SIBRA Mainframe Tools and Professional Services

Eberhard Ramm

SIBRA Ingenieurgesellschaft für Datentechnik mbH
Industriestrasse 35
D-82194 Gröbenzell, Germany
Phone: +49 (0)8142 57264
Fax: +49 (0)8142 57265
email: SibraGmbh@t-online.de
Internet: www.SibraGmbh.com

Table of Contents, Overview

foils

Special Experience and Know How

- SIBRA has special experience and know how with
  - Analysis of existing applications on all platforms for optimal performance and reduction of running costs
  - Analysis and further development of existing solutions
  - Redesign and renovation of existing applications
  - Accessibility of existing applications onto Internet
  - New implementation of individual applications
  - Integration into available system environments
  - Production of prototype versions
  - System tuning and customizing (tailoring)
    for z/OS, Compiler, LE/390, CICS, IMS, DB2, IDMS, ...

Special Experience and Know How

- Platform and database independent migration of applications and data resources, e. g. Mainframe à SAP/R3
- Technology Consulting
- Euro conversions without Big Bang procedure with our SIBRA tools PROST and C_PROST
- SIBRA tool RelMan Dynamic Release Management: A dynamically controlled program execution within a given date and functionality by deadline dependent program object properties
- Adaptation of computer centers on LE/390
- Migration to newer compiler versions, for example IBM Enterprise Cobol and PL/1

Offered Tools and Professional Services

- Improvement of Application Development, Program Quality and Performance
  - Analysis of currently running Load Module Libraries for
    Enterprise, Application Portfolio, and Program Management
  - Building up a DB2 based Load Module Repository LMR from currently running Load Module Libraries for the usage as a
    Configuration Management System

Offered Tools and Professional Services

- Migration to current Language Environment /390 Version (LE/390) without Compile and Link for
  Standardization to only one Runtime Environment and Version for Cobol, Fortran, PL/I and C/C++
- Migration to current Language Environment /390 is prerequisite for
  Flexible migration to current LE-compliant compiler versions, e.g. Enterprise Cobol & PL/I

Offered Tools and Professional Services

- Usage of additional Test Features of LE/390 to improve program quality
- Special Feature: Automated Relink
- CWB Conversion Workbench, the Source Code Repository with many migration and conversion functions for all your applications and your JCL that can be implemented in different programming languages and on different system environments. For more information: http://www.sibragmbh.com/Current/E/StartFrames.html

Offered Tools and Professional Services

- RelMan, the dynamic Release Management, that builds up on the known and well-proved concept of EURO-PROST and C_PROST, which allows a step-by-step adaptation procedure and avoids a program technical Big Bang.
  For more information:
  http://www.sibragmbh.com/Current/E/StartFrames.html

Analysis of Load Module Libraries

- Effective management requires a big picture view of the portfolio. The concerns at this level are with the entire portfolio:
  - Enterprise Management Perspective
  - Application Portfolio Management
  - Program Management
  - Application Development and Computer Center Administration are supported with a comprehensive Configuration Management System

Enterprise Management Perspective

- Managers can plan the most effective way to migrate to a new technology or implement a set of standards
  - How many programs are there ?
  - What languages and technologies are employed ?
  - Are appropriate quality standards being met ?
  - How fast is migration to a new technology taking place ?

Application Portfolio Management

- Portfolio management looks at a single application or possibly a family of programs
  - What programs will be affected by a change to a subroutine ?
  - What programs are dependent on specific vendor run-time routine ?
  - Which programs are "easy to migrate" and which contain specific migration inhibitors ?

Program Management

- Developers migrating or modifying particular programs also need portfolio information
  - What components make up a particular program ?
  - What level of the compiler was used for each component ?
  - What compiler options were specified when the program was last compiled ?
  - Exist serveral compiled CSECTs from a program source with different compile dates and attributes ?
    These CSECTs might have different functionality !

Program Management

- Which compiler run-time routines and run-time option overrides are included as part of the application load module ?
- What is the release level of the runtime routines used or link edited into the application?
  Do the routines need to be refreshed to the latest level to migrate the application?
- Does the COBOL module use ACCEPT FROM DATE, or the PL/I module use DATE or DATETIME Builtin Functions to request the system date?

