Case Studies on Active Database Applications

H.-J. Appelrath, H. Behrends, H. Jasper, O. Zukunft

Universität Oldenburg, Fachbereich Informatik
Escherweg 2, D-26121 Oldenburg

Abstract. Active databases provide event-condition-action-rules for the implementation of (re)active behaviour in database applications. There exist a lot of proposals for such language extensions as well as some reported implementations. Usually, the suggested features are motivated by application requirements. In contrast, there is a severe lack of reported experiences with applications. Thus, there is no way to judge on the usefulness of the proposals given in the various technical papers up to now. This paper presents first results on the application of active database technology to real-world problems.

In our experimental setting an enhanced active rule language on top of a commercial database management system was used. Three real-world application domains were studied. They encompass the areas of executive information systems (maintaining derived data), cancer clustering (knowledge discovery in databases), and software process management (process management and control). Active capabilities proved to be especially useful for the implementation of the flow of control between various subsystems. Deficiencies of the technology were due to the lack of both modelling and design techniques as well as strong database functionality regarding multi-user mechanisms for the active part of the DBMS.

1 Introduction

Active databases provide event-condition-action-rules (ECA-rules) as reactive language extensions to their respective data model. Their active capabilities are commonly motivated by application demands. Proposed application domains are for instance integrity constraints, maintaining derived data, expert systems [WCL91], network management, Air Traffic Control, program trading, Computer Integrated Manufacturing [CKAK94], and additionally battle management, chemical and nuclear process control [DBB+88]. Generally, there exists a separation into internal and external applications from the database point of view. Internal applications use active rules for the implementation of conventional or extended database functionality, e.g. integrity constraints, view maintenance, access control and optimisation. Applications that use active databases for supporting domain-specific reactive behaviour are called external.

The research in our A.I.S. (Active Information Systems)-project emphasises the application of active database technology to real world scenarios, i.e. mostly external applications. In this paper we present three case studies providing first results on the usability of the reactive paradigm. We have chosen the application
domains from the following separate areas: executive information systems (maintaining derived data), cancer clustering (knowledge discovery in databases) and software process management (process management and control).

A substantial amount of related work has been done in the area of active database systems, see for example [DBB+88, GD92, CKAK94, BZBW95]. Although most of this work is motivated by both internal and external applications only a few experiments with the practical use of active databases are reported.

For internal applications, triggers as predecessors of ECA-rules (see [Esw76]) are traditionally used for the implementation of integrity constraints. ECA-rules have for example been used for constraint and view maintenance and the implementation of deductive databases in the Starburst project, see [CW90, CW94]. A somehow different perspective on "view"-maintenance is given in [DJPa94]. Furthermore, active rules can be used to support the implementation of advanced transaction models, as shown in [GD93].

External applications are described in [Ber94, KNHH94, Jas94]. The computer integrated manufacturing and shop floor control domains are well suited for reactive implementation techniques. In [Ber94] the usage of active object-oriented databases in this domain is proposed but no experiences are reported. The same holds for [KNHH94] where active database techniques were employed for coordinating knowledge discovery algorithms in a dynamic environment. Experiences with our prototype on software process control as proposed in [Jas94] are given in section 3.

To the best of our knowledge, the only experiences with active databases for external applications are reported in [Kot94] and [SKD95]. There, triggers of commercial database management systems are used in banking environments. The main results reported showed severe problems in rule maintenance and performance. Nevertheless, we notice that experiences with ECA-rules are very necessary in order to argue on their usefulness for the external applications proposed in the literature. In the rest of this paper, we first introduce our experimental setting where ECA-rules are used for the coordination of control between various systems, especially databases. This is followed by an elaboration of three case studies in the third section. After that, the lessons learned from these are presented. The paper concludes with the perspectives for future work.

2 The Experimental Setting

When launching the A.I.S.-project in 1991, we tried a pragmatic and application oriented perspective on ECA-rules for active databases. Application and technology improvement were examined in parallel in order to support activity management for real-world applications. The first necessity was to establish a universal architectural frame for active database applications which we call active information systems (AIS). This framework characterises an AIS as an active database application that utilises additional subsystems, e.g. graphical user interfaces. Therefore, an active information system has to offer facilities coupling the active part of the database system with technical subsystems. With respect