United States Patent [19]

Schmidt et al.

[11] **Patent Number:** **4,558,413**

[45] **Date of Patent:** **Dec. 10, 1985**

[54] **SOFTWARE VERSION MANAGEMENT SYSTEM**

[75] Inventors: **Eric E. Schmidt**, Los Altos, Calif.; **Butler W. Lampson**, Philadelphia, Pa.

[73] Assignee: **Xerox Corporation**, Stamford, Conn.

[56] **References Cited**

U.S. PATENT DOCUMENTS

4,309,756 1/1982 Beckler 364/300

OTHER PUBLICATIONS

Morse et al., DOS/AMAP Model of the DOS Control Program, *I.B.M. Technical Disclosure Bulletin*, vol. 14, No. 3, Aug. 1971, pp. 852–853.

Alan L. Glasser, "The Evolution of a Source Code Control System", Proc. Software Quality and Assurance Workshop, Software Engineering Notes, vol. 3, No. 5 pp. 122–125, Nov. 1978.

Ira P. Goldstein & Daniel G. Bobrow, "Representing Design Alternatives", Proceedings of the Artificial Intelligence and Simulation of Behavior Conference, Amsterdam, Jul. 1980.

A. Nico Habermann, Robert Ellison, Raul Medina–Mora, Peter Feiler, David S. Notkin, Gail E. Kaiser, David B. Garlan & Steven Popovich, "The Second Compendium of Gandall Documentation" CMU Department of Computer Science, May 24, 1982.

Gail E. Kaiser & A. Nico Habermann, "An Environment for System Version Control" in The Second Compendium of Gandalf Documentation, CMU Department of Computer Science, Feb. 4, 1982.

B. W. Lampson et al., "Practical Use of a Polymorhic Applicative Language", Proceedings of the 10th Symposium on Principles of Programming Languages, Austin, Texas, Jan. 1983.

W. F. Tichy, "Design Implementation and Evaluation of a Revision Control System", Proceedings of the 6th International Conference on Software Engineering, Tokyo, Japan, Sep. 1982.

Dissertation of Eric Emerson Schmidt, entitled "Controlling Large Software Development in a Distributed Environment", approved Nov. 1982.

Xerox Palo Alto Research Centers Publication, CSL–8-2–7, dated Dec. 1982, "Controlling Large . . . Environment", Schmidt.

Ira P. Goldstein & Daniel G. Bobrow, "A Layered Approach to Software Design," Xerox Parc. Tech. Report, CSL–80–5, Dec. 1980.

James G. Mitchell et al., "Mesa Language Manual, Version 5.0" Xerox Parc. Tech. Report, CSL–79–3, Apr. 1979.
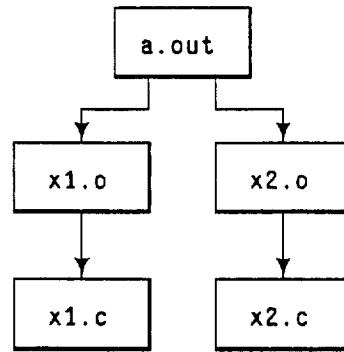
(List continued on next page.)

[57] **ABSTRACT**

A software version management system, also called system modeller, provides for automatically collecting and recompiling updated versions of component software objects comprising a software program for operation on a plurality of personal computers coupled together in a distributed software environment via a local area network. The component software objects include the source and binary files for the software program, which stored in various different local and remote storage means through the environment. The component software objects are periodically updated, via a system editor, by various users at their personal computers and then stored in designated storage means. The management system includes models which are also objects. Each of the models is representative of the source versions of a particular component software object and contain object pointers including a unique name of the object, a unique identifier descriptive of the cronological updating of its current version, information as to an object's dependencies on other objects and a pathname representative of the residence storage means of the object. Means are provided in the system editor to notify the management system when any one of the objects is being edited by a user and the management system is responsive to such notification to track the edited objects and alter their respective models to the current version thereof.
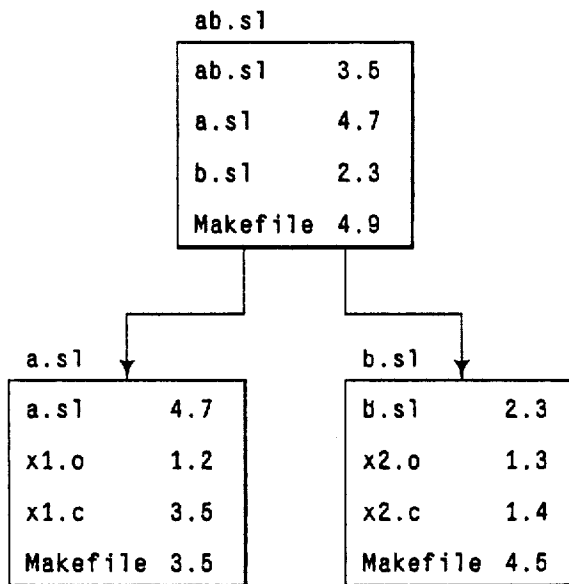
6 Claims, 29 Drawing Figures

