On-Demand Enterprise Services: Where’s the Catch?

Hakan Erdogmus

Enterprise software is plainly problematic, no matter whose perspective you take—that of the vendor selling it, the buyer acquiring and operating it, or the people using it. It’s especially a nightmare to deploy and upgrade. Everybody suffers. Software as a service promises to address some of the ills afflicting product-based enterprise software. But a few flies might be lurking in the ointment.

The virtues of on-demand

In a keynote address at a recent major conference, the executive of an up-and-coming IT company evangelized the virtues of delivering enterprise automation as an on-demand service. I believe he made a good case.

He recounted the several benefits of a common, scalable platform catering to multiple subscribers through a unified underlying engine and data model. The underlying platform is shared, but a customer-specific metadata layer lets individual subscribers generate and deploy applications without the nightmarish worries of traditional enterprise systems. Ease of management and tailoring, faster release cycles, painless customer provisioning, and maintenance-free upgrades were among the benefits he touted. He admitted that platform upgrades that are completely transparent to subscribers are a challenge for service providers, but once overcome, this challenge translates into handsome payoffs for customers when they instantly inherit any improvements made.

Ease of application customization and integration is intended to empower customers. They can create virtual applications by defining and specializing layouts, data objects, workflows, reports, and so on, which persist as part of their private metadata. While wizard-like features can guide simple tailoring, the real power would come from a programmatic interface in which concepts such as database tables and workflows are first-class objects. Such an interface, in addition to customization, would let customers glue separate applications, hopefully with less complexity than current product-based alternatives require.

What about security and privacy?

In this context, security and disaster recovery aren’t as disconcerting as they might initially seem. Really. A common multisubscriber platform must build in security and redundancy from the ground up. And service providers are better positioned than individual organizations to offer effective related capabilities at low cost—with defendable assurances of protection from intrusions, attacks, and failures.

The same should apply to reliability and performance. Maintaining and monitoring a single underlying platform lets the provider quickly identify bottlenecks and glitches and deploy global fixes to address them.

Privacy is touchier than security. The thought that their organization’s critical data are physi-
FROM THE EDITOR

Closed for business

On top of platformwide internal redundancy, service providers sometimes encourage customers to further protect themselves by offering an external backup capability. Customers may then regularly back up their hosted data in some standard database format to a server that they control. The actual physical data the provider uses internally for its multisubscriber platform are of no value to the customer. Neither may the provider make the raw data available without compromising other subscribers’ sensitive information and its own intellectual property. The idea is that the platform assembles a snapshot of the subscriber’s private data on demand and makes this snapshot available for external backup. This strategy should theoretically offer a reasonable assurance to the customer, but in practice, it does not.

Suppose the provider goes out of business. The customer is left with a static database and no metadata. The end users don’t directly interact with a database but with a virtual application that implements a set of business processes encoded within the metadata. So, the machinery that brought the whole thing alive has effectively disappeared with the provider.

This machinery consists of the metadata (created and owned by the customer) and the engine (created and owned by the provider) that interprets the metadata. Putting new machinery in place could amount to nearly the same enterprise automation problem the customer faced before the system existed. Worse yet, now the organization relies on the system for its day-to-day operation.

As a first-order solution, the backed-up data could also include the metadata, a representation of the underlying business processes, including custom workflows, reports, layouts, dashboards, and so on in standard formats. At least the customer would possess all its invested corporate assets, with some hope of recovering them.

A second-order and more complete solution is for the provider to offer an easy-to-deploy, specialized, single-subscriber instance of the engine to be coupled with the external backup. Just like the private database, the hard-coded engine is automatically generated. In response to an interruption in the provider’s existence, the customer would have the right to deploy the instantiated, complete system as a short-term solution to keep operations up and running—that is, until a more permanent solution becomes available.

Although developing such a second-order capability is quite hard, the provider must do it only once at the platform level. The question is whether the risk-averse customer would be willing to pay for it in return for the added protection of a significantly reduced downtime.

Who performs application development?

