



Turning System Architect into a Repository Gold Mine

Many current modeling tools, such as Popkin's System Architect, are based on repository technology. The repository is specifically designed to support modeling and model analysis and to store the huge quantities of information that analysts and designers produce. The good news: you're already benefiting from repository technology just by way of using System Architect — it's all there by default.

The even better news: you've probably just scratched the surface. By knowing how to dig deeper into the repository gold mine, you can get much more than you might be aware.

Benefits of a Repository

What does a repository do that is so important? Even without much pre-planning, there are several key benefits.

The repository is a holding tank for all the semantics of your models — the meaning behind the diagrams, the business rules implied by relationships, the definitions that underlie the graphics. Because this information is stored in a database, the repository, you can organize and re-organize that semantic information in various meaningful ways and examine it using multiple views.

The repository keeps the project's data organized. All the diagrams and the underlying definitions stay together. You can view the repository as a mini-software configuration management system, linking together all of the related pieces that go into specifying a system. Having all of the data in one place reduces project management errors and loss of information.

The repository supports the sharing and synchronization of metadata within the models. For example, the attributes of an entity can be declared the components of a data flow. Data stores can be directly related to data entities or data models. The capability to integrate the components of a project reduces chances for making mistakes, and for replicating data.

The repository can store additional relationships that become too unwieldy for a diagram. For example, text versions of requirement statements can be linked to diagrams that support the requirement. Several diagrams can be related to a single subsystem application.

The repository enables automatic schema definition from a data model. The underlying definitions and properties stored in the repository become the properties needed for the schema DDL.

The repository is a receptacle for reverse-engineered data. This data can be then be viewed as a physical model, and worked back into a logical model. The reverse-engineered model can be related to a standard model to understand discrepancies.

Empowering the Team

All of the benefits listed are important whether you're working on a big project or a small one. However, usually a *team* of developers works together to develop project requirements and design specifications. This concurrent effort speeds up overall progress and allows experts to work on their own part of the problem without interfering with others.

It's the repository that enables a team development effort. With a repository, you can begin to integrate the data and software requirements across the enterprise, across the system life cycle, and across planned, current, and legacy systems. But — and this is a big but — as soon as you have more than one person involved, you run into issues:



White Papers

Doreen Evans Associates, Inc.

- What if you have no standards in place when you create your models? If each team member makes a different choice, the nice reports and meaningful views described earlier may have less value.
- What about naming standards? You only have to remember the mess our legacy system data is in, partly because we had no control over how things were named, to predict the chaos uncontrolled names in models will bring.
- What about using data that already exists? Wouldn't you like to pre-load project repositories with existing definitions? Finding this data requires both that it is saved in the first place, and saved in a coherent manner to be found and used when needed.
- How should repository contents be archived? You need to know how to name things so you can recognize them, as well as where to store them.
- Wouldn't you like to support a corporate-wide data management strategy? A planned System Architect repository environment can be a major asset in the overall data management scheme.

Many organizations are grappling with issues such as these. Using the System Architect repository, some of our customers are on their way to a managed solution.

Using the System Architect Repository

Like many organizations, one of our clients was revitalizing its basic business processes to maintain and build a strong competitive position. A migration team studied their business needs and created models of proposed business processes. At the same time, IT was shifting its strategy to support two goals: (1) reducing the dependence on a few power developers by instituting processes that enable more developers to be productive on their own, and (2) building applications that can be integrated with each other through their shared data.

What they were really looking for was significant and sustained improvements in productivity and systems quality. The tremendous growth of the business and the growing reliance on the information systems put a lot of pressure on the information systems infrastructure. They believed a repository was the way to facilitate the sharing of systems information while providing some independence for the project teams.

The first task was to design a solution that would support the integration of applications through shared data. The client had anticipated that System Architect would be only a part of the solution. After two days of intensive training, they realized that System Architect could support many of the things they needed to do. Another two days of consulting led to a plan.

The heart of the plan was called a "Virtual Repository," because several System Architect encyclopedias work together as though they were one. This strategy resulted from the identification of five key requirements:

1. Allow small teams to work together on an application, each team sharing its own SA encyclopedia. One team concentrates on the data. Have the ability to bring data and process models together as needed to complete a project.

