

# Pattern-Driven Software Problem Solving Introduction

Presenter: Dmitry Vostokov  
Memory Dump Analysis Services

# Prerequisites

Experience in software troubleshooting  
and/or debugging

# Agenda (Summary)

- A Short History
- Basic Definitions
- Pattern Categories
- Future Research Directions

# DumpAnalysis.org\*

## 5 Years!

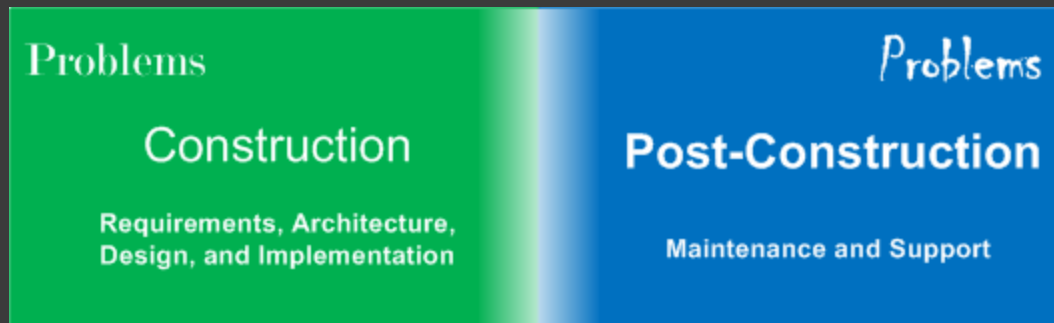
\* **Now Includes:** [TraceAnalysis.org](http://TraceAnalysis.org), [www.ForensicAnalysis.org](http://www.ForensicAnalysis.org), and [Victimware.org](http://Victimware.org)

# Agenda (Basic Definitions)

- ⦿ Software Problems
- ⦿ Software Patterns
- ⦿ DA+TA
- ⦿ Pattern Hybridization

# Software Problems

- Construction: Requirements, Architecture, Design, and Implementation
- **Post-Construction: Maintenance and Support**

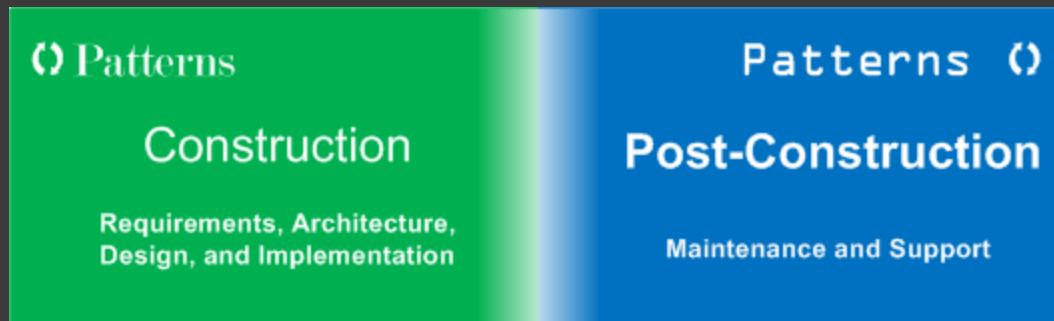


**Post-construction problem:** any observed deviations in structure and behavior between modeling expectations and the actual built system.\*

\* [What is a Software Defect?](#)

# Software Patterns

- Construction: Requirements, Architecture, Design, and Implementation
- Post-Construction: Maintenance and Support**



**Pattern:** a common recurrent identifiable problem together with a set of recommendations and possible solutions to apply in a specific context

