



## DELAWARE VALLEY PROGRESS USERS GROUP

[HTTP://WWW.DVPUG.ORG](http://www.dvpug.org)

### What is this Document?

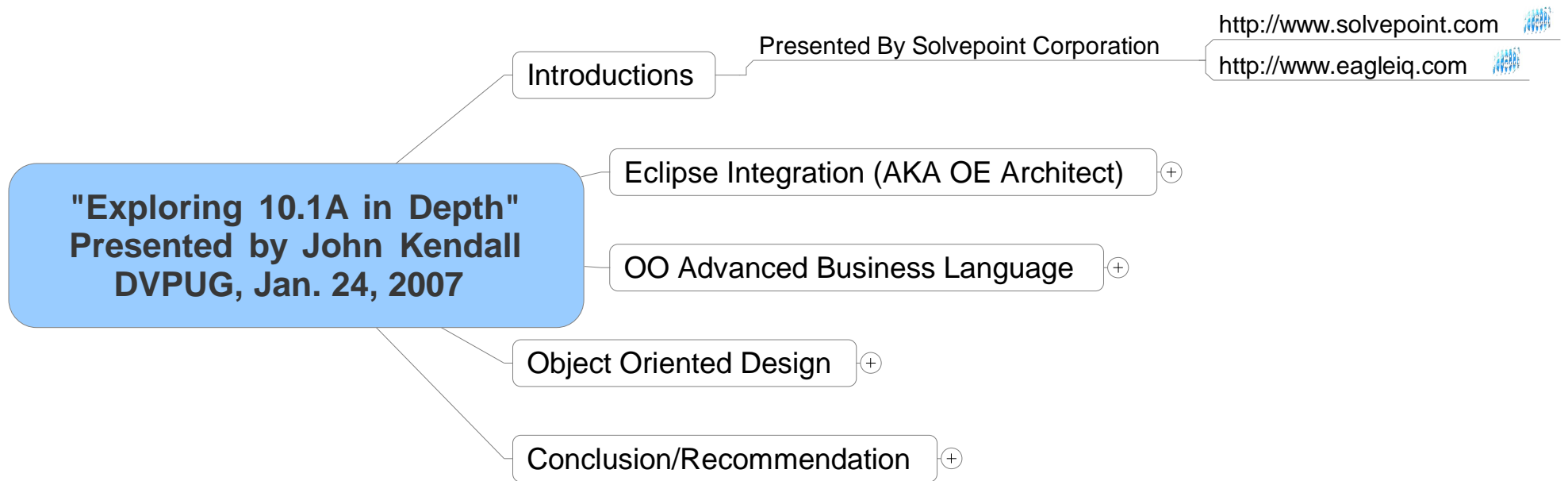
At the January 2007 DVPUG meeting, John Kendall gave a presentation introducing us to some of the new features in PROGRESS OPENEDGE 10.1A as well as some tips on object-orientation, the new Eclipse integration, and more. John used a combination of traditional slides and a mind map created using Mindjet's MindManager software (see <http://www.mindjet.com> for more information) to effectively communicate the content of his presentation. This document is a usable PDF export of that mind map document.

