

Object Management Issues for Software Engineering Environments — Workshop Report—

Maria H. Penedo^{*}, Erhard Ploedereder^{**}, Ian Thomas[†]

^{*}TRW, One Space Park
Redondo Beach, CA 90278

^{**}Tartan Laboratories, Inc., 461 Melwood Ave.
Pittsburgh, PA 15213

[†]Bull, 68 Route de Versailles
78430 Louveciennes, France

A key objective of Software Engineering Environments (SEE) is to support software projects in the generation, management and control of the vast amount of data and associated information (e.g., properties, constraints, relationships to other data) which is generated and used during the project life-cycle. We denote by the term *objects* the units of data which are identifiable and accessible within an SEE, and by *Object Management Systems* (OMS) the SEE components whose objective is to manage those objects, i.e., provide the means by which they are generated, retained, accessed and modified. A precise definition of an object is largely dependent on the type model provided by an OMS. Object Management Systems have been identified as the core of SEEs; they are being used in various environments as a key mechanism for tool integration.

SEEs deal with a large number of objects. Examples of those objects are:

- Products — generated by the project during its life cycle in support of the generation of the software under development, e.g., requirements specifications, design specifications, manuals, programs, design, test cases.
- Resource information — maintaining information about resources consumed by the project in order to develop the products, e.g., dollars, personnel, computer resources.
- Plans — generated by the project in order to manage the process and to map the expenditure of resources to the creation of products, e.g., development plans, work breakdown structures.
- Automated capabilities — either reused or generated by the project in support of selected project activities, e.g., process programs, tools.
- Measurement data — collected during the project life-cycle in support of the analysis and evaluation of the product under development, the development process itself and/or the automated capabilities.

Issues in object management for SEEs are being addressed by both the software engineering and database research communities. While many of the database issues and solutions applied to other applications still hold for the SEE application, additional unique SEE database requirements have been and are continuing to be identified. As an example, typical database systems deal with a large number of elements and a small set of types of data. SEE database systems deal with a relatively smaller number of objects for a much larger number of types. The impact of the difference on OMS design is still being explored.

On June 29-30, 1988, a group of researchers in the area of environments met to discuss issues related to data management support for software engineering environments. Attendees were: R. Balzer, B. Boehm, S. Boyd, L. Clarke, M. Dowson, S. Graham, B. Meyer, L. Osterweil, M. Penedo, E. Ploedereder, A. Pyster, P. Rook, I. Thomas, A. Wasserman. Two working groups were formed to deal with two topics: data models and integrity management. The discussions which occurred in those working groups are described in the following reports, *Data Models in Object Management Systems*, authored by E. Ploedereder, and *Integrity Management in Object Management Systems*, authored by I. Thomas.

This is an area where it is extremely difficult to achieve consensus on concepts and even on terminology. The reports will provide a glimpse of many of the unresolved problems and issues in the area of OMS support for SEEs. Significant progress needs to be made in this area to support the requirements of future environments.

Permission to copy without fee all or part of this material is granted provided that the copies are not made or distributed for direct commercial advantage, the ACM copyright notice and the title of the publication and its date appear, and notice is given that copying is by permission of the Association for Computing Machinery. To copy otherwise, or to republish, requires a fee and/or specific permission.

©1988 ACM 0-89791-290-X/88/0011/0226 \$1.50