

Theoretical Software Diagnostics

Collected Articles

Dmitry Vostokov
Software Diagnostics Institute

Published by OpenTask, Republic of Ireland

Copyright © 2016 by Dmitry Vostokov

Copyright © 2016 by Software Diagnostics Institute

All rights reserved. No part of this book may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, without the prior written permission of the publisher.

You must not circulate this book in any other binding or cover and you must impose the same condition on any acquirer.

OpenTask books are available through booksellers and distributors worldwide. For further information or comments send requests to press@opentask.com.

Product and company names mentioned in this book may be trademarks of their owners.

A CIP catalog record for this book is available from the British Library.

ISBN-13: 978-1-908043-98-6 (Paperback)

First printing, 2016

Revision 1.01 (November 2016)

Preface	13
About the Author	15
Threads of Thinking	17
Volume 1: August 2006 – December 2007	18
Volume 2: January 2008 – September 2008	19
Volume 3: October 2008 – June 2009	20
Volume 4: July 2009 – January 2010	21
Volume 5: February 2010 – October 2010	22
Volume 6: November 2010 – October 2011	23
Volumes 7 – 9 and Beyond: November 2011 – Present	24
Volume 1	25
Multiple Exceptions	25
Four Pillars	28
Five Golden Rules	29
Critical Thinking	30
Troubleshooting as Debugging	31
What is a Software Defect?	33
Four Causes of Crash Dumps	35
What is Memory Dump Analysis?	37

Crashes and Hangs Differentiated	38
Memory Dump - A Mathematical Definition	39
Threads as Braided Strings in Abstract Space	41
Volume 2	45
Debugware Patterns	45
Three Main Ideas of Debugging	46
The Hidden Tomb in Pyramid of Software Change	47
Ceteris Paribus in Comparative Troubleshooting	48
Object-Oriented Debugging and Troubleshooting	49
Component-Based Debugging and Troubleshooting	50
Domain-Driven Debugging and Troubleshooting	51
Causal Models	52
PARTS: Problem Solving Power of Thought	53
Memoretics	54
Memiotics	55
Memory Analysis	56
Volume 3	57
Introduction to Software Trace Analysis Patterns	57
Software Narratology: A Definition	58

Software Trace: A Mathematical Definition	59
Geometrical Debugging	60
Riemann Programming Language	62
The Measure of Debugging and Memory Dump Analysis Complexity	63
I'm RARE	64
T&D Labyrinth	65
Efficient vs. Effective: DATA View	67
A Copernican Revolution in Debugging	68
Is Memory Dump Analysis a Science?	69
Universal Memory Dump: A Definition	70
Quantum Memory Dumps	71
On Subjectivity of Software Defects	72
My Dangerous Idea: Parameterized Science	73
Unique Events and Historical Narratives	74
Chemistry of Virtual Memory	75
Graphical Notation for Memory Dumps	77
Volume 4	81
Succession of Patterns	81

Workaround Patterns	82
Metaphorical Bijectionism: A Method of Inquiry	83
Memory Dumps as Posets	86
MemD Category	88
Operating Closure of Memory	91
Fiber Bundle of Memory Space	93
Manifold Memory Space	94
Extending Multithreading to Multibraiding (Adjoint Threading)	96
Volume 5	101
Software Behavior Patterns	101
Structural Memory Patterns	102
General Memory Analysis	103
Memory Systems Language	104
Notation for Memory and Trace Analysis	105
Models of Software Behavior	106
Category Theory and Troubleshooting	108
Collective Pointer	109
On Unconscious	112
Psychoanalysis of Software Troubleshooting and Debugging	113

Archaeological Foundations for Memory Analysis	114
Software Chorography and Chorology: A Definition	115
Basic Software PLOTs	117
The Extended Software Trace	119
Presenting a Software Story	120
Two Readings of a Software Trace	121
Volume 6	123
Intelligence Analysis Patterns	123
A.C.P. Root Cause Analysis Methodology	124
Function Activity Theory	125
Close and Deconstructive Readings of a Software Trace	126
Analysis, Architectural, Design, Implementation and Usage Debugging Patterns	127
Generative Debugging	128
Metadefect Template Library	129
Orbifold Memory Space	130
Uses of Memoretics	131
Crossdisciplinary Memoretics as Interdisciplinary Science	132
Coarse vs. Fine Grained DNA of Software Behavior	133