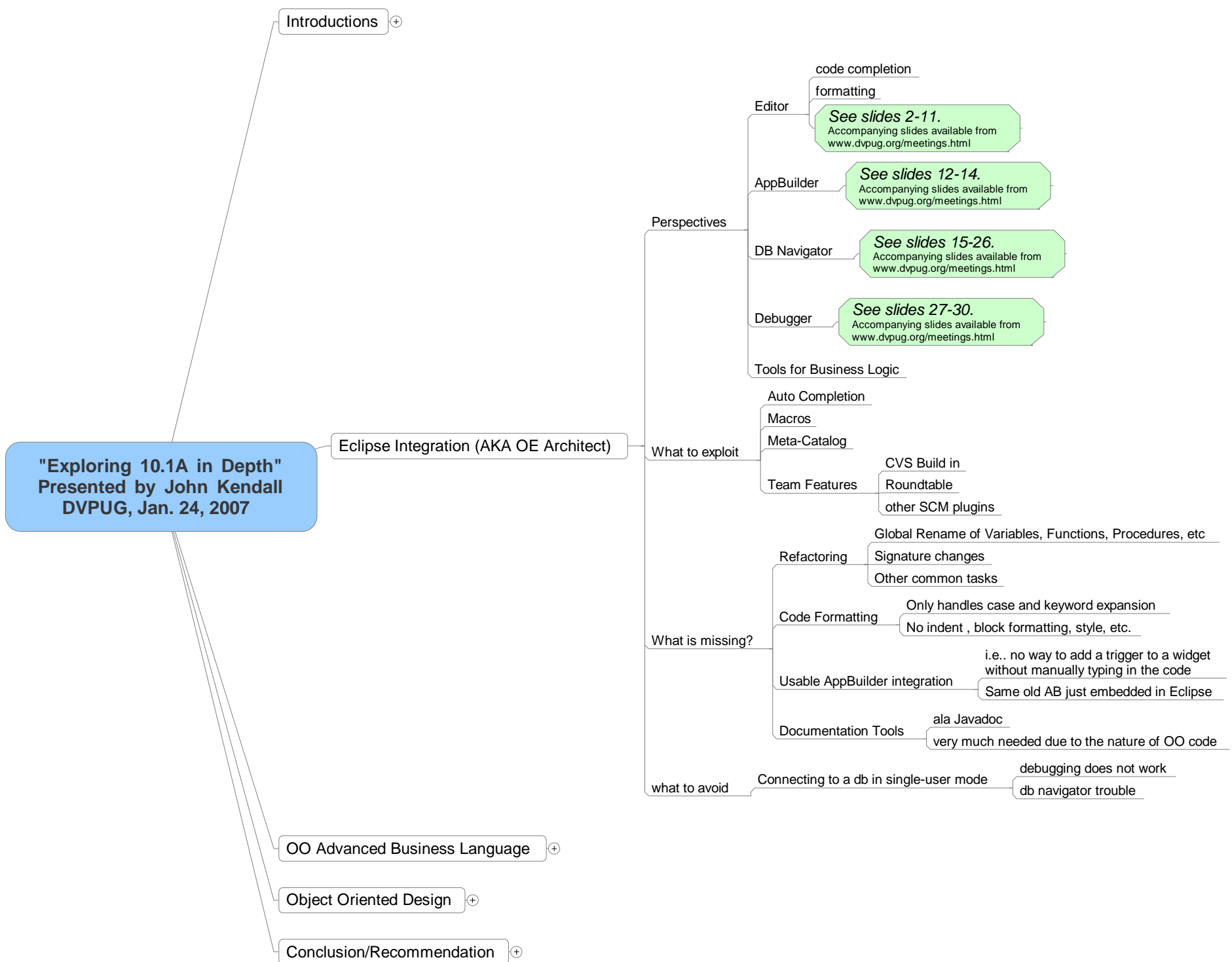
The slides are referenced in this mind map, and are available from the *Past Meetings* section of the DVPUG website at:

<http://www.dvpug.org/meetings.html>

### How Do I Use this Document?

Each of the following pages is a capture of the mind map that John Kendall used in his presentation. Each page is the *same* mind map with different branches exposed for both on-screen and printed media readability: the first page exposes the first few branches of the map, the second page exposes the next few branches, etc. Starting on the first page and reading in succession should provide the reader with a comprehensive read of the map, from the top down.



