The KRISYS Project: a Summary of What We Have Learned so far

S. Deßloch, F.-J. Leick, N.M. Mattos, J. Thomas
University of Kaiserslautern, CS Department
P.O.Box 3049, 6750 Kaiserslautern, Germany
e-mail: {dessloch,leick,thomas}@informatik.uni-kl.de

Abstract
KRISYS is a prototype of a Knowledge Base Management System whose first implementation was completed at the University of Kaiserslautern in 1989. Since then, the system has been used for the development of various applications which allowed us to perform a well-founded evaluation of the system. In this paper, we summarize our evaluation by describing the major lessons we have learned from the design and implementation of KRISYS and, above all, from its use in the development of these applications. We address issues related with the concepts available for application modeling and processing, the support of designing an application, as well as the overall means for efficient processing in a workstation/server environment. Additionally, we point out in how far these experiences validate our approach or stimulate improvements and future research.

1. Introduction
In the last years, substantial research efforts in the area of Database Management Systems (DBMS) have been conducted to support advanced or so-called non-standard database applications [HR85]. This research was, among other things, sparked by the lack of semantic expressiveness in current DBMS [HK87, KOE90]. They do not support the following modeling concepts, which are indispensable in obtaining a more accurate model of complex application domains:

- Abstraction concepts [BMW84, Br81, Ma88a, SS77] are primarily important for the support of a semantically enriched object description. Additionally, they define means for object organization [MM89] which, in turn, can be used to describe distinct application aspects [MDL91].

- There is a need for the integration of behavior into the application model in the form of procedural attributes, user-defined functions, or methods [At89, MMM92]. Such procedures can be used to describe actions in which application objects are involved, thereby permitting the integration of application-oriented operations into the system [DHMM89].

- Reasoning facilities [DK76, Fr86] are necessary to exploit intensional information, to deal with incomplete specifications as well as to control the overall application process, thereby also supporting an active system behavior [DHMM89].

Besides the drawbacks of their modeling concepts, current DBMS also fail to support the process of developing a complex application. This deficiency is becoming even more apparent due to the increasing costs that originate from the use of different models and tools in the development phase (e.g., ER model) and the operation phase (e.g., relational model), which is required when current DBMS are applied as the underlying management system. The need for a single, uniform tool for modeling support has a significant impact on the functionality of future DBMS: they have to be able to act not only as management systems, but also as modeling tools for developing complex applications.

Finally, the processing needs of the applications have to be fulfilled in an efficient and reliable manner, requiring also the consideration of an appropriate runtime environment for non-standard applications [HM90, Ma91]. While ‘classical’ DBMS technology was largely based on a centralized system architecture, workstation/server environments have emerged as typical for advanced application systems. Therefore, the overall architecture of future DBMS should be suitable for such a hardware environment [DFMV90, HHMM88, KDG87, Ma91]. In this setting, locality of reference should be exploited as far as possible; buffering objects close to

1. The address of Mr. Mattos is: IBM, Database Technology Institute, 555 Bailey Av., San Jose - CA - 95150, USA, e-mail: mattos@stlvm14.vnet.ibm.com.
the application seems to be the only means to achieve efficient object references. Also, coupling some kind of 'DBMS' and 'XPS' components in existing architectures is responsible for cumbersome handling and for quite poor performance in most cases [Ma90]. For this reason, the integration of knowledge-based and DBMS techniques in an effective way is one of the main issues to be addressed.

The enhancement of DBMS according to the above mentioned requirements resulted in so-called Knowledge Base Management Systems (KBMS) [BM86,Ma89,ST89]. In such systems, pieces of applications in form of user-defined functions, methods on abstract data types, abstraction relationships, and inference rules are moved inside the KBMS for better performance and higher flexibility.

Along these lines, the KBMS KRISYS (Knowledge Representation and Inference System) was developed at the University of Kaiserslautern [Ma89]. More than thirty diploma and project thesis works were involved in the overall project. The system became completely operational in 1989 [Kr89], when we started to develop several applications from different areas with it. Since 1991, KRISYS is also successfully used in a practical semester course on KBMS and object-orientation at our university.

The experiences gained with the development of these applications as well as the feedback received by using KRISYS in this practical course served as a broad and solid basis for evaluating the system from various points of view. While some of the results consolidated our approach towards KBMS, others helped to reveal some deficiencies. The goal of this paper is to summarize the results of this evaluation and present the 'concrete lessons learned' in the KRISYS project so far. After this introduction, Section 2 gives a brief overview of the architecture of KRISYS and of its main components. In Section 3, the main part of the paper, we present the results of our evaluation followed by some conclusions and an outlook which are given in Section 4.

2. A Brief Overview of KRISYS

2.1 Overall System Architecture of KRISYS

From a conceptual point of view, there are three orthogonal ways of looking at KBMS [BL86,Ma88b], corresponding to the different kinds of requirements that should be supported by these systems: the needs of the applications (i.e., knowledge manipulation means for solving problems), knowledge engineering support (i.e., modeling concepts for KB construction), and suitable resources and implementation aspects (i.e., mechanisms for efficiently coping with knowledge storage and retrieval). The support of these three classes of requirements leads to a natural division of the KBMS architecture in three layers, which were denoted in the KRISYS project as application, engineering, and implementation layer [Ma89] (Figure 1a).

KRISYS follows this conceptual architecture of KBMS, refining it in order to become suitable for a workstation/server environment: The implementation layer is divided into the working-memory system residing at the workstation (together with the application and engineering layer components) and the DBMS kernel managing the KB on the server side (Figure 1b) [Ma88b, Ma91]. Considering the overall system architecture, KRI-

![Figure 1: The KRISYS approach towards KBMS](image-url)