# DA+TA

- ◎ DA: Dump Artifact / Dump Analysis

Memory snapshots: process, kernel, physical memory dumps

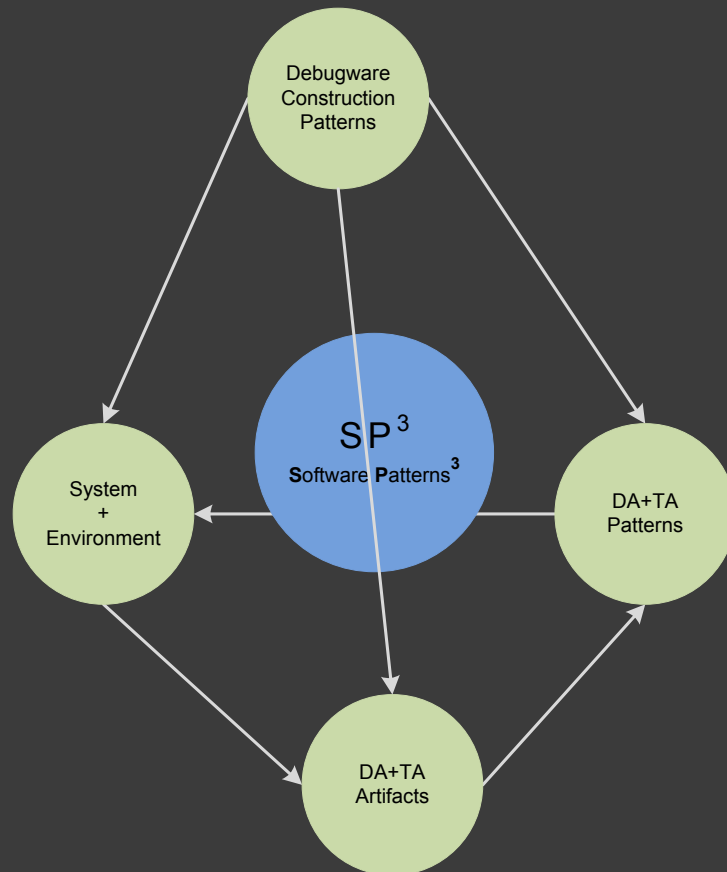
- ◎ TA: Trace Artifact / Trace Analysis

Software traces: Event Tracing for Windows, logs



# Pattern Hybridization

A bit of Chemistry: Carbon  $SP^3$  [orbital hybridization](#)



## Software Patterns<sup>3</sup>

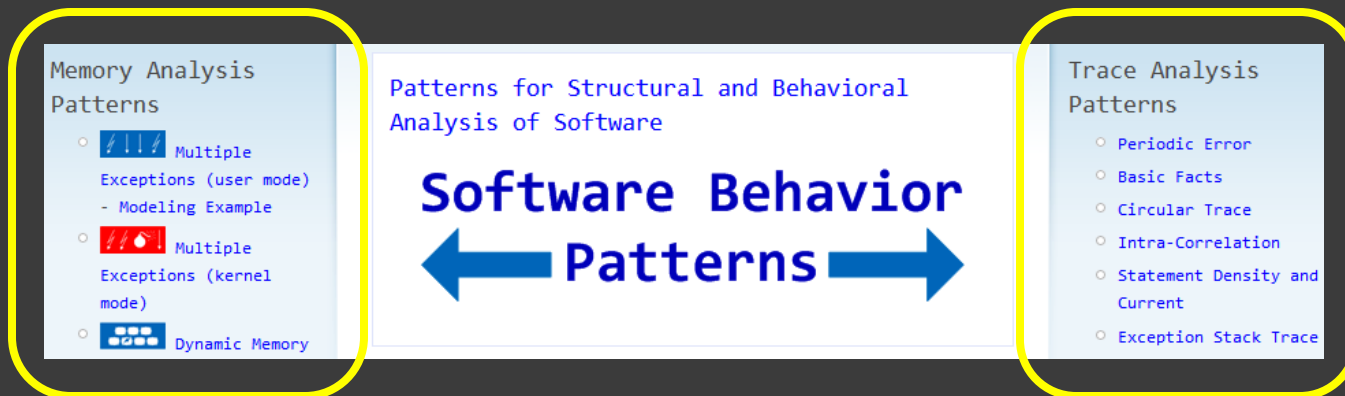
- Structural
- Behavioral
- Debugware
- Debugging

# Agenda (Pattern Categories)

- ④ Software Behavior
- ④ Debugware Tools
- ④ Problem Workaround
- ④ Unified Debugging

# Software Behavior

- Memory dump and software trace
- Examples: [Spiking Thread](#), [Discontinuity](#)
- +200 patterns (DA+TA)
- [DumpAnalysis.org](#)



# DA: Software Behavior

- ◎ Memory dump: a memory snapshot
- ◎ Definition, partial classification and historical list
- ◎ Pattern identification case studies


# TA: Software Behavior

“Imagine you got a software trace from hundreds of modules you haven’t written or haven’t seen source code of...”