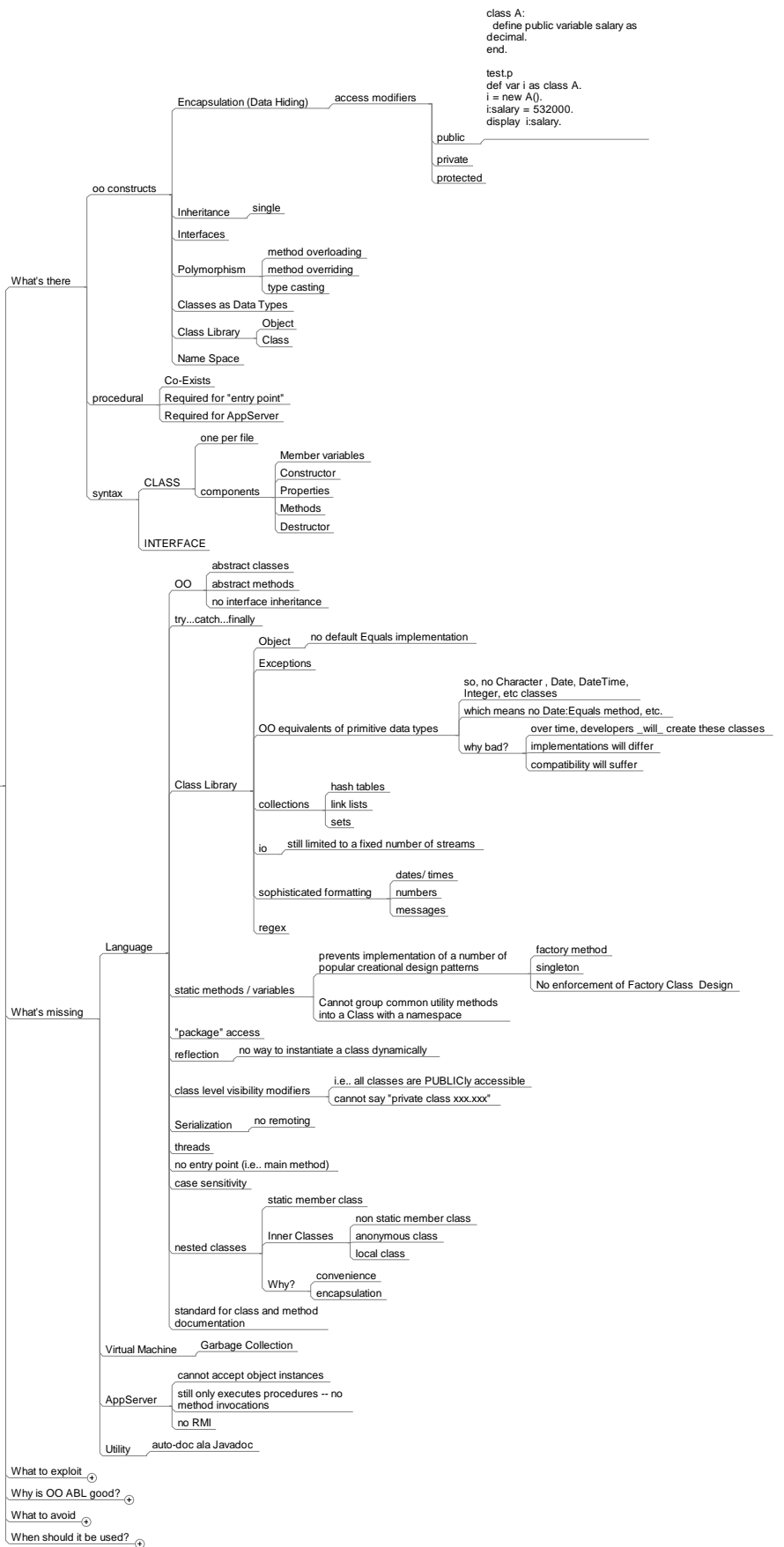


"Exploring 10.1A in Depth"  
Presented by John Kendall  
DVPUG, Jan. 24, 2007

Introductions

Eclipse Integration (AKA OE Architect)

OO Advanced Business Language



Object Oriented Design

Conclusion/Recommendation

**"Exploring 10.1A in Depth"**  
**Presented by John Kendall**  
**DVPUG, Jan. 24, 2007**

Introductions +

Eclipse Integration (AKA OE Architect) +

OO Advanced Business Language

Object Oriented Design +

Conclusion/Recommendation +

What's there +

What's missing +

What to exploit

Plugin Capability of Eclipse

vim !!

Java Perspective (JDT)

Deep thoughts by Jack Handy

``After three days without programming,  
life becomes meaningless."`

Why is OO ABL good?

better encapsulation

improved error handling?

more maintainable code

more easily documented

promote resume through more obvious  
and purposeful responsibility

Strong (compile time) typing

Able to create implementations of popular design patterns

Deep thoughts by Jack Handy

"A well-written program is its own heaven;  
a poorly-written program is its own hell."

" It is easier to write an incorrect program  
than understand a correct one."

What to avoid

complex inheritance hierarchies

favor composition and/or delegation

using classes where a simple procedure will suffice

When should it be used?