Program Management

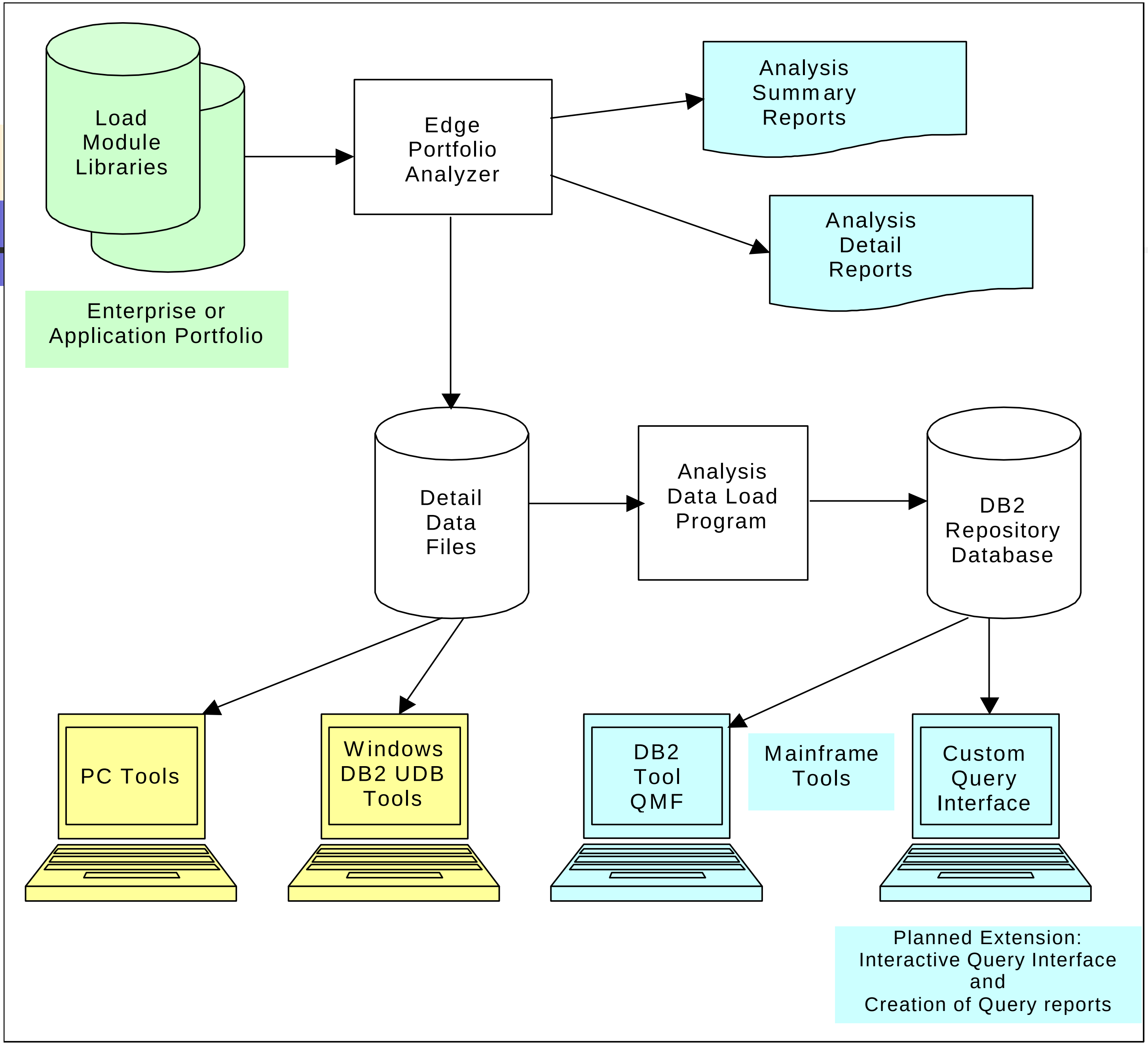
- Similar questions might be asked about the use of SORT, DB2, IMS, CICS or other LE or system functions if those components or subsystems are being upgraded
- Does the application make use of some Language Environment Callable Service that has changed in the latest release of LE ? This happened in OS/390 V2.9 and V2.10
- Does a COBOL program use dynamic CALLs to load an independently linked module for execution ?

Program Management

- Does an Assembler routine issue a LINK or LOAD macro to dynamically invoke a separately link edited module ?
- Do Assembler routines use other SVCs that may require attention when this application is modified ?
- Are there some components in a load module that require updating or relinking to allow the program to run with LE or to run above the 16MB line ?
- and many more attributes . . .