- ◎ Software trace: a sequence of memory fragments ordered in time
- ◎ Definition, and historical list
- ◎ Pattern identification case studies

# Debugware Tools

- Troubleshooting and Debugging
- Example: Troubleshooting Unit of Work
- 13 patterns, case study
- DumpAnalysis.org



memory)

- Insufficient Memory (control blocks)
- Spiking Thread
- Module Variety
- Stack Overflow (kernel mode)
- Stack Overflow (user mode)
- Stack Overflow (software implementation)
- Managed Code Exception
- Truncated Dump

Join Facebook DATA (Dump Analysis + Trace Analysis) group:  
<http://www.facebook.com/group.php?gid=95282722070>

## Encyclopedia of Debugging

We announce the start of the ultimate multi- and cross-platform debugging reference project. Useful for years to come, this encyclopedia would cover all aspects of debugging with articles from leading experts from both industry and academia. Preliminary product information:

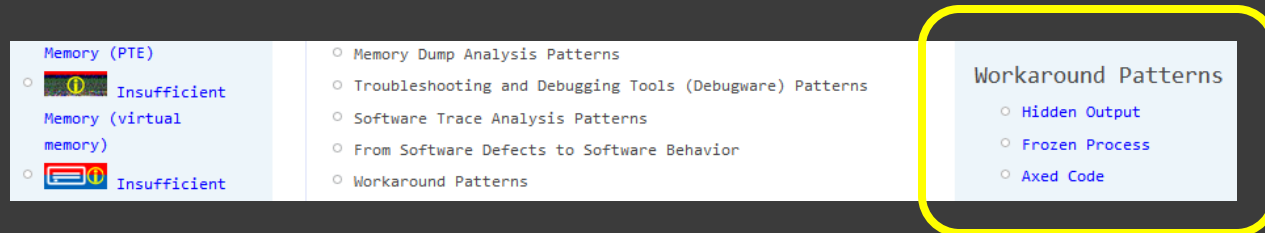
- Title: Encyclopedia of Debugging
- Editor: Dmitry Vostokov
- Hardcover: 1200 pages
- Publisher: OpenTask (March 2014)
- ISBN-13: 978-1-908043-12-2

## DebugWare Patterns

- API Query
- Tool Façade
- Configuration Wrapper
- Dual Interface
- Tool Chain
- Tool Box
- Trace Expert
- Troubleshooting Unit of Work
- Checklist
- Supporting Module
- Span Differentiator
- Self Extractor
- System Description Snapshot

# Problem Workaround

- Troubleshooting when debugging is not an option
- Example: [Frozen Process](#)
- [Patterns](#)
- [DumpAnalysis.org](#)



The image shows a screenshot of a website's navigation menu. On the left, there are three items: 'Memory (PTE)' with a warning icon, 'Memory (virtual memory)' with a warning icon, and another 'Memory (PTE)' with a warning icon. The main menu contains several items: 'Memory Dump Analysis Patterns', 'Troubleshooting and Debugging Tools (Debugware) Patterns', 'Software Trace Analysis Patterns', 'From Software Defects to Software Behavior', and 'Workaround Patterns'. The 'Workaround Patterns' item is highlighted with a yellow rounded rectangle. A sub-menu is visible for 'Workaround Patterns', containing three items: 'Hidden Output', 'Frozen Process', and 'Axed Code'.