In many situations, the customers’ illusion of partial control over data and processes might be powerful enough to automate organization-specific business processes with reasonable fidelity. Although such control is theoretically both complete and real in traditional product-based enterprise systems, the cost of change is often so prohibitive that this ultimately results in very little actual control. So we may indeed be better off with the service approach. In any case, the customer still deals with the residual complexity of application customization.

STAFF

Senior Lead Editor
Dale C. Strok
dstroki@computer.org

Group Managing Editor
Crystal Shih

Senior Editors
Shani Murray, Dennis Taylor, Linda World

Assistant Editor
Brooke Miner

Editorial Assistant
Molly Gamborg

Publication Coordinator
Hilda Carman
software@computer.org

Production Editor
Jennie Zhu

Technical Illustrator
Alex Torres

Associate Publisher
Dick Price
dprice@computer.org

Membership/Circulation Marketing Manager
Georganne Carter

Business Development Manager
Sandra Brown

Senior Production Coordinator
Marian Anderson

CONTRIBUTING EDITORS

Robert Glass, Keri Schreiner,
Joan Taylor

CS PUBLICATIONS BOARD

Jon Rokne (chair), Mike Blaha,
Doris Carver, Mark Christiansen, David Ebert,
Frank Ferrante, Phil Laplante, Dick Price,
Don Shafer, Linda Shafer,
Steve Tanimoto, Wenping Wang

MAGAZINE OPERATIONS COMMITTEE

Robert E. Filman (chair), David Albonesi, Jean Bacon,
Arnold (Jay) Bragg, Carl Chang, Kwang-Ting (Tim)
Cheng, Norman Chonacky, Fred Dougills,
Hakan Erdogmus, David A. Grier, James Hendler,
Carl Landwehr, Sethuraman (Panu) Panchanathan,
Maureen Stone, Roy Want

Editorial: All submissions are subject to editing for clarity, style, and space. Unless otherwise stated, bylined articles and departments, as well as product and service descriptions, reflect the author’s or firm’s opinion. Inclusion in IEEE Software does not necessarily constitute endorsement by the IEEE or the IEEE Computer Society.

To Submit: Access the IEEE Computer Society’s Web-based system, Manuscript Central, at http://cs-ieee.manuscriptcentral.com/index.html. Be sure to select the right manuscript type when submitting. Articles must be original and not exceed 5,400 words including figures and tables, which count for 200 words each.
The IEEE Software Boards

Warren Keuffel, the long-time editor of our popular Bookshelf department, has recently retired. The Editorial Board and staff thank him for his many years of contribution and wish him success in his future endeavors.

Art Sedighi is the new Bookshelf editor and a member of the Editorial Board. Art has been a frequent reviewer for the department. In his new role, he will identify relevant and worthy publications to review, solicit reviewers, and oversee the reviews’ quality.

Erik Meijer and Linda Rising have joined the Advisory Board. Both are thought leaders, sought-after commentators, and extremely active in their respective communities. Best known as a patterns guru, Linda helps development teams reflectively transform to incorporate new ideas into their processes. Erik is an architect at Microsoft’s SQL Server Group and an authority on functional programming languages, type systems, and Internet programming.

Welcome Erik, Linda, and Art. —Hakan Erdogmus

and integration after the provider has neatly dealt with all the platform issues. The customer’s private metadata is key to achieving control. But exactly what kind of people on the customer side would be responsible for creating, evolving, and maintaining the metadata? End users would in all likelihood be a couple of levels removed from these activities, as is the case with traditional systems. If robust application customization and integration hinge on leveraging programmatic interfaces (likely to be supported by plenty of modeling activity), such responsibility would naturally fall on the shoulders of IT departments (and consultants). The IT folks won’t be absolved from solving out complex requirements, mapping them into data models and weaving them into interacting workflows.

On the upside, the IT person’s job becomes easier. Liberated from dealing with daunting and recurrent platform issues, the focus can now be shifted to their core problems and strengths and take advantage of scale economies when doing so. While the provider maintains the platform and creates generic capabilities in a transparent manner, customers create and maintain artifacts related to their core corporate assets. This synergy is preferable to frictions resulting from end-user-inaccessible modeling) remains, well, a dream.

