



## Convergence of PLM, ALM and Software Product Lines

### Greetings from BigLever:

Over the last few weeks - in preparing for my participation in several upcoming industry conferences - I've begun to further synthesize my thoughts regarding a convergence taking shape among the closely related technologies of Product Lifecycle Management (PLM), Application Lifecycle Management (ALM) and Software Product Lines (SPL).

I believe this convergence offers a glimpse into the future of software product lines in mainstream embedded software development practice. I thought it would be fun to share these ideas with the software product lines community, and if it generates sufficient interest we can start a new topic on the SoftwareProductLines.com discussion forums.

### The PLM / ALM / SPL Convergence

First, let's define each technology arena to ensure a common understanding:

- **PLM** refers to a class of software tools for managing the lifecycle of a product line of mechanical products. PLM systems typically support both the *variation in time* (evolution throughout the lifecycle) and the *variation in space* (variations among the products within a portfolio at any point in time) for a product line. However, PLM systems offer only rudimentary support for software.
- **ALM** refers to an analogous class of software tools for managing the lifecycle of a software product. ALM systems typically support *variation in time* (via a configuration management tool), but do not support variation in space.
- **SPL** development tools provide support for *variation in space* within a software product line portfolio.

The relative strengths of PLM, ALM and SPL converge in near-perfect jigsaw puzzle fashion. PLM's limited support for software can be overcome by ALM tools. ALM's limited support for software portfolio variation in space can be overcome by SPL tools.

The result is a powerful solution for managing the development lifecycle of software-intensive mechanical product line portfolios - or what might be called Product Line Portfolio Lifecycle Management (PLPLM).

>> [More about the convergence](#)

Software Product Lines and their Architectures



### Switching Gears for the Life Cycle: BigLever's product line automation tool ties into requirements systems

By David Rubinstein  
*SD Times* Editor-in-Chief

If you can create an application from parts, then break it down and reassemble those parts in a slightly different manner into a new application - with its components already written, tested and proven - great savings in time, effort and cost can be realized...

>> [See full article](#)

### SD Best Practices: Special VIP Conference Invitation

As a speaker at the upcoming SD Best Practices Conference in Boston, I would like to extend a special VIP Pass invitation for you to join BigLever at a discounted rate. Here is information on the conference, the BigLever session, and how to receive the VIP rate:

### [SD Best Practices Conference & Expo](#)

Hynes Convention Center  
Boston, MA  
September 11-14, 2006

*BigLever session:*

[Introduction to Software Product Line Development Methods](#)

Process & Methods track

Tuesday, September 12, 2006

## Software Product Lines and their Architectures

On another front... I recently received an e-mail query from Paul Clements (co-author of the Software Engineering Institute's book on Software Product Lines). He was having a debate with some colleagues about whether or not a software product line will always have one-and-only-one architecture, or whether one product line can have multiple architectures.

Seeking additional opinions, Paul solicited comments from three SPL development enthusiasts - David Weiss (author of Software Product-Line Engineering: A Family-Based Software Development Process), Rob van Ommering (principal researcher at Philips Research Laboratories) and myself. This invitation incited a highly engaging, detailed discussion on the meanings of "software product lines", "software architecture", "software product populations" and "software product line hierarchies".

The results of the discussion seemed to be comparable to those of a small workshop, so we felt it would be valuable to publish this exchange on the SoftwareProductLines.com forums.

>> [Software Product Lines and their Architectures](#)  
>> [Final summary by Rob](#)

I hope that you find these latest technology explorations useful in stimulating your own thoughts and discussions.

I welcome your comments and feedback regarding the PLM / ALM / SPL convergence, as well as the software product line architectures discussion. I appreciate your continued interest and look forward to continuing this dialogue.

Best Regards,

Charles W. Krueger, BigLever Software CEO  
SoftwareProductLines.com Moderator

512-426-2227  
[ckrueger@biglever.com](mailto:ckrueger@biglever.com)  
[www.biglever.com](http://www.biglever.com)

3:45pm - 5:15pm

To reserve your VIP Pass, register online - enter the priority code 6ESPK81 - by September 6, 2006 and save \$400 on the conference admission price. (You will be asked to submit the priority code in Step 2 of the registration process.)

### About BigLever

BigLever Software, Inc. (TM) is a leading provider of software product line development tools and services. BigLever's patent-pending solution, Gears (TM), dramatically simplifies the creation, evolution and maintenance of embedded or standalone software for a product line.

### Subscriber Functions

[Forward](#)  
[Subscribe](#)  
[Update Info](#)  
[Unsubscribe](#)

### Contact Info

BigLever Software, Inc.  
10500 Laurel Hill Cove  
Austin, TX 78730 USA  
512-426-2227  
[www.biglever.com](http://www.biglever.com)