

The 3000 News Wire

Independent Information to Maximize Your HP e3000



Alfredo Rego

Practicing the Joy of Evolution

Alfredo Rego wants to be known for a career in constant forward motion. It's no easy mission for a programmer so indelibly linked to the heritage of the HP 3000. His accomplishments for the platform include tireless advocacy — HP honored him with its HP 3000 Contributor Award in 1999. Adager, the company he leads with CEO Rene Woc, made its reputation on his 1978 creation of the first direct database transformation software for a computer which is defined by its superior database. He's been programming long enough to recall relying on serially accessed, small tape cassettes as his only storage device for an accounts receivable application he wrote for a 9830 HP calculator. He also wrote applications for the HP 3000's predecessor, the HP 2100, on systems with a total of 5Mb of disk and 800 BPI tape drives that didn't have read-after-write. Rego takes pride in such accomplishments which he says "common commentators label as impossible."

His mission has often been one of preservation, too. In the mid-1990s, Adager led the 3000 community through the Y2K transition with a comprehensive set of routines that examined a vast array of data types and formats, so customers could be assured their information would survive the millennium shift.

An instinct for survival appears to drive Rego beyond historic accomplishments, however. Aware that his critics point to his history and call him a throwback to a lost world of computing, he keeps reaching for new projects which pique his interest. This year he began training for competition alpine skiing in Europe at age 54, evolving his passion for a sport that has included an appearance in the 1988 Olympic Games — a pursuit he modestly refers to as "more of a symbolic participation than an athletic one." Rego spent weeks away from his company's Sun Valley base during training and competition, but stayed connected to his work with IMAGE/SQL and the 3000 through Internet links, as well as a mobile HP 3000 installation he managed to get into Alpine ski lodges where he trained.

About the same time, Rego took on a project to rescue several hundred HP e3000s in the Pacific Rim which needed evolution. Firms using that many 3000 systems needed to convert data in the Chinese language from the ccdc Chinese representation to another (Big5) this year. In addition to the challenges inherent to the Chinese conversion itself, Rego had to develop totally new ways to deal with the rehashing and resorting required whenever search and sort fields were involved in the Chinese IMAGE database conversion. The task, which kept those 3000 customers in the HP fold, meant grappling with B-Trees, SQL

working directly with the database's RootFile. Our conversation spanned communication both old and new: written replies to questions through the Internet, as well as telephone discussions. We wanted to check in with the programmer whose experience spans so much of the community's history, and ask how the genetics of database mastery are helping him continue his own evolution.

Why is the RootFile so important to database users who work on HP 3000s?

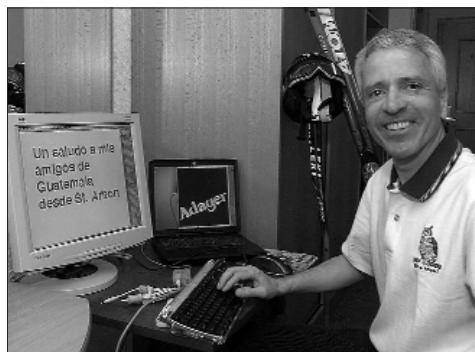
The database's RootFile is the repository for all the structural specifications for the database. There must be a strict correspondence between a database's RootFile and a database's "data" files, which include datasets, B-Trees, DBEnvironments, and so on. In addition, of course, there must be absolute consistency within the RootFile and within each "data" file. For a classic description, please see www.adager.com/LiteraturePDF/ConsistencyCheck.pdf

Is there an alternative to doing so much in the RootFile during transformations, like schema processing?

I'll answer more generally first. There are two basic ways to "transform" anything:

The first method, "Delete the thing and rebuild it again from scratch," involves a long and painful sequence of tasks. First, you unload all of your database's data to tape. Then, you delete your database. Then, you edit your schema and you create a new (empty) database. Then you load all of your database's old data from tape (being very careful to "filter" the data if its layout or format changed).

The second method, "Modify the thing only where it needs to be modified," is the method that Adager introduced to the database industry in 1978. Adager keeps a careful description of what needs to be done to accomplish your desired database transformations. Once you are satisfied with your requests, you ask Adager to "apply changes." Adager then schedules all the necessary tasks so that a minimum of processing will get your desired results. You tell Adager what you want (more items, less



Alfredo Rego
Programmer
Adager Corporation

DBEnvironments, and more, while keeping a tight grip on the issues associated with Chinese-character conversions.

Rego is a subject whose reputation can sometimes interfere with seeing his accomplishments from a fresh perspective. Attaching one's career to the fortunes of a single database not well marketed by its creator carries the risk of relegating a life's work to footnote status. But as HP has evolved its e3000 business, it recently began to point to IMAGE/SQL as the defining value of its longest-thriving business server. In this month's Webcast, the company noted with some pride that no other maker of business servers includes an award-winning database with every sale.

