Chapter 11

Conclusion

A whole is that which has
a beginning, a middle, and an end.

Aristotle, Poetics

In this thesis, we proposed the concept and design of an integrating event notification service. The service adapts to different application requirements, in particular regarding the handling of composite events and event filtering methods. In conclusion of the thesis, we give a summary of the research presented in this work in Section 11.1. In Section 11.2, we outline the open issues with regard to event notification services, and give an outlook on future work.

11.1 Summary of this Thesis

As the motivation and driving force of this thesis, we presented advanced scenarios for remote monitoring and control, logistics control & traffic support, and web-based education in Chapter 2. These examples of multipurpose applications require the efficient integration of event information from different sources under various or changing application requirements. For the applications, we observed drawbacks of the current situation and summarized the applications' demands towards event notification support. Concluding our observations, we defined the following six major requirements for an adaptive integrating event notification service: (R1) Elaborated event model, (R2) Active and passive observation of events, (R3) Integration of composite Events, (R4) Flexibility in event filtering, (R5) Temporal awareness, and (R6) Performance and scalability. In this thesis, we followed the approach of *adaptability*. We addressed the identified requirements by providing flexible support for event composition, for different event observation methods, and for event filtering.

System	R1	R2	R3		R4	R5	R6	
	Elaborated Event Model	Active/Passive Observation	Integration of Composite Events: Profile Support	Integration	Flexible Filtering	Temporal Awareness	Performance	Scalability
SIFT	- -	- -	- -	- -	- -	- -	-	- -
Elvin	+ -	-	- -	- -	- -	- -	+	+ -
Siena	+ -	- -	+ -	+ -	- -	- -	+	++
OpenCQ	++	++	++	++	-	-	+ -	+
CORBA NS	- -	- -	- -	- -	- -	- -	+ -	++
OmniNotify	+ -	- -	+	+	- -	+ -	+ -	+
COBEA	+ -	- -	+	+	- -	+ -	+ -	+
Keryx	-	- -	-	- -	- -	- -	+	++
LeSubscribe	-	- -	- -	- -	- -	- -	+	+
Gryphon	-	- -	- -	-	-	+ -	+	++
REBECA	+ -	- -	+	+	+ -	+	++	++
NiagaraCQ	+ -	++	- -	- -	- -	- -	++	++
SAMOS	+ -	+ -	+	+ -	+ -	-	+ -	-
Sentinel	+	+ -	++	+	+ -	-	+ -	+
A-mediAS	++	++ a)	++	++	++	++ a)	++	++ b)

a) realized in design (not fully implemented)
b) using techniques, e.g., from REBECA and Siena

Figure 11.1: Comparison of related work and A-MEDIAS with regard to requirements

As a first step, we introduced our model for event notification systems in Chapter 3. The model provides a clear definition of the central terms in the context of ENS, the identification of the substantial parts of an ENS and their interaction, and the specification of the event notification process. We started by defining the thesis' central term of 'event notification service'. Based on that definition, we identified 14 related terms and concepts and interpreted how their concepts relate to that of event notification used throughout this thesis. Our model consists of three parts: a reference model, an event model, and an event processing model. The reference model of ENS provides a general terminology for the description of logical parts and their interactions within event notification services. Within the event model, the items handled by the ENS are defined in detail. The tasks of an ENS and the event notification sequence have been described within our event process model. This model addresses the requirements R1 and R2 (see Figure 11.1); it served as the basis to evaluate existing services for their support of the identified requirements.

In the related work, we identified four main approaches to ENS. We distinguished (1) centralized and document-centered systems, (2) simple distributed systems only supporting primitive events, (3) distributed networks of servers with basic support for composite events, and (4) the distributed systems with sophisticated profile definition languages. Four characteristic prototypes representing these approaches have been analyzed in detail, namely SIFT, Elvin, Siena, and OpenCQ. We further selected 10 typical systems to be analyzed for their relation to these prototypes. Along the identified requirements and based on our ENS model, we presented a detailed analysis of the related systems. As result of the analysis, we concluded that none of the existing ENS provide the required flexible adaptation to application characteristics (see Figure 11.1). The design proposed in this thesis is represented in the table by our prototype system A-MEDIAS. Our design fulfills all six requirements identified in Chapter 2.