- Memory (PTE) Insufficient
- Memory (virtual memory) Insufficient
- Memory (PTE) Insufficient
- Memory Dump Analysis Patterns
- Troubleshooting and Debugging Tools (Debugware) Patterns
- Software Trace Analysis Patterns
- From Software Defects to Software Behavior
- Workaround Patterns
  - Hidden Output
  - Frozen Process
  - Axed Code

# Unified Debugging

- ◎ Systematic Pattern Language
- ◎ Example:

**Analysis Patterns**

**Architectural Patterns**

**Design Patterns**

**Implementation Patterns**

**Usage Patterns**

*Shared Buffer Overwrite*

*Debug Event Subscription / Notification*

*Punctuated Execution*

*Breakpoint (software and hardware)*

*Kernel vs. user space breakpoints*









# Agenda (Future Directions)

- ① Structural Memory Patterns
- ① Unified Debugging Pattern Language
- ① Domain Pattern Hierarchy
- ① Periodic Table of Software Defects

# Structural Memory Patterns

- Examples: [Memory Region](#) and [Region Boundary](#)
- [DumpAnalysis.org](#)

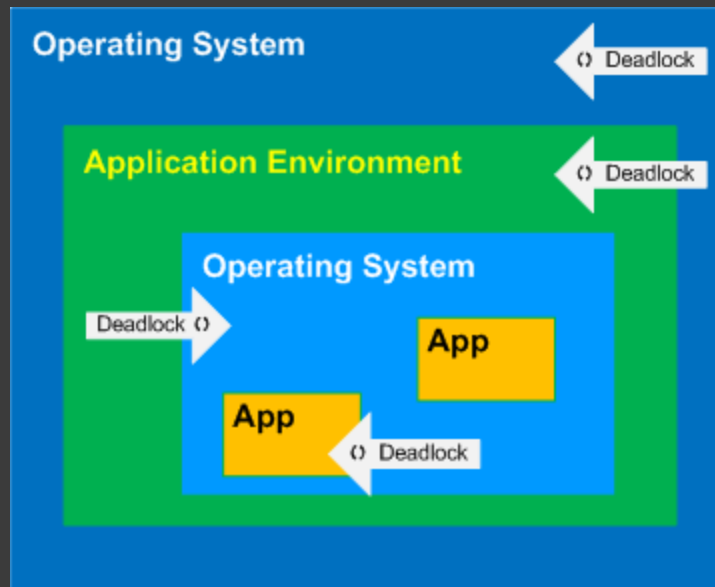
-  OMAP Code Optimization
-  No Component Symbols
-  Insufficient Memory (committed memory)
-  Insufficient Memory (handle leak)
-  Insufficient Memory (kernel pool)
-  Insufficient

## Structural Memory Patterns

- Memory Snapshot (Structured or BLOB)
- Aggregate Snapshot
- Snapshot Collection
- Memory Region (Open or Closed)
- Region Boundary
- Memory Hierarchy (General)
- Anchor Region

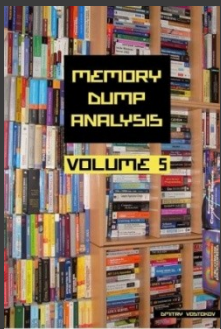
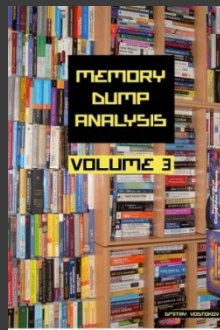
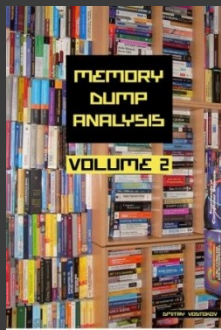
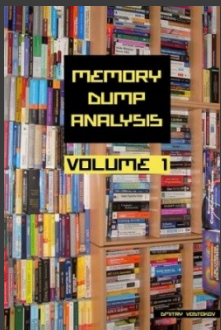
# Domain Pattern Hierarchy

Repeated patterns through the layers of software

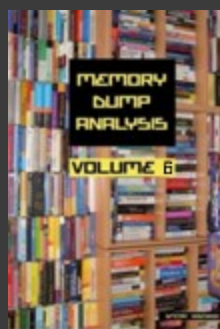


# Resources

- [DumpAnalysis.org](http://DumpAnalysis.org)
- [Pattern-Driven Memory Dump Analysis](#)
- [Memory Dump and Trace Analysis: A Unified Pattern Approach](#)
- [Advanced Software Debugging Reference](#):



Vol 5  
April,  
2011



Vol 6  
November,  
2011

# Q&A

Please send your feedback using the [contact form on DumpAnalysis.com](#)

Thank you for attendance!

[Join DA+TA Facebook Group](#)