HP is beginning to acknowledge that IMAGE/SQL is at the root of the HP 3000, and it can be argued Rego works at the root of IMAGE, given Adager's dedication to

datasets, reformatted fields, whatever) and Adager takes care of giving it to you.

Now, I can answer your specific question: Yes, there is an alternative to working directly with a database's RootFile and "data" files. This alternative involves fooling around with schemas and such.

Fortunately, Adager obsoleted such a primitive alternative back in 1978.

Adager has always worked through the RootFile, directly. I have been sitting too much on my behind regarding Adager's "schema" interface (for those people who have a hard time letting go of their old "schema-oriented" mentality). One of these days, I am going to get off my behind and do it. Old habits die hard, I guess. I thought Adager had obsoleted database reloads (and, by implication, schema-related stuff) back in 1978. Wrong! The retrograde forces of inertia are strong.

Naturally, RootFile transformations are just one part of the task (the "genetic" part). Dataset transformations ("dataset mapping") are another part (the "cosmetic" part). There are several others: SQL DBEnvironment re-synchronization, B-Tree and TPI index re-synchronization, and so on.

Why should 3000 database administrators care about RootFile versus schema transformations?

RootFile-only transformations are almost instantaneous with Adager. They are not when using a schema intermediary step.

Dataset transformations, which can be lengthy depending on the size of the datasets involved, are best handled in a batch-like mode, with a lot of management and scheduling up-front to accomplish everything with a single pass per dataset.

I wonder how schema tools handle the case when a user renames a data item — or a dataset — and reshuffles its location at the same time, within the same run. Even better, how about the case when a user renames several data items or datasets — and then recycles some of the old names by applying them to other datasets while reshuffling their locations?

Adager uses proprietary data structures to keep track of these trivial things (as well as many more not-so-trivial things). Try doing them with a simplistic schema methodology! For extra credit, try changing datasets from detail to master (and vice versa) in the process. Adager will not even break out a sweat. This is computer science at its finest and it all happens on the HP 3000 under any flavor of MPE and IMAGE. I am very proud of these accomplishments and, as far as I know, my "Database Genetics and Cosmetics" paper is the only one that addresses these issues.

In 1987, the BARUG benchmark proved

that Adager beat the competition hands down, thanks to this proprietary technology. The technology is now stronger than ever, thanks to the thorough review that I did for the Chinese project in the last few months.

SIGIMAGE voted to approve an implementation of the Enchilada limited data caching project which uses the RootFile. What's your technical opinion on the chosen method of extending the root file to do this work?

I defer all Enchilada questions to [SIGIMAGE/SQL chair] Ken Sletten. In fact, given the current climate everywhere, I would rather stay out of politics and religion.

HP is very involved in a technical investigation of thread-aware (and hopefully thread-savvy) IMAGE. What kind of advantages would this bring to the 3000 customers?

More concurrence within the Posix environment and, as a consequence, more concurrence in general. But there are huge tradeoffs involved. It has to do more with Posix threads, which I understand are somewhat weak in the 3000. I haven't followed them on purpose, so I can focus on my own work. I cannot afford to be all things to all people. I invite readers to browse to jazz.external.hp.com/papers/image-threads.html for Tien-You Chen's paper on the subject. He is the best bet for this hot topic.

We're hearing about ventilation again in database transformation tools, but the term doesn't seem all that new. What's the concept about, and what's its heritage in the 3000 community?

Our Adager Guide available on our Web site has had the word "ventilation" as a key word in it for many years. We also have a classic paper that has been around for a very long time, "Do Migrating Secondaries Give You Migraines?" It has a lot of good theory and good practical advice. "Ventilation" appears four times there.

Metropolises like New York tend to have a tremendous density of buildings, so things are very crowded there. In a master dataset, due to the way that addressing works, you associate an address with a value for a search field; for instance, Seybold might go into position 25. But perhaps White also goes into position 25, so now you have a synonym. They both mathematically belong to the same address, but they cannot both go there. One is the primary, and the next one to get there is the secondary.

You have to find a space for the secondary to live. In a place like New York City, you may have to go to New Jersey, or somewhere where there's open space. You spend

some time looking for that vacancy, and that's where the secondary goes.

The density of population may be due to all of them being primaries, or all of them being secondaries. You do your search, and you find that all of your surrounding territory is just primaries, and there's nothing you can do, because you cannot migrate them. If you have a high density of secondaries, most likely they will be migrated, and that's wasteful.

A ventilation shaft is nothing more than an empty spot. Adager gets those empty spots by relocating selected groups of secondaries elsewhere. I have the freedom to choose where I put secondaries when I rehash a master dataset. IMAGE automatically does not do that, and just puts the secondaries wherever they may fall. It's conceivable that a single secondary may migrate thousands of times in its lifetime. You want to minimize the migration of secondaries. A ventilation shaft is nothing more than a strategically located "empty" span (without any entries of any kind) that provides a place for DBPUT to breathe.

HP continues to extend the limits on IMAGE with the latest release of the database. Which limits are receiving the most pressure from the customers you speak with?