The Way of Philip Marlowe: Abductive Reasoning for Troubleshooting and Debugging	134
The New School of Debugging	135
A Periodic Table of Software Defects	136
User Interface Problem Analysis Patterns	137
Volume 7	139
Software Anti-Narrative	139
Narremes in Software Narratology	140
Narralog - A Software Trace Modeling Language	141
What is a Software Narrative?	142
Software Narrative Planes	143
Software Narratology Square	145
Software Trace Analysis Patterns Domain Hierarchy	146
Generalized Software Narrative and Trace	147
Unified Computer Diagnostics: Incorporating Hardware Narratology	148
Introducing Software Narratology of Things (Software NT)	149
What are Software Trace and Memory Dump Analysis? A One Sentence Definition	150
Software Problem Description Language	151
Software Problem Description Patterns	152

Software Behavior Pattern Prediction	153
Patterns of Software Diagnostics	154
Highly Effective Diagnostics	155
Network Trace Analysis Patterns	156
Pattern-Based Software Diagnostics	159
Software Diagnostics Discipline	160
Architecture of memCPU	161
Phenomenology of Software Diagnostics: A First Sketch	162
Software Diagnostics Report Schemes	163
Software Diagnostics Training: Two Approaches	164
The Structure of Software Problem Solving Organization	165
Software Disruption Patterns	166
Static Code Analysis Patterns	167
Bridging the Great Divide	168
Elementary Software Diagnostics Patterns	169
Zero Fault Software Diagnostics	170
Agile Software Diagnostics	172
ADDR Pattern Catalog	173
Thinking-Based Software Diagnostics	174



Memory Acquisition Pattern Catalog	175
Trace Acquisition Pattern Catalog	176
Patterns of Software Diagnostics Architecture	177
Detecting and Predicting the Unknown	179
Software Diagnostics as Psychology	180
Software Diagnostics as Literary Criticism	181
Rapid Software Diagnostics Process (RSDP)	182
Right First Time Software Diagnosis	183
Software Diagnosis Codes	184
Vulnerability Analysis Patterns (VAP)	185
Analytic Memory Dump - A Mathematical Definition	187
General Abnormal Patterns of Structure and Behavior	188
Malware Analysis Patterns	189
Software Trace Diagrams (STDiagrams)	190
Volume 8	195
A Pattern Language for Performance Analysis	195
The Timeless Way of Diagnostics	196
Pattern-Oriented Debugging Process	198
Malnarratives	201

Higher-Order Pattern Narratives (Analyzing Diagnostic Analysis)	203
Special and General Trace and Log Analysis	207
Projective Debugging	210
Pattern! What Pattern?	218
I Didn't See Anything	221
Coding and Artcoding	223
Adjoint Space	225
Volume 9	229
Diagnostics, Forensics, Prognostics: The Copernican Revolution	229
Pattern Repertoire	231
Pattern-Oriented Software Internals: Pattern Paradigms and Software Internals Pattern Stack	233
Software Diagnostics Canvas	237
Software Traces and Logs as Proteins	239
Patterns-Based Root Cause Analysis Methodology	240
Teaching Complex Diagnostic Scenarios with Artificial Debugger (ArtDbg) and Pseudo-Memory Dumps	244
The Scope of Software Diagnostics	246
Diagnostics of Things (DoT)	248
Riemann Root Cause Analysis Language	249



Problem Solving as Code	252
Diagram Graphical Diagnostic Analysis Language	253
Iterative Pattern-Oriented Root Cause Analysis	255
Theoretical Software Diagnostics and Education	257
Volume 10	259
Topological Software Trace and Log Analysis	259
Software Diagnostic Space as a General Graph of Software Narratives	260
Software Diagnostics as Archaeology	265
Unpublished	267
Functionalist Trace Analysis	267
Notes	269

Preface

This book reprints selected articles from Memory Dump Analysis Anthology volumes 1 – 9 related to theoretical aspects of pattern-oriented software diagnostics. Some articles from the forthcoming volume 10 are also included.