OTHER PUBLICATIONS

A. Nico Habermann, "Tools for Software System Construction", Proceedings of the Software Tools Workshop, Boulder, Colorado, May 1979.

Arra Avakian, Sam Haradhvala, Julian Horn & Bruce Knobe, "The Design of an Integrated Support Software System", Proceedings of the SIGPLAN '82 Symposium on Compiler Construction, pp. 308–317, Jun. 23–25, 1982.

Lee W. Cooprider, The Representation of Families of Software Systems, Ph.D Thesis, CMU Computer Science Department, CMU–CS–79–116, Apr. 14, 1979.

Eugene Cristofor, T. A. Wendt & B. C. Wonsiewicz, "Source Control + Tools = Stable Systems", Proceedings of the Fourth Computer Software and Applications Conference, pp. 527–532, Oct. 29–31, 1980.

A. Demers & J. Dononhue, "Data Types, Parameters, and Type Checking", Proceedings of the Seventh Symposium on Principles of Programming Languages, Las Vegas, Nevada, pp. 12–23, 1980.

Frank DeRemer & H. Kron, "Programming-in-the-Large Versus Programming-in-the-Small", IEEE Transactions on Software Engineering, vol. 2, No. 2, pp. 80–86, Jun. 1976.

Stuart I. Feldman, "Make–A Program for Maintaining Computer Programs", Software Practice and Experience, vol. 9, No. 4, pp. 255–256, Apr. 1979.

Ira P. Goldstein & Daniel G. Bobrow, "Descriptions for a Programming Environment", Proceedings of the First Annual Conference of the National Association of Artificial Intelligence, Stanford, California, Aug. 1980.

Eric Harslem and LeRoy E. Nelson, "A Retrospective on the Development of Star", Proceedings of the 6th International Conference on Software Engineering, Tokyo, Japan, Sep. 1982.

Thomas R. Horsley & William C. Lynch, "Pilot: A Software Engineering Case Study", Proceedings of the 4th International Conference on Software Engineering, pp. 94–99, 1979.

Evan L. Ivie, "The Programmer's Workbench–A Machine for Software Development", Communications of the ACM, vol. 20, No. 10, pp. 746–753, Oct. 1977.

Hugh C. Lauer & Edwin H. Satterthwaite, "The Impact of Mesa on System Design", Proceedings of the 4th International Conference on Software Engineering, pp. 174–182, 1979.

D. Redell, Y. Dalal, T. Horsley, H. Lauer, W. Lynch, P. McJones, H. Murray & S. Purcell, "Pilot: An Operating System for a Personal Computer", Proceedings of the Seventh Symposium on Operating System Principles, Dec. 1979.

Marc J. Rochkind, "The Source Code Control System", IEEE Transactions on Software Engineering, vol. 1, No. 4, pp. 364–370, Dec. 1975.