The design of an adaptive integrating event notification service is the central contribution of this work. It has been introduced in detail in the chapters 5 to 8.

In Chapter 5, we addressed the first aspect of our central topic of adaptability in event notification services: qualitative adaptation. Instead of forcing clients to redefine their profiles for every new source and application context, we proposed to automatically adapt the client profiles. To enable this adaptivity, we introduced a parameterized event algebra for event notification services. The event algebra describes event operators for composite events, it answers the requirements R3 and R4. Additionally, we introduced parameters for event instance selection and event instance consumption. Event instance selection describes which events are taken into account for composite events, and how duplicate events are handled. Event instance consumption defines whether unique composite events or all combinations of events are taken into account. The combination of both parameters offers a flexible means to adapt client profiles to changing applications and sources. A detailed analysis of the profile languages of related systems has shown that our approach supports a more clearly differentiated set of event operators. Moreover, we provide a structured and flexible means for profile definition.

As identified in the requirements, event notification services need to support active and passive event observation. Accordingly, an implementation of our algebra requires the consideration of the influences of the event detection strategies to ensure correct notifications. In Chapter 6, we analyzed how the observation and timestamping methods affect the accuracy of the event information at an ENS. Moreover, the influence of time systems and architecture of a distributed network of ENS has been analyzed. We proposed a strategy for the handling of timestamps of events that aids in minimizing the temporal delays and to increase the event detection accuracy. Additionally, we proposed extensions to the time systems used by event notification systems. The extensions lead to higher accuracy in the detection of primitive and composite events. In that way, an ENS with those extensions supports the accurate integration of event information from different sources. The extensions address the requirements R2, R3, and R5 (see Figure 11.1).

The performance of an ENS has been identified as one of the basic requirements. Our analysis of a performance model for ENS identified the filter as the most influential part for the system's processing time. Therefore, we focused in Chapter 7 on the filter performance. We proposed two algorithms for the efficient filtering of primitive and composite events. For the filtering of primitive events, we introduced a distribution-dependent improvement of a tree-based filter algorithm. The profiles are stored in a tree. We introduced value-dependent and attribute-dependent selectivity measures for the reordering of the profile tree. Our distribution-based approach significantly improves the average case performance for event and profile constellations that are typical in ENS.For the filtering of composite events, we presented a single-step algorithm. We propose a new method for the filtering of composite events that integrates the detection of composite events into the detection of primitive events. After the filtering of a primitive event, its contribution to a composite event is tested. In that way, the composite event is detected successively. No additional step is required for the identification of the composite event after the last contributing primitive event has been detected. The identification of the composite event after the last contributing primitive event is accelerated and the overall filtering time is reduced. Both filter algorithms comply with the Requirement R6 (see Figure 11.1).

In Chapter 8, we address the question of how to realize adaptivity in an integrating event notification service. We distinguish qualitative and quantitative adaptivity. We proposed two methods to

adapt a system to changing application contexts and new sources. First, the translation of client queries according to application profiles and source profiles has been described. In a similar way, accuracy tolerances can be taken into account for event composition. Second, we discussed strategies to determine the distributions of events and profiles. We extended the reference model of ENS with an additional analyzer & optimizer part that controls the system's adaptation. Our approach towards adaptivity meets the requirements R4 and R5 (see Figure 11.1).

The proof of concept in chapters 9 and 10 presents the implementation and tests of the proposed design. In Chapter 9, we introduced A-MEDIAS – an integrating event notification service that adapts to changing event sources and applications. The service implements our parameterized event algebra and our filter algorithms for primitive and composite events. The effectiveness of the adaptation of profiles in A-MEDIAS has been demonstrated in a case study for a facility management application in Chapter 10. The efficiency of the algorithms has been shown by extensive performance tests. For both algorithms, the filter time was reduced by up to 85 to 90 percent. We have shown the effect of the adaptation of the algorithms to changing event and profile distributions in two case studies. The A-MEDIAS system has been implemented as a distributed system consisting of a network of A-MEDIAS servers. The scalability of the system has been evaluated in detail in [Bit03] – A-MEDIAS conforms to Requirement R6 (see Figure 11.1). In summary, our design for an integrating event notification service fulfills the six requirements R1 to R6.

