range, and connectivity window (ca. 300KB/sec for 2-3 sec at 100km/h in case of a single vehicle) allowing the exchange of small and a few messages only [17]. The received information can be used to adapt driving behaviour since drivers are informed about occurrences in upcoming road segments. Therefore, subscribers are interested in information which a) is represented by the very last event delivered by the pub/sub medium, b) is represented by an aggregated set of events, or c) is a prioritized set of the delivered events. In a special case events can even cancel each other and should not be delivered at all.

4 Architecture

This section pictures how Aspects on top of Space Containers [12] can be deployed for the events processing and illustrates the integration between the Aspect-oriented Space Container and the pub/sub paradigm.

4.1 Aspect-oriented Space Container

Space Containers realize some parts of Aspect-oriented Programming (AOP) by registering so called Aspects[2] at different points of a Space Container. Aspects are executed on the peer where the Space Container is located and can be triggered by the various operations on the container. The join points of AOP are called interception points (IPoints) and can be located before (pre) or after (post) the execution of an operation on the Space Container. Adding and removing Aspects can be performed at any time during runtime. In addition to the parameters of a Space Container operation a so called Aspect Context can be passed along with every operation allowing the client to communicate with the installed Aspects. Aspect may contain any computational logic, thus can be used to realize security (authorization and authentication), the implementation of a highly customizable notification mechanism, or the additional manipulation of already stored or incoming entries. In case multiple Aspects are installed on the same Space Container, they are executed in the order they were added.

Before an operation can be executed on a Space Container it has to pass the installed pre-Aspects. If all Aspects return OK, the container interprets the selectors of the operations and executes the operation [12]. Afterwards, all post-Aspects are executed. Depending on the result of the post-Aspects the result of the operation is either returned to the requesting peer, or the operation is rolled back. As already mentioned, an Aspect can manipulate the execution of the operation which triggered it. This is realized by the returning values an Aspect can throw. The returned value is analysed and the execution of the operation manipulated accordingly. In case of **OK:** the Aspect does not require any changes of the operation; **NotOK:** the execution of the operation is stopped and the transaction is rolled back; **SKIP:** the operation is neither performed on the container, nor on any following pre-Aspects, but post-Aspects are executed immediately afterwards; **Reschedule:** the execution of the operation is stopped and will be rescheduled for a later execution.

[2]the complete API JavaDoc can be found at http://www.mozartspaces.org

4.2 Execution of Aspect-oriented Space Containers in Publish/Subscribe Scenarios

A reason why we recommend that a Space Container shall be used instead of a database is the fact that the number of different events in an ITS is not known beforehand and as a consequence an appropriate data model is difficult to establish. By means of Coordinators [12], a Space Container is capable of using 'dynamic' data models which can be plugged in whenever needed. A Coordinator allows different views, optimally implemented with respect to accessing requirements, on the entire data in the Space Container at the same time. As depicted in Figure 1, such a Space Container and its Aspects are deployed by means of a DHT in the RSU network. The principle is explained in [11], where we described how to combine Space-Containers with an overlay network based on DHT concepts in order to a) make such Space Containers uniquely addressable in a fault-tolerant and scalable manner, and b) to replicate Space Containers in order to increase fault-tolerance and their availability.

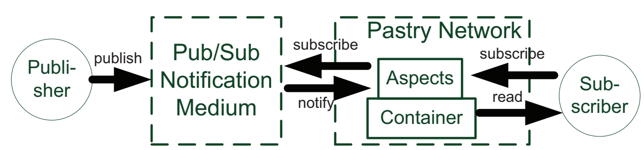

Figure 1: The operation of ASC in pub/sub scenarios

The original subscriber (e.g. a vehicle) places its intention in receiving events from publishers by deploying a Space Container, installing Aspects and publish it in the DHT network. The Aspect registers itself as a subscriber in the pub/sub medium on behalf and according to the requirements of the original subscriber. From now on the Aspect will, independent of the connectivity mode of the original subscriber, receive events which are then processed by other installed pre-Aspects and the results are then stored in the Space Container. When the original subscriber reestablishes a connection to the network, it uses a readselector to pick up the results from the Space Container. If the Space Container is replicated, Aspects are replicated as well. This means that the original subscriber is subscribed as often as many replicas of that Space Container exists. This is necessary in order to avoid missing events in case one of the replicas, including the subscribed Aspect, is off-line. The way how the replicated Space Containers handle incoming events in order to stay consistent is up to the implementation of the deployed pre-Aspects. Either, the replicas are completely independent of each other and perform every operation as many times as replicas exists, or an incoming event is registered and not used for further processing until the result based on that event has been announced from a designated replica. The latter approach may be more efficient with respect to computational resources but require knowledge about group coordination.