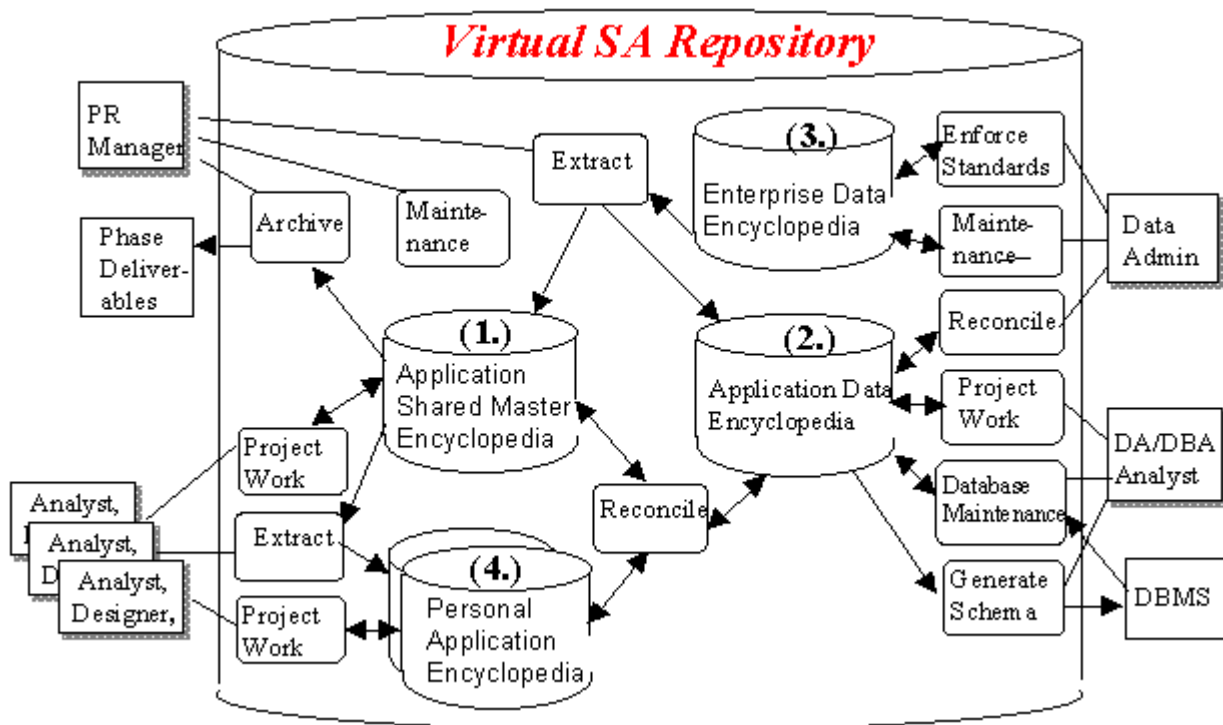


White Papers

Doreen Evans Associates, Inc.

2. Allow DBAs, as the party responsible for the physical data models and the performance and maintenance of the operational database, to work in their own encyclopedia.
3. Keep the logical data models from all projects together in a common SA encyclopedia. This encyclopedia is the responsibility of Data Administration.
4. Support the need to keep all of the models — logical and physical, data and process — synchronized with each other with regards to any shared entities.
5. Provide an environment to support reuse.

The figure below shows a somewhat simplified version of the repository sharing model designed by DEA. There are three main sections.



Encyclopedia 1. Analysts and designers manage the application’s logical data and process models and definitions in an Application Master Repository. Team members share a server-based project repository. Additionally, (in Encyclopedia 4) individual team members can work in a private repository on their own machine during those times when they are not able to be connected to the team repository server, or they need the extra performance of a stand-alone system.

Encyclopedia 2. The DBA builds and maintains the physical database design in the Project Data Repository. Using Schema Generator, they create the initial database DDL. They can tune and make changes to the DBMS schema for performance and other production issues. Periodically, or



White Papers

Doreen Evans Associates, Inc.

after making changes, they use System Architect Reverse Engineer to keep the physical data model synchronized with the database schema. Then they compare the resulting physical data with the previous version of the physical data model. They notify data administration when there is an impact on the corresponding logical data model.

Encyclopedia 3. Data Administrators control the logical models in the Enterprise Data Repository. Data Administration receives logical data models and data definitions from individual projects, and initiates new projects with existing logical data models and data definitions.

The underlying strategy separates the physical data models from the logical models, while still maintaining an essential tie between the two.

The client saw this ability to keep the physical database schema synchronized with the logical data model as an exciting step towards reducing maintenance costs. They felt this would minimize the typical tendency for physical data models to creep away from their corresponding logical models.

Why Use a Virtual Repository Strategy?

There are three significant benefits from the Virtual Repository concept:

1. The multiple encyclopedias provide for the necessary partitioning of work based on a person's role.
2. The partitions can work together as a whole because each encyclopedia can be synchronized with each other.
3. The approach allows an evolutionary, cost-effective implementation of repository technology to begin to realize the benefits of managed metadata.

Using this model, the client can manage its encyclopedias as though they are all part of the same virtual, shared repository environment. The payback for the effort they expend on designing this environment is maintainable, integrated applications. People do their own work without getting in each other's way. Yet each piece becomes part of the whole as needed.

Getting Help for Your Repository Solution

System Architect's rich repository structure can help you achieve your goals. Do you need help thinking through the issues? A comprehensive plan must answer several questions:

Software Configuration. Which System Architect modules will your organization be using? Will you work from server-based team repositories or will each individual work in a local environment?

Environment Enhancements. In order to support your approach, will you need to extend the System Architect repository? Which standard reports will you need?

Standards. Which diagrams and definitions will your organization want to use at each life cycle stage? Which properties need to be completed? How will you use System Architect's setting options to provide consistent appearance?



White Papers

Doreen Evans Associates, Inc.

Integrating Multiple SA Encyclopedias. How will you partition applications to make the best use of the project team? What will you archive and when?

User Roles. What roles will team members play? How will they interact with the repository environment and what processes will they follow? Do you need to add roles for software re-use, for a process manager, for a repository manager?

About Doreen Evans Associates

Doreen Evans Associates (DEA) is a professional services firm that focuses on business process improvement. We can help you change a process, build an enterprise architecture, or define requirements for your systems and technologies. Founded in 1992 as a woman-owned, privately-held small business, our mission is to ensure that business need drives solutions.