

The role of the repository

F. Arnold Romberg

Database Programming &

Design. 8.1 (Jan. 1995): p58.

Copyright: COPYRIGHT 1995

UBM LLC

<http://www.ubm.com/home>

Repositories are databases with supporting software and procedures that can store all the information collected, developed, and recorded in development of software. The information can be enterprise goals, critical success factors, organizational structure, logical data, process models, screen designs, report layouts, module structures, logical specifications (code), database definitions, or operating instructions.

Software connected to the repository can be CASE tools, diagramming tools, analysis

tools, development platforms, program generators, or database managers. Use of meta-entity types to develop applications requires development and retention of clear, precise data definitions; repositories are used to record and update meta-entity definitions as they are developed and improved, as well as to access them. The difficulties that repository products must overcome are examined.

Full Text:

REPOSITORIES FOR

Information systems

development projects are being recommended and discussed much more

frequently than they are being used, although their adoption is certainly on the upswing. A repository is a database with

supporting software and procedures that can store all the information collected, developed, and recorded during the development of a software system. The emphasis is on the word "all." The information that can go into a repository could be almost anything: for example, enterprise goals, critical success factors, and organizational structure; logical data and process models; screen designs, report layouts and module structures; logic specifications (code), database definitions, and operating instructions. You could think of a repository as an automated project notebook. As a project is carried through, the repository accepts

information as it evolves--in different representations and with increasing detail. Each development activity extracts information from the repository, perhaps supplements it with added new information, transforms it (often using other systems-development software), and puts the results back in the repository. The software interfaced with a repository may include CASE tools, diagramming tools, analysis tools, development platforms, programs generators, and database managers.

The results are stored in the repository in the form of documents, diagrams, and, ultimately, system components (such as programs, screen management modules, and

database definitions). In short, a repository serves a systems-development organization in the same way that a corporate database serves an enterprise. Obviously, systems developers who want to use meta-entity types to increase the power and flexibility of applications must develop and retain clear precise data definitions. It may be less obvious that data administrators and DBAs who will support the operating databases must also have a clear understanding of the data structures. The repository would be used to record and update meta-entity definitions as they are developed and improved, access them when user and system documentation is

prepared, and perhaps access them again when performing maintenance on the system and data.

Integrated CASE tools, such as Bachman/Analyst, Intersolv's Excelerator, Texas Instruments' IEF, and KnowledgeWare's ADW include their own repositories. Repositories included in these tools tend to be called something else, such as encyclopedias and data dictionaries. (Alice would have called "data dictionary" a portmanteau word. It is an old name with so much appeal that it is continually updated to contain whatever related meaning is currently needed. It started out meaning just a list of the record or segment types and fields in a database, with

explanations and characteristics such as data type and length.)

With the acceptance of the idea that all systems-development information should be in a common database and the growth of open systems for systems development, repositories independent of specific tools but interfaced with them have come to the marketplace. Brownstone and InfoSpan (now part of Manager Software Products [MSP]) are examples.

Repository products have some difficult problems to overcome, including managing different variations and successive versions of the same information, and handling different representations (diagrams,

charts, tables, and spreadsheets) of similar or overlapping information. These concepts are complicated, and the repository facilities for handling them must be accompanied by explanations that are easily grasped and used by systems developers. Another challenging area is change control and configuration management for constructed software and documentation. This area--called software change management (SCM)--is large and specialized enough to be the target of a separate set of software products. Most current repository software falls considerably short of the comprehensive capabilities provided by specific SCM products.

Future releases of repository software packages will include more and better facilities in these challenging areas, more tightly coupled interfaces with other tools, more ways of organizing and presenting information, and more powerful facilities for automatically handling changes. Current SCM products can accept a program logic or database structure change and automatically regenerate, distribute, and reinstall the affected system elements. As advances are made in generating system elements automatically from specifications and requirements advances, we will need repositories that automatically react to changed system analysis and

design information by determining impact, proposing actions and carrying them out automatically, and invoking whatever other processors are required to get the job done.

Source Citation (MLA 8th Edition)

Romberg, F. Arnold. "The role of the repository." Database Programming & Design, Jan. 1995, p. 58. Computer Database, <http://link.galegroup.com/apps/doc/A16623819/CDB?u=oran95108&sid=CDB&xid=96d34439>. Accessed 26 July 2018.