Walter F. Tichy, Software Development Control Based on System Structure Description, PH.D., Thesis, CMU Computer Science Department, CMU–CS–80–120, Jan. 1980.
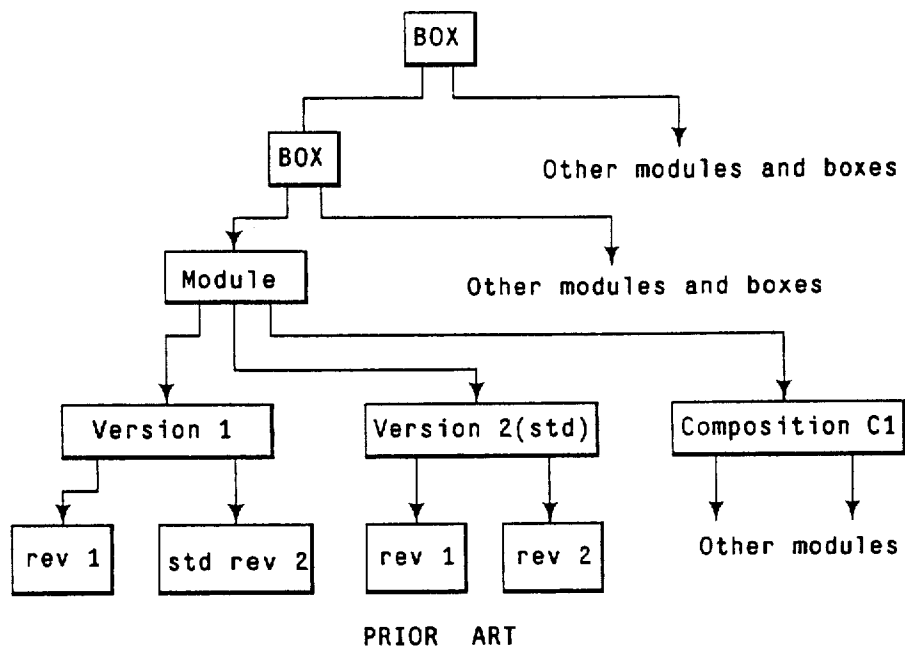
PRIOR   ART   *FIG. 1*

PRIOR   ART
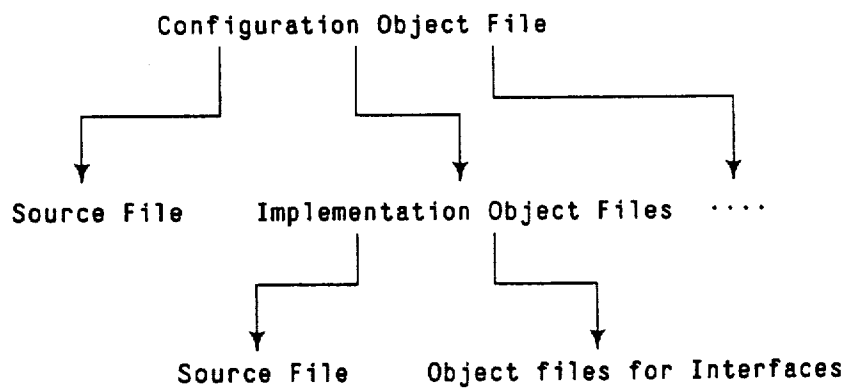
FIG. 2

PRIOR ART

FIG. 3

FIG. 6

```
Client:PROGRAM IMPORTS SqrtInt = {

 .......

 r ← SqrtInt.Sqrt[3.0];

 .......

 }.
```

```
Implementor: PROGRAM EXPORTS SqrtInt ={

 .......

 Sqrt: PUBLIC PROC[s: REAL]RETURN[REAL]={

      ..code to compute sqrt of a number

      };

 }.
```

```
SqrtInt:DEFINITIONS = {

 .......

 Sqrt: PROC[REAL]RETURNS[REAL];

 }.
```