To write business functionality that

Should be Encapsulated

Needs to be Re-used across the enterprise

Needs to be enforcing a name space

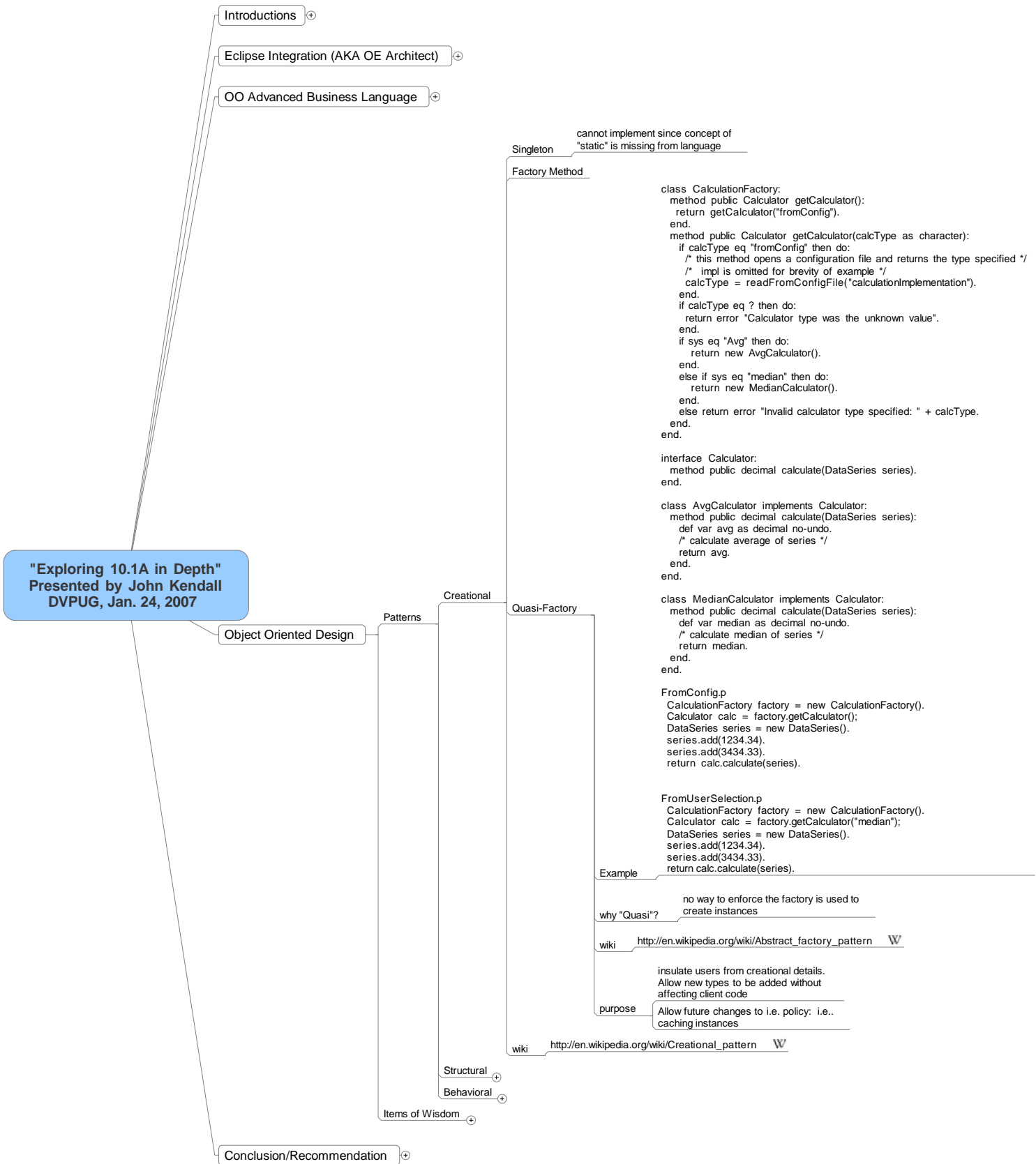
Can benefit from the application of  
popular and useful design patterns