5 Conclusion and Future Work

In this paper we described the concept of ASC, Aspect-oriented Space Containers, distributed via Distributed Hash Tables for efficient publish/subscribe scenarios in the Intelligent Transportation System. The benefits of bridging mobile peers with pub/sub systems via ASC are reduction of the complexity of application implementation since the processing logic has been moved to the Aspects. Therefore, this solution allows efficient delivery of events to mobile peers, since the relevant information have been extracted in installed Aspects while the peer was off-line.

Future work contains an evaluation on how efficient the ASC approach is compared to current pub/sub systems. Further investigations regarding the way how the usage of Space Containers and Aspects change the relation between subscribers and the notification medium is intended. Further questions that we will consider are: Is a durable notification medium still necessary if Space Containers are replicated and distributed via DHTs, since DHT use the same network as the notification medium? Does this have an effect on the semantics of durability? What is the influence on QoS coming from the number of replica set up by the subscriber? Another future work will deal with the question how to move Space Containers along the mesh of RSUs to minimize Space Container access time.

References

[1] I. Aekaterinidis and P. Triantafillou. Pastrystrings: A comprehensive content-based publish/subscribe dht network. In *26th IEEE Int. Conf. on Distributed Computing Systems (ICDCS'06)*, page 23. IEEE Computer Society, 2006.

[2] J. P. Ahull, P. G. Lpez, and A. F. G. Skarmeta. Caps: Content-based publish/subscribe services for peer-to-peer systems. *2nd Int. Conf. on Distributed Event-Based Systems (DEBS'08)*, 2008.

[3] S. Bhola, R. Strom, S. Bagchi, Y. Zhao, and J. Auerbach. Exactly-once delivery in a content-based publish-subscribe system. In *DSN*, pages 7–16, 2002.

[4] M. Caporuscio, C. M. Caporuscio, A. Carzaniga, A. Carzaniga, E. L. Wolf, and E. L. Wolf. Design and evaluation of a support service for mobile, wireless publish/subscribe applications. *IEEE Trans. on Softw. Eng.*, 29:1059–1071, 2003.

[5] A. Carzaniga. *Architectures for an Event Notification Service Scalable toWide-area Networks*. PhD thesis, Politecnico Di Milano, December 1998.

[6] M. Castro, P. Druschel, A.-M. Kermarrec, and A. Rowstron. Scribe: a large-scale and decentralized application-level multicast infrastructure. *Selected Areas in Communications, IEEE Journal on*, 20(8):1489–1499, Oct 2002.

[7] G. Cugola, E. Di Nitto, and A. Fuggetta. The jedi event-based infrastructure and its application to the development of the opss wfms. *IEEE Trans. Softw. Eng.*, 27(9):827–850, 2001.

[8] P. T. Eugster, P. A. Felber, R. Guerraoui, and A.-M. Kermarrec. The many faces of publish/subscribe. *ACM Comput. Surv.*, 35(2):114–131, 2003.

[9] A. Gupta, O. D. Sahin, D. Agrawal, and A. E. Abbadi. Meghdoot: content-based publicsh/subscribe over p2p networks. In *5th ACM/IFIP/USENIX Int. Conf. on Middleware (Middleware'04)*, pages 254–273, 2004.

[10] Y. Huang and H. Garcia-Molina. Publish/subscribe in a mobile environment. *Wirel. Netw.*, 10(6):643–652, 2004.

[11] E. Kühn, R. Mordinyi, H.-D. Goiss, S. Bessler, and S. Tomic. A p2p network of space containers for efficient management of spatial-temporal data in intelligent transportation scenarios. *International Symposium on Parallel and Distributed Computing (ISPDC'09)*, 2009.

[12] E. Kühn, R. Mordinyi, L. Keszthelyi, and C. Schreiber. Introducing the concept of customizable structured spaces for agent coordination in the production automation domain. *The 8th International Conference on Autonomous Agents and Multiagent Systems*, 2009.

[13] C. M. M. Pereira, D. C. Lobato, C. A. C. Teixeira, and M. G. Pimentel. Achieving causal and total ordering in publish/-subscribe middleware with dsm. In *Proc. of the 3rd Wsh on Middleware for service oriented computing (MW4SOC'08)*, pages 61–66, 2008.

[14] P. R. Pietzuch. *Hermes: A Scalable Event-Based Middleware*. PhD thesis, Queens' College University of Cambridge, February 2004.

[15] A. Rowstron and P. Druschel. Pastry: Scalable, decentralized object location, and routing for large-scale peer-to-peer systems. *Proc. of the 18th IFIP/ACM Int. Conf. on Distributed Systems Platforms (Middleware'01)*, pages 329–350, 2001.

[16] R. van Renesse and A. Bozdog. Willow: Dht, aggregation, and publish/subscribe in one protocol, 2005.

[17] M. Zaera. Wave-based communication in vehicle to infrastructure real-time safety-related traffic telematics. Master's thesis, Telecommunication Engineering, University of Zaragoza, August 2008.

[18] A. Zeidler. *A Distributed Publish/Subscribe Notification Service for Pervasive Environments*. PhD thesis, Technischen Universitt Darmstadt, 2004.