

Logic Online Inc.

RIPPLE-TRAC ADVANTAGES



©

Written by
Daniel B. Boucher Sr,
z/OS Solutions Architect,
Logic Online, Inc.

A Separation of Concerns (SoC) Tool

www.logiconlineinc.com

© 2011 Logic Online Inc ALL RIGHTS RESERVED.

This document contains proprietary information, protected by copyright. No part of this document may be reproduced or transmitted for any purpose other than the reader's personal use without the permission of Logic Online Inc.

WARRANTY

The information contained in this document is subject to change without notice. Logic Online makes no warranty of any kind with respect to this information. Logic Online shall not be liable for any direct, indirect, incidental, consequential, or other damage alleged in connection with the use of this information.

TRADEMARKS

All trademarks and registered trademarks used in this guide are property of their respective owners.

RIPPLE-TRAC is a registered trademark in the state of New Hampshire

Logic Online Inc.
1500A Lafayette Rd #170
Portsmouth, NH 03801
e-mail: dan.boucher@logiconlineinc.com
www.logiconlineinc.com



www.logiconlineinc.com is a resource for:



<http://www-03.ibm.com/systems/z/destinationz/>

Section 1 Overview

Multi-dimensional separation of concerns is an approach to separation of concerns, supporting construction, evolution and integration of software. Its goals are to enable:

- Encapsulation of all kinds of concerns in a software system, simultaneously.
- Overlapping and interacting concerns.
- On-demand modularization.

Separation of concerns is a concept that is at the core of software engineering. It refers to the ability to identify, encapsulate, and manipulate those parts of software that are relevant to a particular concern (concept, goal, purpose, etc.). Concerns are the primary motivation for organizing and decomposing software into manageable and comprehensible parts. Many kinds of concerns may be relevant to different developers in different roles, or at different stages of the software lifecycle. Appropriate separation of concerns has been hypothesized to reduce software complexity and improve comprehensibility; promote traceability; facilitate reuse, non-invasive adaptation, customization, and evolution; and simplify component integration.

The term *multi-dimensional separation of concerns* (MDSOC) refers to flexible and incremental separation, modularization, and integration of software artifacts based on any number of concerns. It overcomes limitations of existing mechanisms by permitting clean separation of multiple, potentially overlapping and interacting concerns simultaneously. MDSOC promotes reuse, improves comprehension, reduces the impact of change, eases maintenance and evolution, improves traceability, and opens the door to system refactoring and reengineering.

MDSOC summary:

Involves decomposition of software according to one or more dimensions of concern. A concern is any piece of interest or focus in a program. http://en.wikipedia.org/wiki/Separation_of_concerns

The separation allows:

- To allow people to work on individual pieces of the system in isolation;
- To facilitate reusability;
- To ensure the maintainability of a system;
- To add new features easily;
- To enable everyone to better understand the system;
- To allow support for multi-dimensional separation of concerns.

Remember, a dimension of concern is simply an approach to decomposing, organizing, and structuring software according to concerns of a particular kind. **RIPPLE-TRAC** falls into the realm of multi-dimensional separation of concerns.

Business Advantage of RIPPLE-TRAC (RT)

RIPPLE-TRAC is named for the Ripple effect -a series of repercussions or consequences of change particularly appropriate to the software industry where the effects of one event set off other unexpected events.

The technology engaged by RIPPLE-TRAC presents the information from the point of view of the concern – the architect only sees information that is related to the concern and has control on what concerns should be brought to the forefront or obscured into the background. The uniqueness of RIPPLE-TRAC is the targeted approach, which we believe has potential to support longer term refactoring of application logic into reusable components and services (SOA). RIPPLE-TRAC can also be used as a support tool for migrating from one platform to another.

ADVANTAGES:

1 No Additional Hardware Required.

2. Assessment of Change.

- Know the full impact for the project before starting a project.
- Formulate a cohesive blueprint of the enterprise.
- Make business decisions quickly and with resolve.
- Prevent teams from going off on parallel and conflicting paths.
- Allow developers see outside the box instead of having a limited sphere of vision.
- Provide cross-disciplinary, cross-functional transparency and allow developers to rapidly visualize interdependent components working on related projects.
- Prevent stymied efforts to craft and deploy efficient, cost effective solutions.
- Allow the development teams to zoom in on progressively granular views of issues and possible solutions without being blurred or off target.