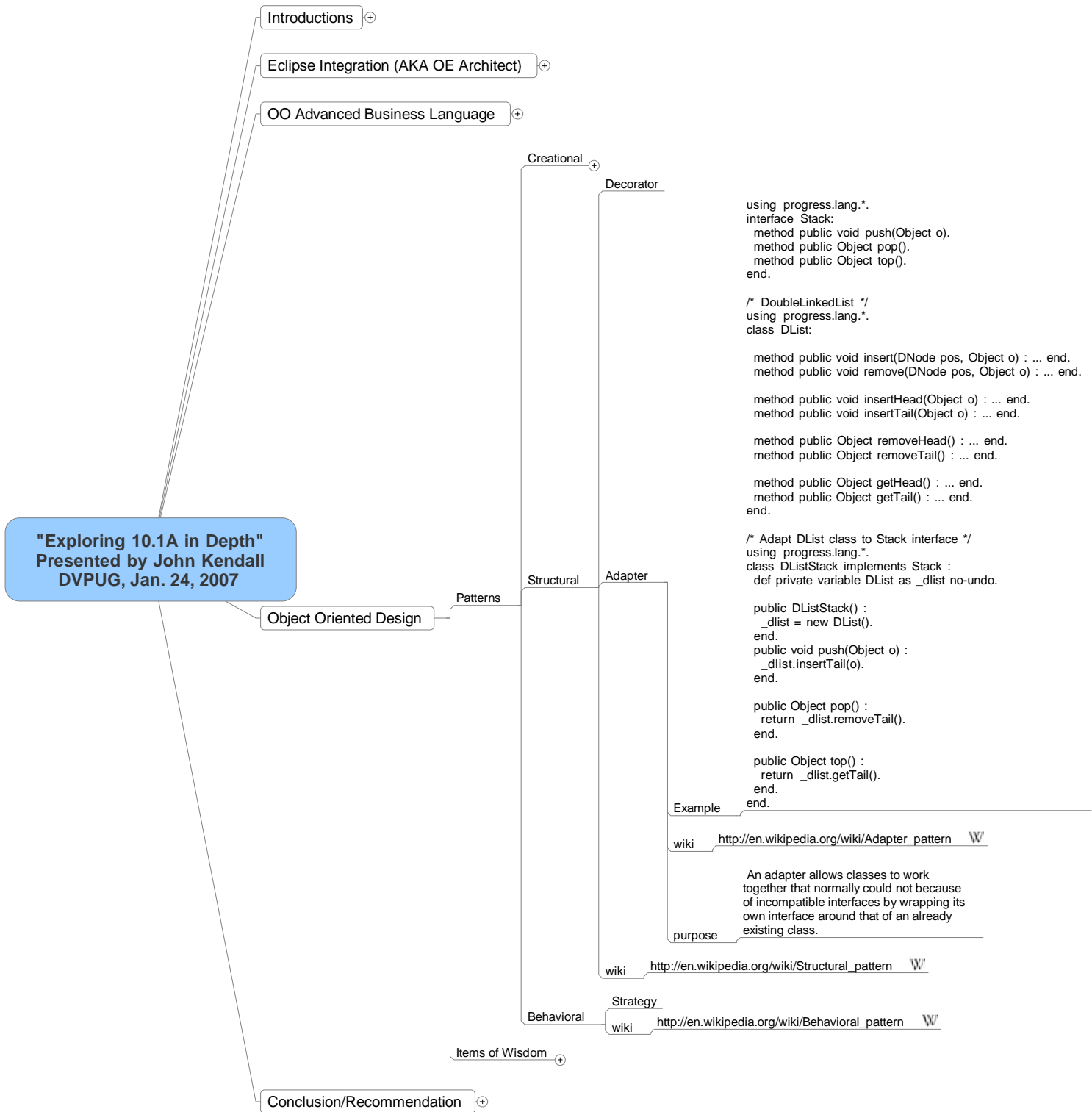
Needs to describe a hierarchy of data  
and/or behavior

