On Using an OODBMS in a CASE Environment

Walid Taha
walidt@cse.ogi.edu
Oregon Graduate Institute

April 14, 1995

Abstract

Three papers were studied in an attempt to explore the impact of OODBMS technology on CASE environments. In this report, first, an effort to unify the terminology of the papers is made. Then, one paper which reports a positive experience with using OODBMS technology in a CASE environment is discussed in more detail.

1 Introduction

The motivation behind this report was the question: Is there firsthand evidence supporting the claim that Object Oriented Database Management Systems (OODBMS) have a positive impact on the quality and ease of development of Computer Aided Software Engineering (CASE[1]) environments?

The report is primarily based on three works: An account of experience with a software development environment (SAMS) built on top of a generalized OODBMS interface[1], an examination of Integrated Project Support Environment (IPSE) technology[2], and an overview of Repository technology[3].

Although all three papers are concerned with aspects related to the automation of software development tasks, each has a different perspective.

[1]Specifically, ones capable of capturing large existing systems, as opposed to ones only capable of automatic code generation.

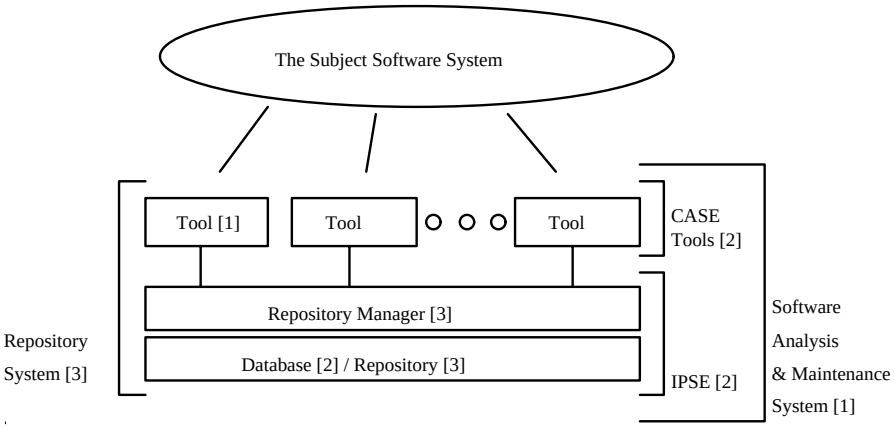

Figure 1: Unifying The Terminology

SAMS is a comprehensive CASE environment. IPSEs and Repositories, on the other hand, do not attempt to provide an *environment*, rather, provide *frameworks* for the interoperation of different tools.

Although the terminology of each of the three papers is also slightly different, two themes common to the three papers are 1) the need to manage the "metadata" in large and evolving software systems, and 2) the fact that tool integration is a fundamental problem in software development.

The structure of this report is as follows: Section 2 contains an attempt to reconcile the terminology of the three papers, and discusses some of the subtle differences between their perspectives. Section 3 relays the findings of Ketabchi relating to OODBMS in CASE. Section 4 concludes the paper, and a short bibliographic note is made in Section 5.

2 Unifying The Terminology

Figure 1 displays the result of an attempt to unify the terminology used in the three papers to refer to what I call a CASE environment. The numbers between the square brackets refer to the paper in which this component of the system is referred to by this name.

One reason for the difference in terminology is the fact that these papers come from different "research communities". Ketabchi is concerned with an object-oriented CASE environment. Brown is concerned with IPSE research. Bernstein's primary concern is metadata management in engineering applications. Another factor, but to a lesser extent, is the tendency for "[Object technology] to blur interdisciplinary boundaries"[8], and it becomes harder to identify exactly which component is responsible for what functionality.

Ketabchi does not appear to emphasize naming or classifying components of the system, and tools are the main focus in his paper. Brown, in [2] and [4], defines IPSE to be the combination of the database with an application layer surrounding it. Bernstein's perspective has some points in common with both Ketabchi and Brown. Bernstein's paper describes a Repository, which is a shared database, and a Repository Manager, which is basically an application layered on top of the Repository.

Bernstein's Repository Manager provides the functionality of checkout/ checkin, version and configuration management, notification, context management, and workflow control. It should be noted here that in his discussion, he assumes the DB to be an RDB, and he actually later admits that OODBMS provide a major part of the Manager's functionality. On the other hand, he introduces these two new and interesting concepts of context management and workflow control[2], which are not typical of OODBMS functionalities. However, it is also not clear why they should not be so, as they would provide useful accounting and performance structures.