3. Reliability and Maintainability.

- Know the number of modifications, where located and complexity involved.

4. Job Satisfaction.

- Developer will feel in control by having a map and reliable information that guides their effort.

5. Productivity.

"Know what you're taking on before you're in the middle of an effort and find significant unknowns."

- Simultaneously search up to 500+ concerns of interest.
- Minimum learning curve
- Each developer will be able to work and address modification more effectively.
- Estimate Project and work assignments correctly.
- Determine the full project impact and scope before starting.

6. Competitive Advantage.

- No monitoring agent left on the mainframe to consume resources.
- Set criteria of the project concerns of interest, once and RT delivers all the information needed to assess, assign and manage the project.
- Flat pricing, no limit on concurrent users on search technology.

7. Business Logic.

- Highlights and identifies business rules implemented in the code.
- Code can be extracted.
- Code can be flowcharted individually or batched.
- Complexity of the code determined and evaluated.
- Allows for the editing of business rules graphically in flowchart mode.

8. Regulatory Requirement Support.

- Document the implemented business logic as opposed to the business rules believed to be in place.
- Know what is running in production as opposed to what you think is running in production.

9. Data Model.

- Present Data Modeler with RIPPLE-TRAC table result set for the modeling tool and production DDL updates.

10. SOA Service Oriented Architect.

- Finds and lists the SOA method names.
- Lists where the method is used.
- Help maximize service reliability, scalability and performance.

11. DB2/IMS Performance Assessment and Tuning.

- Search up to 500+ predicate string arguments simultaneously.
- Track any instruction via the external argument table.
- Manually retrieve metadata from the modeling tool – table/segment names, description of table/segments and load them into RT.
- Track where used for tables/segments.
- Track where used field names in both DB2/IMS and COBOL (COBOL - vs _ DB2 naming convention).
- Isolate business rule established in the predicate or search condition.
- Isolate predicates and analysis performance based on how predicates are built and sequenced.
- Prioritizes parameter for performance base on the significance of parameters used. (DB2 only)
- Track BIND PARAMETERS as well as create new bind parameters with our optional spreadsheet guide(DB2 only)

- Produces result set to review with the DB administrator to Examine DB physical layer, indexes, buffer pools and data.
- Retrieve data from DBDGEN, PSBGEN

12. Data Warehouse Benefits.

- Reconcile components across Silos.
- Can augment Data profiling.
- Migration; help capitalize on Strategic Business Investments While Minimizing Risk
- Identify redundancies, differences and cross organization concerns.
- Provide a reliable consolidated report and analysis for code reuse.
- Bridges physical difference and creates a single view.
- Identifies common file used in a corporate system.
- Highlights cross project conflicts.
- Promote code reuse and understanding of existing business rules.
- Prevent conflicts between organizations – and provide support for Master Data Management.

13. Operating System Analysis.

- At system level, examine where macros are used and examine sysgen configurations across LPAR's.
- Assess and manage the impact of making a change.
- Failure modes analysis is used to identify and order the impediments to reuse in a given LPAR.

14. VSAM.

Locate any imbedded recovery points for logical record such as rollback, syncpoint, commit, etc.

15. CICS, Performance Analyses.

- Extract any CICS command.
- Automatically extract the CICS command value without knowing what it is.

16. JCL Decomposition Product

- Produce DSN where used across entire system.

17. Data Quality Issues Can Be Tracked To the Application

18. Identify Performance Issues that Can Be Tracked to the Application and System Level.

19. Supported code and Environments

Supported	Code	Flow Chart Capable
ASSEMBLER	YES	NO
BIND	YES	NO
BMS	YES	NO
C / C++	YES	YES
CICS	YES	YES
COBOL	YES	YES
DB2 PREDICATE	YES	YES
DB2 TABLE	YES	YES
DDL	YES	NO
FORTRAN	YES	YES
IDMS	YES	YES
IMS	YES	YES
JAVA	YES	YES
PASCAL	YES	YES
PL/1	YES	YES
POWERBUILDER	YES	* YES
REXX	YES	YES
RPG	YES	NO
SOA	YES	YES
SORT	YES	NO
SYS1.PARMLIB	YES	NO
VSAM	YES	YES
XML	YES	NO
z/OS JCL	YES	YES
*	upload to mainframe	