

JOHN OLSON, EDITOR-IN-CHIEF

Moving to a Distributed Architecture



Refreshing – that's how it feels when I see a company easily moving portions of its client/server software to a component server. They've designed right and built right, and now they'll reap the rewards. For developers who spent a little extra time designing, the transition isn't too costly or painful. They create whole new Web interfaces for their programs, but only have to program the GUI. The business logic and processing are already stored as components in their Jaguar server. Hopefully this pertains to you!

Unfortunately, most PB shops are experiencing difficulties moving portions of their code to a middle tier. Most programs weren't written with the business logic separated from the GUI. Proper design was often sacrificed due to short-term and high-pressure scheduling. Now what? Getting there is still possible, but it'll be more time-consuming and costly than for heavily OO, service-based and partitioned programs. Here's a rough guide for moving to a distributed architecture.

1. Identify the logic to be shared. Determine which parts of your code you want to share with the Web interface you're going to build. If you're not building one, proceed as if you were. GUI pieces probably won't be reusable in your application, though some DataWindows might be. Thus your focus will be on business logic and data processing. You should move all the code that processes data into components, but you may want to leave some bits of business logic behind – for example, edit masks, computed columns, validations and other business logic built into your DataWindows. You could move some of the bits to your nonvisual objects (e.g., DW column validations could be put in your `nvo.itemchanged` event, which is delegated to by the DW `itemchanged`), but most are okay where they are. The perfect solution is to reuse your DataWindows and deploy them as Web DataWindows (HTMLDW), which would result in your validations being delivered as JavaScript to the end user's browser. Reusable or not, you have to be sure to duplicate that logic in your new interface.

2. Design a method for transitioning the code. Review each code area and design method to move the code to nonvisual objects – a difficult task. If the existing code is spaghetti-ish, you'll probably end up rewriting it. If it was written in a structured fashion, moving it to nonvisual objects should be easy. Objects originally containing the code would become requestors and delegate control to the nonvisuals rather than doing the work themselves.

3. Test in a client/server environment. Once the code is moved to nonvisuals, test your software. If it behaves the way it did before, you've successfully partitioned your business logic from your GUI. This is where you managers and developers kick yourselves for not having designed and coded it this way in the first place. Don't kick yourself too hard. Design and development methodologies considered "best practices" have changed dramatically over the years. By the time large systems are put into production, the design methodologies originally chosen have usually been relegated to museums.

4. Complete and deploy your components. The nonvisual objects containing the real logic of your system are distributed as Jaguar components. For them to be used by a different interface (e.g., a Web app), you probably need to tweak their public interfaces and internal code and make them into self-contained objects. Deploying these objects in PB7 is a piece of cake – set up some parameters and let PB do the work.

5. Test your n-tier application. Now that your components are distributed, you have a true multitier application. Again, test it until you're satisfied it's ready for production. You should find that your clients run faster because they're smaller and because much of the work is done on the transaction/component server rather than the workstation – one of many benefits to distributing your application. Others are batch processing, asynchronous communications, data sharing, scalability, transaction pooling and easier deployments.

Of course, some features may have driven you to partition your business logic, such as multiple interfaces/front ends that can reuse the business components and can be built using a variety of development tools and languages. Regardless of the language used to build the interfaces, new development can focus on the GUI and business logic doesn't have to be rewritten. This also results in consistent business logic across all interfaces because the same components are used to deliver that logic.

Whether or not you plan on going to a distributed architecture, leave your options open by designing for it now. It won't cost you much extra to do so. If you then choose to build other interfaces or decide you need the performance increases or other benefits of a distributed architecture, the costs and pain of transition will be minimized. ☺

john@sys-con.com

AUTHOR BIO

John Olson is a principal of Developpower, Inc., a consulting company specializing in software solutions using Sybase development tools. He's a CPD professional, charter member of TeamSybase and contributing author of SYS-CON's Secrets of the PowerBuilder Masters books.

