Generating Automatic Documentation for Impact Analysis

Maintaining System documentation is a major undertaking
Almost always...

- Documentation *lags behind* development

- **Time is required** to transfer knowledge for documentation

- **Time is required** to produce the *bwcharts* and diagrams

- **Changes are difficult** to incorporate timely

---

Result

- Documentation is a highly involved and costly process

- Documentation is not expected to be always *up-to-date!!*
Contents of program documentation

- Data flow diagrams
- System flowcharts
- What does the program do?
- Who made the changes?
- What were the last changes?

Contents of database documentation

- Database structure
- Table structures
- Index columns
- Referential integrity
- What were the last changes?
Documentation can include:

- Aid in tracking failures
- Which tables are related to this program?
- Which programs are accessing this table?
- What would be the impact of changes?
- On demand Error→Program→TABLE→

Documentation can address:

- Auto-notification to a developer with the necessary documentation
- Generated and distributed immediately
- Statistics on failures:
  - How many times a program has failed?
  - How many times the program has failed for the same reason?
- Resource usage for TIMEOUT and DEADLOCK analysis
- Identify unused resources
- Help database administration
Documentation can address:

- Not only the conventional changes to the document
- But, a chronological log of actions taken for aid in trouble-shooting
- Creation of a knowledge bank for the organization:

  Help Production Support
  Automatically
  Real-time

Documentation can automatically include:

- Track resource in perpetual timeouts
- Track objects involved in perpetual deadlocks
- Identify unused resources
- Help database administration
Documentation can be searched

Search: All programs, database objects, error messages, log-files, reference-date, comment-lines...

...on a key word or phrase

- Which programs contain the literal “warrants”?
- List all table names with “warrant” literal
- List all column names and metadata
- What queries contain the literal “warrant”? 

Complete documentation

- Design documents
- Changes at all stages of development
- Impact on changes
- Run-time errors
- Resource utilization
- Statistics on failures
- Up-to-date diagrams
- Search facility
- On-demand generation and distribution
A new approach for System Documentation

The Approach

Much information is already available from an operational system
1

System documentation can be generated automatically from the system itself

2

System documentation can automatically be linked with daily production runs for trouble shooting
3

Flowcharts, diagrams, statistics and graphs can be generated automatically

4

Daily resource usage can be tracked and documented automatically
5

Upon failures, developers get up-to-date documentation generated real-time

Impact Analysis

It is the challenge to cross-reference between the program’s documentation and the database’s documentation
Impact Analysis

- Use a query to *automatically* generate the following:
  - From an application database table's perspective, a list of all programs that use the table
  - From a program's perspective, a list of all of the database tables used in the programs
  - Search based on application and/or any reference key words across programs and the all of attributes in the database tables

How it works

PHASE-1

- Collect data from db system objects
- Collect data from daily OS and SQL logs
- Load document database tables
- Parse programs in several cycles for database information
- Retrieve information related to documentation
How it works

PHASE-2
- Generate links between program information and database information
- Create documentation data from links
- Generate diagrams through Visio graphic engine
- Generate web pages report with embedded diagram

PHASE-3
Periodic updates of document database
Impact Analysis

IMPACT ANALYSIS FLOW CHART

Program repository

Source Code

PARSE TO FIND SQL STATEMENTS

PARSE SQL STATEMENTS

VERIFY TABLE-NAMES,COLUMNS

SQL System Tables

Tables info

Column info

LINK

Load USAGE TABLES

VISIO API

Hands on demonstration
System Flow Chart

Data Flow Diagrams

Search BO text

Search column names

Search table names

Export result
Benefits
Benefits

- Since the application is automated, the tedium of manual maintenance is eliminated

- Documentation can be generated as often as required

- Maintenance time is dramatically reduced

- Found very useful by the developers, production support and even auditors

Benefits

1. Organization

Any large IT system can be instantaneously documented with flowcharts and data flow diagrams

This is impossible to do manually for each and every change in the system
Benefits

2 Developers

- Use as an impact analysis tool
- Use as an interactive design helper
- Use for system-wide searches for impact analysis

Benefits

3 Database Administration

- Tracking resource usage for DBAs
- Link programs with timeouts and deadlocks
4 Production Support

- Link to error log (Production run errors)
- Rapid troubleshooting
- Track change requests

5 For Auditors

- Easier trace from process-to-data store
- Reconciliation simplified
Conclusion

- A running system has much information which can be programmatically organized
- System documentation need not be static, it can automatically include all changes, build statistics
- System documentation can include daily production results to help production support
- Troubleshooting experience can be included automatically to build up a knowledge bank over time
- All complex drawings, flow-charts can be generated real-time
- A new concept of system documentation as a generic tool for any large or small system

Thank You

Louisiana Department of Revenue

Tapan Sarkar