The features of our design of an adaptive integrating event notification service and the contributions of this thesis can be summarized as follows:

1. Analysis of application scenarios and identification of requirements for integrating ENS
2. Model for integrating event notification services
3. Review and analysis of event notification systems
4. Algebra for adaptable event composition
5. Extended timestamping and event ordering method for composite events
6. Efficient distribution-dependent filter algorithm for primitive events
7. Efficient single-step filter algorithm for composite events
8. Prototypical implementation of the adaptive integrating event notification system A-MEDIAS

In short, this thesis presents the *concept and design of an adaptive event notification service* that flexibly supports (1) event integration under various and changing applications and event sources, as well as (2) efficient algorithms that keep high performance under changing system load.

A-MEDIAS is designed to be used for multi-purpose applications such as the facility management in commercial buildings as introduced here. Currently, we are analyzing the A-MEDIAS system in the context of Commercial Big Buildings (CBB), in cooperation with LichtVision GmbH, a Berlin-based company for facility management.

11.2 Open Issues and Future Work

In this section, we point out open issues and suggest promising directions for future work.

Planned Prototype Extensions. One of the open issues is the extension of A-MEDIAS to support different event detection strategies under real-time conditions. This topic has been addressed in Chapter 6, but is not implemented in the prototype. Additionally, future work will extend the prototype to provide automatic adaptation of filter algorithms to changing event and profile distributions as discussed in Chapter 8. Future research will focus on extending the application profiles into sophisticated transformation rules as discussed in Chapter 8. Furthermore, tools for supporting the application administrators in the rule-definition process have to be provided.

Persistency. The problem of persistency of event messages and notifications has been omitted in this thesis, because it does not apply to the considered application fields. However, other applications require persistency and guaranteed delivery of notifications. For example, the support of mobile devices and off-line clients requires a strategy to cope with disconnectedness. Similarly, applications with very long event composition time spans have to store parts of their event history. For event notification, persistency can be achieved by employing an active database system or by integrating the ENS into the database system. Another promising approach is the separation of the persistency facility into an autonomous component, e.g., to support persistency in ad-hoc networks.

Distributed Filtering of Composite Events. The efficient filtering of composite events in a network of ENS servers that considers the characteristics of the distributed environment is an open issue. Several methods have been proposed for primitive event detection. For example, the distribution of events or profiles within a network of servers, e.g., rendezvous nodes [PB02] and profile flooding [CRW99] using optimization methods [MFB02]. In these approaches, composite events are either not supported or simply filtered after the detection of primitive events. Filtering of composite events using the distribution characteristics of the service is another important step towards scalability for integrating event notification services.

Structural Integration. The integration of new or changing sources may also be included without the support of source profiles. An approximate approach could be followed, similar to approximate search [Sch03b]. The filtering of event messages does not require a fixed message structure, e.g., as defined by the event message type. Instead, the messages are filtered approximately, allowing for insertions, renamings, and deletions of structure elements. Each change of an event message would generate certain costs. Then, a profile is matched by an event if the costs of the message changes do not cross a predefined threshold. We are currently investigating this approach in cooperation with T. Schlieder from Xyleme, Paris.

New Application Fields. We believe that an event notification service such as proposed in this thesis does not stand for itself but provides a means to support several application fields. Therefore, another issue is the employment of event notification techniques in new areas, such as location-based services,

semantic web, peer-to-peer networks, and multimedia context. For example, adaptive systems such as A-MEDIAS may be used for mobile systems, where the available data sources change over time and the client can only process a limited amount of data. These areas introduce new challenges for event notification, e.g., routing event data in a peer-to-peer network with unreliable clients. In location-based services, the changing location of a system user may be seen as an event that triggers the delivery of information. We are studying the application of event-based techniques in location-based services in cooperation with A. Voisard from Fraunhofer ISST, Berlin [HV03].

Our goal is to transfer the achievements of this thesis to new applications to face new challenges.