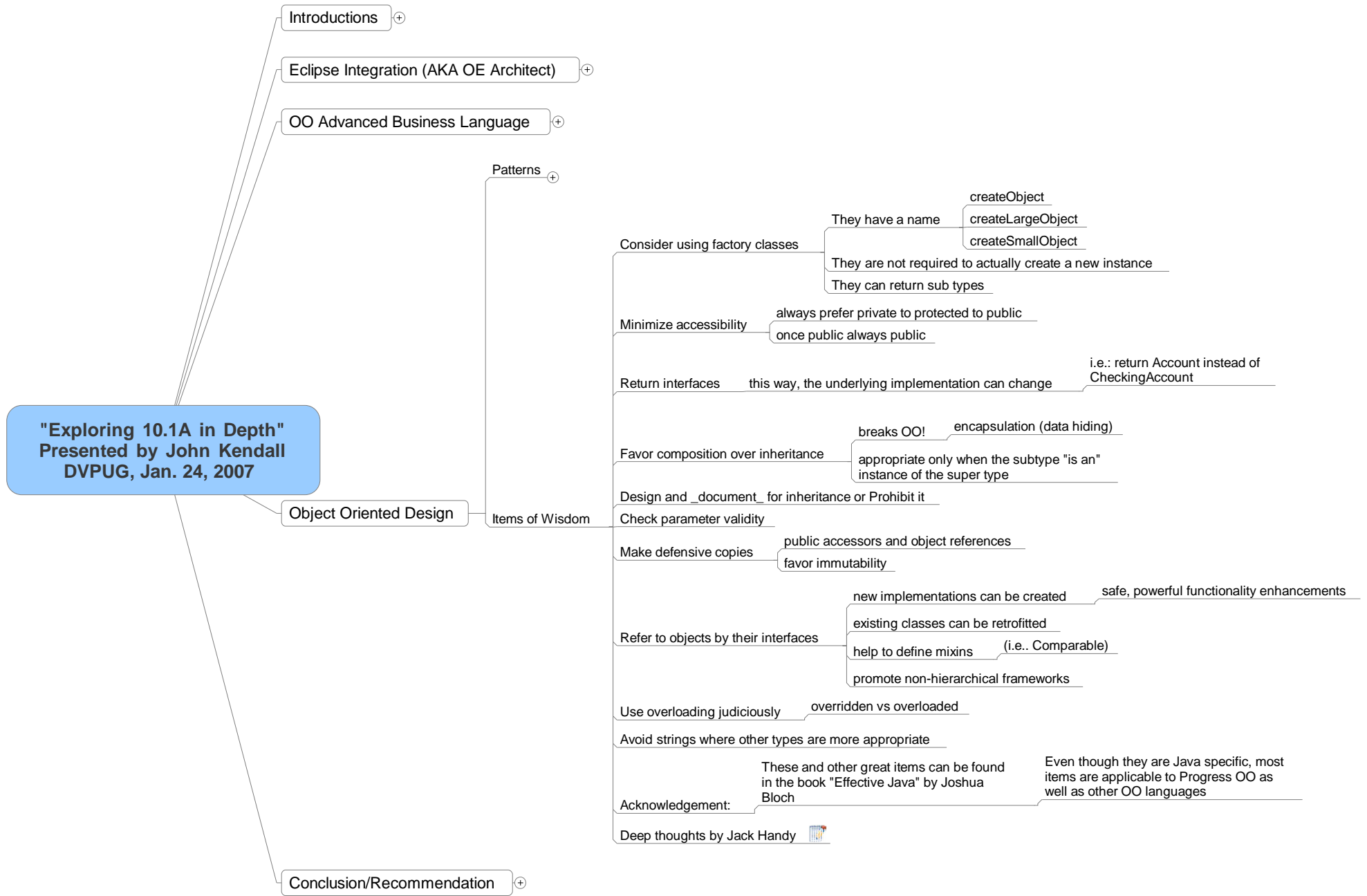
Describes/Implements a sub-system that  
requires (dynamic) extensibility

Promotes well defined interfaces

Promotes adhering to good design principals









**"Exploring 10.1A in Depth"**  
**Presented by John Kendall**  
**DVPUG, Jan. 24, 2007**

Introductions ⊕

Eclipse Integration (AKA OE Architect) ⊕

OO Advanced Business Language ⊕

Object Oriented Design ⊕

Conclusion/Recommendation

Embrace the OO, but realize they are still  
a work in process

Begin to familiarize yourself with OO terminology

Begin to familiarize yourself with Design Patterns

Start learning and playing around in  
Eclipse and/or OE Architect

Download the OE101A Evaluation

<http://www.psdn.com>