Conclusion of Analysis

- "The truth only exists in the code that goes into production every night"
- Having the right information readily available makes the developer more productive. Problem areas can be addressed immediately, avoiding a lengthy trial-and-error process



- Load Module Repository LMR built on Analysis Data

Load Module Repository provides

- Analyzer Summary Reporting
  The summary information of an entire load library is an inventory of all modules showing totals of modules that fit certain characteristics

  The summary information is vital for gaining an overall understanding of your enterprise and application portfolio and for quickly highlighting troublemakers

Load Module Repository provides

- Analyzer Detail Reporting
  The detail report describes the characteristics of the load modules and the CSECTs within them

  The detailed information is crucial for many operational activities such as the successful migration to a new compiler

Load Module Repository provides

- Analyzer derivative Reporting
  - Global Cross Reference Information
  - Performance Opportunity Identification
  - LE Migration Sensitivity
  - COBOL Migration Planning

The Load Module Repository helps you

- Locate load modules over all load libraries that will be impacted by changing a specified CSECT or program
- Improve performance of certain load modules simply by recompiling them with different options
- Identify load modules that are more likely to be easy to successfully migrate to a new compiler
- Identify load modules that will create difficulties during the migration to a new compiler and/or runtime environment
- Identify load modules containing Assembler CSECTs that use certain system functions via an SVC or service CALL that may have changed in a new release

The Load Module Repository helps you

- Determine the language used by all programs that use a particular facility of the system
  (e.g., What languages are used to access DB2 facilities?)
- Ensure that production programs conform to the installation standards for the compiler and linkage editor options (e.g., Inhibit "Test" compile option or linked in "Debugger" modules)
- Cleaning the load libraries from old and no more longer used programs and modules
- LMR is available on z/OS Mainframe and on Windows PC and Server (On UNIX on request)

Migration to current LE/390 version

- Standardization to only one runtime environment and version for Cobol, Fortran, PL/I and C/C++
- A prerequisite for migration to a current compiler version which all need LE/390 runtime
  (All old not LE-compliant compilers are now out of maintenance)
- Reduces number of runtime libraries in LINKLIST with performance profit
- Reduces complexity and sources of error for application development and operation

Migration to current LE/390 version

- Migration without Compile and Link but with Relink of all affected load modules with new LE/390 runtime modules (over all load libraries)
- Current functionality of load modules is guaranteed because no application program/CSECT is changed or recompiled,
  - no testing,
  - no integration,
  - no program release with administration

  is necessary

Migration to current LE/390 version

- Analysis data is used to find load modules with language specific or old LE/390 runtime modules to replace and creates REPLACE statements for Relink with Linkage Editor or Binder
- Older LE-compliant compilers can be used furthermore, e. g. VS Cobol II, PL/I V2R3
- Central migration of all operational load libraries to LE/390 needs only a weekend (with IPL for corrected LINKLIST)

Migration to current LE/390 version

- LE/390 offers additional test features to enhance program quality
  - Clearing stack and heap storage to special byte patterns like X‘FE‘ before a program or subprogram is called to find not initialized variables and avoiding usage of values by chance
  - Reports of allocated stack and heap storage to avoid storage extents with performance loss

Migration to LE-compliant Compiler

- Flexible compiler migration includes
  - In a load module/program coexistence of earlier compiled modules with new compiled modules caused by program maintenance
  - Compile, link and runtime options for coexistence and compatibility,
    e.g. format of parameter list
  - Prototyping for coexistence and compatibility

Special Feature: Automated Relink

- Force replacement of IBM supplied run-time subroutines. This might be used in a COBOL application to support newly generated code in NORES or MIXRES environments or to upgrade a PL/I application to use the most recent level of the Resident Library
- Update the run-time interfaces for subsystems such as CICS, IMS and/or DB2 to the latest level
- Propagate new versions of user written subroutines to all applications that use them. Common user routines or ABEND routines might be updated this way

Please don‘t hesitate to contact

Eberhard Ramm
Industriestraße 35
82194 Gröbenzell

Internet:	www.SibraGmbH.com
email:	SibraGmbh@aol.com SibraGmbh@t-online.de
Telefon:	+49 (0)8142 57264
Telefax:	+49 (0)8142 57265
Mobil:	+49 (0)172 8409119