

DATABASE

TRENDS AND APPLICATIONS

Solutions for the Information Project Team • www.dbta.com

Volume 21, Number 3 • March 2007

APPLICATIONS INSIGHT



GUY HARRISON

The Ruby on Rails Paradigm Shift

Despite all our achievements, it still seems that what we do hasn't gotten any easier. Modern development environments allow for the development of applications that would have seemed like science fiction 20 years ago. But while these platforms might help make the complicated achievable, they also seem to make the simple more complicated. Nevertheless, things are looking up. Alternative Web development frameworks have emerged that promise to radically reduce the development time for Web applications, especially those whose primary purpose is to manage data held in relational databases. The most prominent is Ruby on Rails, based on the dynamic scripting language Ruby. Ruby on Rails, or Rails, for short, is an open source development platform first released publicly in 2004. The productivity gains from Rails are the result of a number of factors:

Convention over configuration: Frameworks such as J2EE achieve flexibility and power from their configurability. Sometimes this means that absolutely everything must be configured, even when the configuration is basically stan-

dard. Rails relies on naming and coding conventions to bypass most of this configuration when you're developing a run-of-the-mill application.

Model-View-Controller (MVC) architecture: MVC is a widespread architecture, but often the implementation is the programmer's responsibility. In Rails, MVC is assumed and largely implemented in the framework.

Scaffolding: Rails provides code generators that create basic data entry forms almost instantly that can be used for prototyping or sketching out the main segments of the application.

Automatic Object-Relational Mapping (ORM): ORMs are widely used, but often require a lot of effort to establish. Rails transparently creates a default ORM mapping for your database using the ActiveRecord system.

Productive environment: Rails provides an automated testing framework and distinguishes between development, test and production environments. It also provides AJAX support and other productivity enhancements.

Dynamic programming language:

The use of the Ruby dynamic programming language provides many advantages in itself: objects can be typed dynamically and metaprogramming techniques, where code is written by the framework at runtime, can be applied.

Many J2EE enthusiasts believe Rails is a better choice for a large category of Web-based development projects. On the other hand, Rails cannot match Java or C# for native execution speed or rich IDE support. There are, of course, other Web development environments available, such as Django, a Python-based Web framework, that has a similar philosophy and objectives. These next-generation Web development frameworks show a lot of promise and look like the next big thing in Web-based software development. To comment on this column, go to our **Columnists Commentary Blog** at www.dbta.com/blogs/columnists. Hope to see your insights there.

Guy Harrison is chief architect for database tools at Quest Software. He is the author of "Oracle SQL High Performance Tuning" (Prentice Hall).