If you encounter any error, please contact me using this form:

<http://www.dumpanalysis.org/contact>

or send me a personal message using this contact e-mail:

dmitry.vostokov@dumpanalysis.org

Alternatively, via Twitter @DumpAnalysis

Facebook pages and group:

<http://www.facebook.com/DumpAnalysis>

<http://www.facebook.com/TraceAnalysis>

<http://www.facebook.com/groups/dumpanalysis>

The Software Diagnostics Group on LinkedIn:

<https://www.linkedin.com/groups/8473045>



[This page is intentionally left blank]

About the Author



Dmitry Vostokov is an internationally recognized expert, speaker, educator, scientist and author. He is the founder of pattern-oriented software diagnostics, forensics and prognostics discipline and Software Diagnostics Institute (DA+TA: DumpAnalysis.org + TraceAnalysis.org). Vostokov has also authored more than 30 books on software diagnostics, forensics and problem-solving, memory dump analysis, debugging, software trace and log analysis, reverse engineering, and malware analysis. He has more than 20 years of experience in software architecture, design, development and maintenance in a variety of industries including leadership, technical and people management roles. Dmitry also founded [DiaThings](http://DiaThings.com), [Logtellelect](http://Logtellelect.com), [OpenTask Iterative](http://OpenTask.com) and [Incremental Publishing \(OpenTask.com\)](http://IncrementalPublishing.com), [Software Diagnostics Services \(former Memory Dump Analysis Services\)](http://SoftwareDiagnostics.com) PatternDiagnostics.com and [Software Prognostics](http://SoftwarePrognostics.com). In his spare time, he presents various topics on Debugging.TV and explores [Software Narratology](http://SoftwareNarratology.com), an applied science of software stories that he pioneered, and its further development as [Narratology of Things](http://NarratologyOfThings.com) and [Diagnostics of Things \(DoT\)](http://DiagnosticsOfThings.com). His current area of interest is theoretical software diagnostics.



[This page is intentionally left blank]

Threads of Thinking

Here we chronologically review articles that started various threads of thinking. We use periodization based on time spanned by each Memory Dump Analysis Anthology volume. Articles within each volume may not be reprinted chronologically because we rearranged them according to related topics.

Volume 1: August 2006 – December 2007

The article *Multiple Exceptions* published on the 30th of October, 2006 opened the thread of crash dump analysis patterns. There are more than 300 such patterns at the time of this writing with examples from Windows, Mac OS X, and Linux (October 2016).

After crash dump analysis pattern innovation, the author started thinking about terminology, methodological principles, and questions related to software technical support, troubleshooting, causes of software problems, general principles of artifact analysis:

- [Four Pillars](#)
- [Five Golden Rules](#)
- [Critical Thinking](#)
- [Troubleshooting as Debugging](#)
- [What is a Software Defect?](#)
- [Four Causes of Crash Dumps](#)
- [What is Memory Dump Analysis?](#)
- [Crashes and Hangs Differentiated](#)

During that time the author started applying mathematical ideas to software execution artifact analysis:

- [Memory Dump - A Mathematical Definition](#)
- [Threads as Braided Strings in Abstract Space](#)

Volume 2: January 2008 – September 2008

On the 19th of July 2008, the author published the first pattern from [Debugware Patterns](#) thread as an effort to bring pattern-oriented design approaches to chaotic software troubleshooting and debugging tool development prevalent in software companies. UML notation was used for describing many such patterns.

The author continued thinking about software technical support, troubleshooting, debugging, root cause analysis, and the role of memory dump analysis:

- [Three Main Ideas of Debugging](#)
- [The Hidden Tomb in Pyramid of Software Change](#)
- [Ceteris Paribus in Comparative Troubleshooting](#)
- [Object-Oriented Debugging and Troubleshooting](#)
- [Component-Based Debugging and Troubleshooting](#)
- [Domain-Driven Debugging and Troubleshooting](#)
- [Causal Models](#)

[PARTS: Problem Solving Power of Thought](#) was the first attempt to create problem analysis and resolution methodology.

Recognizing the need to name the emerging discipline the author proposed [Memoretics](#), [Memiotics](#), and [Memory Analysis](#) classification that included memory forensics and intelligence.

Volume 3: October 2008 – June 2009

[Introduction to Software Trace Analysis Patterns](#) opened the thread of trace and log analysis patterns. There are more than 130 such patterns at the time of this writing (October 2016). Such pattern-oriented trace and log analysis thinking led to [Software Narratology: A Definition](#).