Rick Gilligan (with CASE, a provider of banking portfolio management applications) has really appreciated the expanded limits for data items and is already looking forward to the next level.

Amisys (a provider of health-oriented applications) has benefited from the expanded limits for datasets. Ken Sletten and Wirt Atmar have enjoyed the expanded limits for master dataset paths (which allowed them to use B-Trees extensively).

In the past, you've used words like "genetics" and "cosmetics" while describing database science on the HP 3000. What do these concepts deliver to users who want to make changes to databases intuitively?

I still use those words. These concepts are timeless, but the concepts themselves don't deliver anything to users. The concepts are crucial for developers of software tools. A primitive tool (which depends on a database's schema, for example, to rebuild a database's RootFile), does not have the "fine motor skills" required to do delicate and complex work inside the RootFile's structure. An advanced tool (which has its own mechanisms to deal directly with the RootFile's internal -- and privileged -- structures) has full control of all the nuances, such as the ones you mentioned. The fact that you rename data items and datasets at the same time that you reshuffle them is no

big deal for Adager (because Adager keeps track of each data item and each dataset via fundamental invariant attributes - in the mathematical sense - and not via secondary fickle attributes such as name or location).

Coming back to your question, I see that the concepts DO deliver something to users, but in an indirect way. If the tool used by the user is built on these concepts, then the tool delivers unprecedented power to the user. If the tool used by the user is not built on these concepts, then the tool delivers extremely limited power to the user.

Database administration feels like a science that's receding among HP 3000 customer prospects, with all the packaged software being sold today. How do you make a case for this kind of prospective customer — indeed, anybody — investing in the tools and training to become adept at that science?

“Understanding” is the key to anything. If you understand something, you are free to use it wisely. If you don't understand something, you are used by the thing (you become a useful fool).

Some people choose to buy packaged software. No wonder that “services” are the fastest growing segment in the technology sector.

Even within the HP 3000 world, a sophisticated and powerful database administration tool for IMAGE such as Adager costs \$7,600. You can't get a recent college graduate to give you consulting on SAP “administration” for this amount. How do you get training on Adager and IMAGE databases? Very simply and very inexpensively. You make a copy of one of your own databases and you adapt it and manage it as you browse through Adager's Guide. For an impartial benchmark, you can go to a common SAP (or Oracle, or whatever) class and, after many thousands of dollars and a

lot of time and aggravation, you decide.

What led you to work on the Chinese HP 3000 project?

Some people do market research to find out what others find interesting. I have always had the tendency to do what I find interesting. In particular, I have always been attracted by projects that common commentators label as impossible.

For example, Chinese language conversion from one Chinese representation (ccdc) to another (Big5) in 2001.

What do you consider to be a more personal example of such an “impossible” challenge?

Training regularly for world-class alpine skiing events, at the tender age of 54. This particular example might serve as a ray of hope for computer scientists, who tend to spend too much time on our collective behinds. I managed to train with some national teams in Europe, and they always looked funny at me — because here I was working in the afternoon, while those guys were lifting weights, bicycling, training all day. Four hours a day was enough for me. I felt like I earned the right to sit down and program. But after programming for a while, I felt like I needed to move. It was the perfect complement.

Training for alpine skiing requires long-term planning and daily discipline. Some people might be surprised to learn that intense training for the 2001/2002 (northern hemisphere) winter season started last April. One can't just show up on the slopes without having invested many months into a carefully planned (and executed) integrated training regime.

My typical day includes bicycling, running, weight training, hiking, skiing (in European glaciers between April and October). A big difference between a “normal” athlete and me is that I spend the rest

of the day working on my computers, doing Adager R&D on the HP 3000 as well as ancillary Internet work for Adager's business infrastructure. Just lugging the HP 3000 around provides a healthy quota of exercise.

Is IMAGE more difficult to understand and program today than it was 20 years ago?

IMAGE is not, but new technologies to interoperate with IMAGE may be.

Is IMAGE more or less robust than it was ten years ago?

IMAGE is more robust today. Transaction volumes processed by current systems are much higher than 10 years ago and the relative amount of problems is less.

In the last 10 years there have been several complex enhancements (TPI, Netbase, DDX, MDX, Jumbo datasets, B-Trees, SQL).

IMAGE is part of today's high-availability solution, even though it doesn't get mentioned explicitly as much as other products. IMAGE is at the center of any HA solution: Shadowing options, backup options.

What kind of company is the best prospect for a solution that uses IMAGE, rather than something like an SAP-like solution?

Any company that does not have extra money, time, and attention to throw away.

The best prospects for an MPE/iX and IMAGE solution are companies that are owned and managed by people who are intimately linked to their profitability. The best prospects for SAP-like solutions are companies that are owned by sheep-like stockholders and managed by hired guns who have zero loyalty to fundamental values and who thrive in cooking the books.

Copyright ©2001 by The 3000 NewsWire. Reprinted from Vol. 7, Issue 1 of The 3000 NewsWire, 11702 Buckingham Road, Austin, TX 78759.