FIG. 4

```
                Impelementation object file
                                 ....

   Source file        Object files for interfaces
```
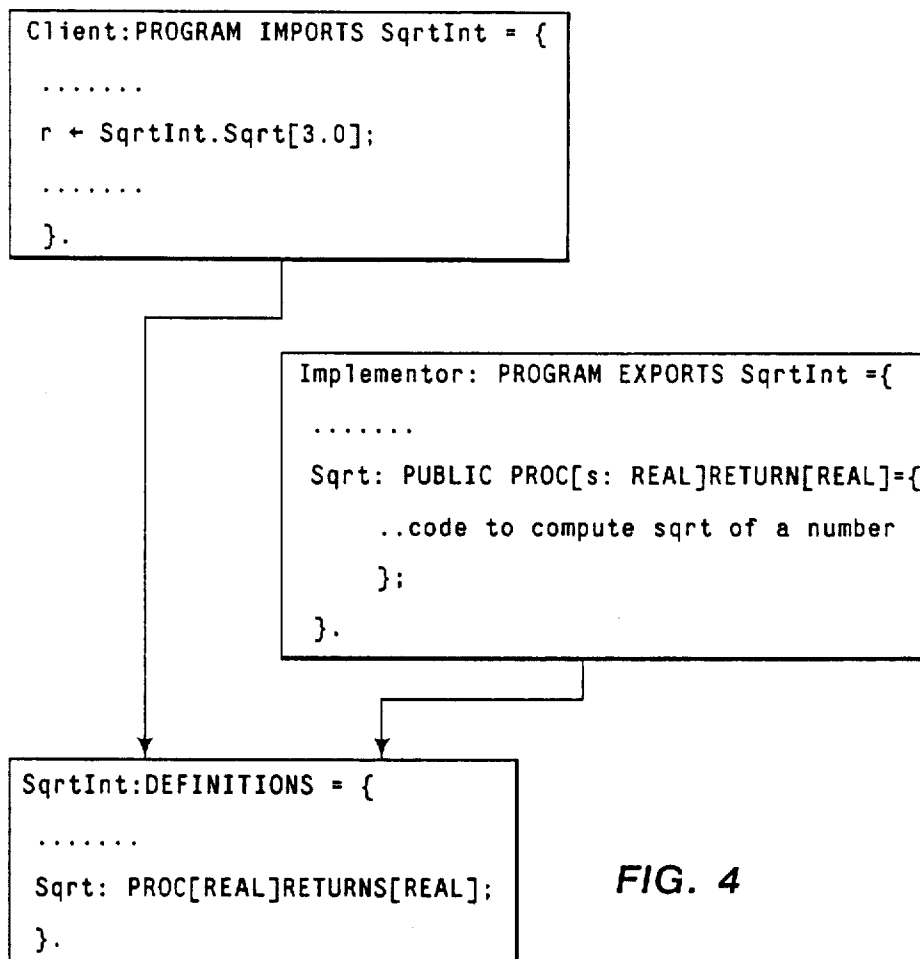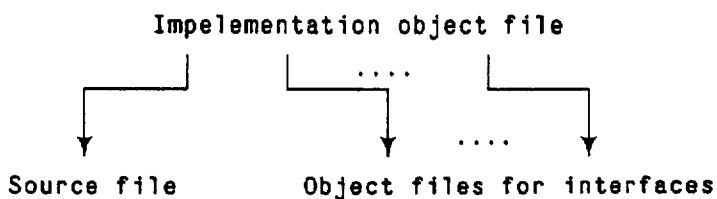
FIG. 5

DOCKET ALARM

Explore Litigation Insights

Docket Alarm provides insights to develop a more informed litigation strategy and the peace of mind of knowing you're on top of things.

Real-Time Litigation Alerts

Keep your litigation team up-to-date with **real-time alerts** and advanced team management tools built for the enterprise, all while greatly reducing PACER spend.

Our comprehensive service means we can handle Federal, State, and Administrative courts across the country.

Advanced Docket Research

With over 230 million records, Docket Alarm's cloud-native docket research platform finds what other services can't. Coverage includes Federal, State, plus PTAB, TTAB, ITC and NLRB decisions, all in one place.

Identify arguments that have been successful in the past with full text, pinpoint searching. Link to case law cited within any court document via Fastcase.

Analytics At Your Fingertips

Learn what happened the last time a particular judge, opposing counsel or company faced cases similar to yours.

Advanced out-of-the-box PTAB and TTAB analytics are always at your fingertips.

API

Docket Alarm offers a powerful API (application programming interface) to developers that want to integrate case filings into their apps.

LAW FIRMS

Build custom dashboards for your attorneys and clients with live data direct from the court.

Automate many repetitive legal tasks like conflict checks, document management, and marketing.

FINANCIAL INSTITUTIONS

Litigation and bankruptcy checks for companies and debtors.

E-DISCOVERY AND LEGAL VENDORS

Sync your system to PACER to automate legal marketing.

fastcase
Smarter legal research.