An uncertain outlook

In sum, the on-demand, multisubscriber paradigm promises a sensible sharing of risks and responsibilities between enterprise service providers and consumers, enabling each party to focus on their core problems and strengths and take advantage of scale economics when doing so. While the provider maintains the platform and creates generic capabilities in a transparent manner, customers create and maintain artifacts related to their core corporate assets. This synergy is preferable to frictions resulting from clunky product-based solutions, but it’s not a panacea. Nor is its success, technologically or in the market, guaranteed.

It will be exciting to see how the enterprise software industry evolves in response to emerging service-based business models. Traditional players have already been incorporating “services” in their vocabulary. Will additional obstacles and opportunities arise? How will providers and their customers innovate in turn? What will the impact be on software development practices? And how will end users be affected? I’m interested in your perspective. Reach me at hakan.erdogmus@computer.org. ☺

FROM THE EDITOR

EDITOR IN CHIEF

Hakan Erdogmus
hakan.erdogmus@computer.org
EDITOR IN CHIEF EMERITUS:
Warren Harrison, Portland State University

ASSOCIATE EDITORS IN CHIEF

Design: Philippe Kruchten, University of British Columbia; kruchten@ieee.org
Empirical Results: Forrest Shull, Fraunhofer Center for Experimental Software Engineering, Maryland; tshull@fc-md.umd.edu
Human and Social Aspects: Helen Sharp, City University, London; h.c.sharp@open.ac.uk
Management: Stan Rifkin, Master Systems; sr@master-systems.com
Processes and Practices: Frank Maurer, University of Calgary; Maurer@Cpsc.ucalgary.ca
Programming Languages and Paradigms: Sophieh Drossopoulou, Imperial College London; s.drossopoulou@imperial.ac.uk
Quality: Annie Combelles, DNV/Q-Labs; annie.combelles@dnv.com
Requirements: Roel Wieringa, University of Twente; roelw@cs.utwente.nl

DEPARTMENT EDITORS

On Architecture: Grady Booch, IBM; grady@booch.com
Bookshelf: Art Sedighi, SoftModule; sediga@alum.rpi.edu
Design: Rebecca J. Wirfs-Brock, Wirfs-Brock Associates; rebecca@wirfs-brock.com
Loyal Opposition: Robert Glass, Computing Trends; rlglass@acm.org
Not Just Coding: J.B. Rainsberger, Diapason Software Services; me@jbrains.info
Requirements: Neil Maiden, City University, London; n.m.maiden@city.ac.uk
Software Technology: Christof Ebert, Vector Consulting; christof.ebert@vector-consulting.de
Tools of the Trade: Diomidis Spinellis, Athens Univ. of Economics and Business; dds@aeub.gr
User Centric: Jeff Patton, ThoughtWorks; jpatton@acm.org

ADVISORY BOARD

Stephen Mellor, consultant (chair) Jennitta Andrea, ClearStream Consulting
Elisa Baniasadi, Chinese University of Hong Kong J. David Blaine, independent consultant
Ward Cunningham, AboutUs David Dorenbos, consultant Kaoru Hayashi, SRA
Simon Helsen, SAP Juliana Herbert, ESICenter Uwaxos
Gargi Kauri, Tata Consultancy Services Karen Mackey, Cisco Systems
Steve McConnell, Construx Software Erik Meijer, Microsoft
Bret Michael, Naval Postgraduate School Ann Miller, University of Missouri, Rolla Deependra Moitra, Infosys Technologies, India
Frances Paulisch, Siemens
Linda Rising, independent consultant Wolfgang Strigel, independent consultant Dave Thomas, Bedarra Research Labs
Rob Thomsett, The Thomsett Company Laurence Tratt, King’s College London
Jeffrey Voas, SAIC
Markus Völter, independent consultant John Vu, The Boeing Company

July/August 2007 IEEE SOFTWARE 7