The author also attempted to unify software traces and memory dump, and continued to apply mathematical ideas:

- [Software Trace: A Mathematical Definition](#)
- [Geometrical Debugging](#)
- [Riemann Programming Language](#)
- [The Measure of Debugging and Memory Dump Analysis Complexity](#)

A few articles continued the thinking thread of problem analysis, troubleshooting, debugging, analysis reporting, efficient and effective strategies, scientific thinking and the foundational role of memory dump analysis:

- [I'm RARE](#)
- [T&D Labyrinth](#)
- [Efficient vs. Effective: DATA View](#)
- [A Copernican Revolution in Debugging](#)
- [Is Memory Dump Analysis a Science?](#)
- [Universal Memory Dump: A Definition](#)

The author also continued to apply ideas from other disciplines:

- [Quantum Memory Dumps](#)
- [On Subjectivity of Software Defects](#)
- [My Dangerous Idea: Parameterized Science](#)
- [Unique Events and Historical Narratives](#)
- [Chemistry of Virtual Memory](#)
- [Graphical Notation for Memory Dumps](#)

Volume 4: July 2009 – January 2010

[Succession of Patterns](#) introduced a pattern-oriented approach to root cause analysis.

The new [Workaround Patterns](#) catalog was added to pattern-oriented problem solving.

The author also continued applying mathematical concepts to software artifact analysis:

- [Metaphorical Bijectionism: A Method of Inquiry](#)
- [Memory Dumps as Posets](#)
- [MemD Category](#)
- [Operating Closure of Memory](#)
- [Fiber Bundle of Memory Space](#)
- [Manifold Memory Space](#)
- [Extending Multithreading to Multibraiding \(Adjoint Threading\)](#)

Volume 5: February 2010 – October 2010

The author introduced [Software Behavior Patterns](#) and [Structural Memory Patterns](#) together with pattern unification, the new pattern language and experimental notation:

- [General Memory Analysis](#)
- [Memory Systems Language](#)
- [Notation for Memory and Trace Analysis](#)

[Models of Software Behavior](#) were also introduced as a useful tool for building pattern catalogs.

The author continued to apply ideas from other disciplines:

- [Category Theory and Troubleshooting](#)
- [Collective Pointer](#)
- [On Unconscious](#)
- [Psychoanalysis of Software Troubleshooting and Debugging](#)
- [Archaeological Foundations for Memory Analysis](#)
- [Software Chorography and Chorology: A Definition](#)

Software narratology was also further explored:

- [Basic Software PLOTs](#)
- [The Extended Software Trace](#)
- [Presenting a Software Story](#)
- [Two Readings of a Software Trace](#)

Volume 6: November 2010 – October 2011

This period continued the gestation of pattern-oriented software diagnostics with the recognition of its systemic aspect before the explosion of ideas during its maturation period that started in 2012.

The most important articles from this period are about pattern-oriented root cause analysis, unified debugging pattern language, and software defect periodicity:

- [A.C.P. Root Cause Analysis Methodology](#)
- [Analysis, Architectural, Design, Implementation and Usage Debugging Patterns](#)
- [A Periodic Table of Software Defects](#)

[Intelligence Analysis Patterns](#) catalog was also inceptioned during that period.

Volumes 7 – 9 and Beyond: November 2011 – Present

The formative years when [Software Diagnostics Discipline](#) was defined including its scope and most theoretical work was done as well as the majority of trace and log analysis [Pattern Repertoire](#). During 2016, the accumulated mathematical ideas gave rise to [Topological Software Trace and Log Analysis](#) and its unification with the previous software narratological work by the introduction of [Software Diagnostic Space as a General Graph of Software Narratives](#).

Many more pattern catalogs were added:

- [Software Problem Description Patterns](#)
- [Patterns of Software Diagnostics](#)
- [Network Trace Analysis Patterns](#)
- [Software Disruption Patterns](#)
- [Static Code Analysis Patterns](#)
- [Elementary Software Diagnostics Patterns](#)
- [ADDR Pattern Catalog](#)
- [Memory Acquisition Pattern Catalog](#)
- [Trace Acquisition Pattern Catalog](#)
- [Patterns of Software Diagnostics Architecture](#)
- [Vulnerability Analysis Patterns \(VAP\)](#)
- [Malware Analysis Patterns](#)
- [A Pattern Language for Performance Analysis](#)

We consider the introduction of mechanisms as the most important addition to [Patterns-Based](#) Root Cause Analysis Methodology which is a part of pattern-oriented problem-solving methodology in which diagnostics plays a major part compared to many black box problem analysis and resolution methods.