3 The Impact of OODBMS on CASE

From my sampling of the literature on the topic, it seems very likely that Ketabchi's paper is the first account dedicated to reporting on experience with using an OODBMS in a CASE environment.

The paper addresses the impact of using an OODBMS in a prototype of SAMS, a novel software development and maintenance system capable of capturing an existing software system into a repository based on a commercially available OODBMS (GemStone). The system is designed to be both independent of the OODBMS and the language of the subject software system. SAMS views a software system as a large structured object, down to the granuality of a lexical token. This has a significant impact on simplifying the task of creating CASE tools. Ketabchi reports that "The most important impact of using an OODBMS as the central component of SAMS is its positive effect on the architectural simplicity of the system."

According to [8], the circumvention of the impedance-mismatch is a major advantage of OODBMS technology. It appears Ketabchi may have sacrificed

[2]Context management is basically providing a way for a user to collect objects relating to each of his tasks together, and workflow control is a way track and hand over results of tasks from one user to the next.

a significant part of this advantage in his design for the sake of database independence, which he achieves using a "database interface layer". However, the extent to which this interface is a bottleneck is dependent on implementation details which are currently not available to me. It is very likely that such "database interfaces" will prevail for some time until a standard for OODBMS is reached.

Although the system suffers from severely inefficient space utilization, it is not clear whether this problem is a result of using an OODBMS, or if it is due to the richness of the models making up the CASE environment. In the latter case, it is likely that the limit on the available physical representations for such models is a major factor.

All in all, I feel that the paper is very insightful, and that the SAMS work is a quantum step into a field that has, and probably still is, longing to be explored.

4 Concluding Remarks

I feel that Bernstein hits on a fundamental attraction of OODBMS technology for CASE when he says that: "Ideally, a repository manager would map its 'object base' into labeled directed graphs, where objects are mapped to nodes and relations [...] to edges". His preference is for OODBMS, but with a reservation on the basis of the "immaturity" of current products.

In all three papers, there was, understandable, interest in being independent of a specific database. This results in introducing a "database interface" in their systems. It is very likely that this interface induces an artificial bottleneck for both design and performance, thus diluting a significant part of the potential of OODBMS. I believe that this calls for a speeding-up of standardization efforts in OODBMS.

5 Bibliographic Note

During the preparation of this report I came across some papers that I felt are relevant to this report, but felt that the exploration of which would require more time than was available. Briefly: Experience with Object database support for CASE at the project planning and tracking level is described in [5]. Experience with an object-oriented framework for systems integration is

described in [6]. For a comprehensive discussion of the requirements for a DB in a SW development environment, [4] refers the reader to, among others, to [7].

6 Acknowledgments

I would like to thank Philip Bernstein for his suggestions and providing me with a copy of his paper, Alan Brown, for valuable comments and pointers to other sources, Paul Hosom and Paul Benninghof for commenting on drafts, and Dick Kieburtz and last, but not least, Dave Maier [3] for the valuable discussions.

References

[1] Ketabchi, M. A., Sadeghi, K. J., "Applying Object Technology to Software Analysis and Maintenance System Development", to appear in J. of Systems and Software.

[2] Brown, Alan W., "An Examination of the Current State of IPSE Technology", Proc. 15th Conf on SW Eng, May 17-21, Baltimore, Maryland.

[3] Bernstein, Philip, Dayal, Umeshwar, "An Overview of Repository Technology", Proceedings of the 20th VLDB Conference Santiago Chile.

[4] Brown, A. W. , "Database Support for Software Engineering", Halsted Press, 1989.

[5] Liu, Lung-Chun, Horowitz, Ellis, "Object Database Support for CASE", pp261-282 in "Object-Oriented Databases with Applications to CASE, Networks, and VLSI CAD", Rajiv Gupta, Ellis Horowitz, Editors, Prentice Hall, 1991.

[3] This report is a termpaper written for the Object Oriented Database Management Systems class taught by Prof. Dave Maier in Winter 1995 at OGI. The paper is available from http://www.cse.ogi.edu/ walidt/oodb.termpaper.ps

[6] Berre, Arne-Jorgen, "COOP - An Object Oriented Framework for Systems Integration", 2nd Int Conf on Systems Integration, June 15-18, 1992.

[7] Bernstein, P. A., "Database System Support for Software Engineering", pp.166-179 in Proceedings of the 9th ICSE (March 1987).

[8] Loomis, Mary E. S., "Object Databases: The essentials", Addison-Wesley, 1995.