DEVELOPER'S PowerBuilder Journal

EDITORIAL ADVISORY BOARD

MICHAEL BARLOTTA, BILL BARTOW, ANDY BLUM,
RICHARD BROOKS, JOE CELKO, KOULOS GORGANI,
BAHADIR KARUV, PH.D., DAVID MCCLANAHAN,
BERNIE METZGER, JOHN OLSON, SEAN RHODY

EDITOR-IN-CHIEF: JOHN OLSON
EXECUTIVE EDITOR: M'LOU PINKHAM
ART DIRECTOR: ALEX BOTERO
PRODUCTION EDITOR: CHERYL VAN SISE
ASSOCIATE EDITOR: NANCY VALENTINE
TECHNICAL EDITOR: BERNIE METZGER
WATCOM SQL EDITOR: JOE CELKO
DATA WINDOWS EDITOR: RICHARD BROOKS

WRITERS IN THIS ISSUE

BRUCE ARMSTRONG, DAVID BRANDOW, RICHARD BROOKS,
BILL BUERMAYER, BRIAN GRIMM, ETHAN HENRY, JENNIFER KOTZEN,
BERNIE METZGER, IDO MILLET, HOYT NELSON, JOHN OLSON,
SREEKUMAR T. SHENOY, PAUL WALLINGFORD

SUBSCRIPTIONS

FOR SUBSCRIPTIONS AND REQUESTS FOR BULK ORDERS,
PLEASE SEND YOUR LETTERS TO SUBSCRIPTION DEPARTMENT

SUBSCRIPTION HOTLINE: 800 513-7111

COVER PRICE: \$14/ISSUE

DOMESTIC: \$149/YR. (12 ISSUES) CANADA/MEXICO: \$169/YR.
OVERSEAS: BASIC SUBSCRIPTION PRICE PLUS AIRMAIL POSTAGE
(U.S. BANKS OR MONEY ORDERS). BACK ISSUES: \$12 EACH

PUBLISHER, PRESIDENT AND CEO: FUJAT A. KIRCAALI
VICE PRESIDENT, PRODUCTION: JIM MORGAN
VICE PRESIDENT, MARKETING: CARMEN GONZALEZ
CHIEF FINANCIAL OFFICER: IGNACIO ARELLANO
ACCOUNTING MANAGER: ELI HOROWITZ
CIRCULATION MANAGER: MARY ANN MCBRIDE
ADVERTISING ACCOUNT MANAGERS: ROBYN FORMA
MEGAN RING
JIDSTORE.COM: JACLYN REDMOND
ADVERTISING ASSISTANT: CHRISTINE RUSSELL
GRAPHIC DESIGN INTERN: AARATHI VENKATARAMAN
SYS-CON RADIO EDITOR: CHAD SITLER
WEBMASTER: ROBERT DIAMOND
WEB SERVICES CONSULTANT: BRUNO Y. DECAUDIN
WEB SERVICES INTERN: DIGANT B. DAVE
CUSTOMER SERVICE: SIAN O'GORMAN
ANN MARIE MILILLO
ONLINE CUSTOMER SERVICE: AMANDA MOSKOWITZ

EDITORIAL OFFICES

SYS-CON PUBLICATIONS, INC.
39 E. CENTRAL AVE., PEARL RIVER, NY 10965
TELEPHONE: 914 735-7300 FAX: 914 735-6547
SUBSCRIBE@SYS-CON.COM

POWERBUILDER DEVELOPER'S JOURNAL (ISSN#1078-1889)
is published monthly (12 times a year) for \$149.00 by
SYS-CON Publications, Inc., 39 E. Central Ave., Pearl River, NY 10965-2306.
Periodicals Postage rates are paid at
Pearl River, NY 10965 and additional mailing offices.
POSTMASTER: Send address changes to:
POWERBUILDER DEVELOPER'S JOURNAL, SYS-CON Publications, Inc.,
39 E. Central Ave., Pearl River, NY 10965-2306.

© COPYRIGHT

Copyright © 1999 by SYS-CON Publications, Inc. All rights reserved.
No part of this publication may be reproduced or transmitted in any form or by any
means, electronic or mechanical, including photocopy or any information storage and
retrieval system, without written permission. For promotional reprints, contact reprint
coordinator, SYS-CON Publications, Inc., reserves the right to revise, republish and
authorize its readers to use the articles submitted for publication.

WORLDWIDE DISTRIBUTION BY CURTIS CIRCULATION COMPANY

739 RIVER ROAD, NEW MILFORD NJ 07646-3048 PHONE: 201 634-7400

All brand and product names used on these pages are trade names,
service marks or trademarks of their respective companies.
SYS-CON Publications, Inc., is not affiliated with the companies
or products covered in PowerBuilder Developer's Journal.

SYS-CON
PUBLICATIONS