Specifik Software

System Understanding

Specifik Software A/S
info@specifik.dk

System Understanding

Mainframe systems are often designed and programmed over a period of 40 years. The total invested resources can be a 4-5 digit number of man years, involving a 3-4 digit number of persons (internal and external, available and not available). This must necessarily mean complex systems and lack of up to date application knowledge.
The complexity is increased due to the more or less consequent use of different concepts, programming languages, data base systems, user interface techniques, transactions monitors etc.
The many company mergers and the growing use of outsourcing further increases the complexity.
And WWW adds extra layers.

Many maintenance tasks involves the major part of a systems programs and other components. Y2K and Euro are well known examples. The prerequisite for solving these tasks economically and timely optimal is, to have a complete knowledge about the system, both on a general level and on a detailed level. Therefore dedicated tools are used for this purpose.

Many other tasks could benefit from a corresponding analysis.

Outsourcing (documentation, consistency, quality).
Unbundling (interface analysis, documentation).
Subsystem replacement (do).
Domain analysis (Basel II, IAS 39, Credit Card Systems, bank id, account numbers, county numbers, etc..
System technique analysis (CICS, SQL, DLI, etc.).
Optimizing (dead code, data types).
Restructuring (flow, slicing).
Quality (metrics, pretty print).
Software migration (documentation, diverse analysis).
Platform migration (do).
Generations shift (knowledge/experience transfer)

This need for analysis, in spite of the task's different objectives, is a permanent state. And the solution for this, naturally must be a permanent tool, which immediately can document a system, and with no or little adjustment can be used for special tasks.

Re2Spec/Product

Re2Spec is an integrated set of tools for documentation, general analysis, domain analysis and ad hoc analysis purposes for the above mentioned systems and tasks. Re2Spec is a new designed and new programmed tool to replace Specifik/Filter, which since1980 has been used for a 3 digit number of analysis and conversion projects (all of them quite different).
The last projects were Y2K verification (IV&V) of 230.000.000 statements for a number of Scandinavians biggest mainframe installations (ex. KMD, KD DATA, Volvo, Schenker-BLT, BGC, Foreningssparbanken, Spintab, Svenska Handelsbanken, SKF, Electrolux, Wasa, Alm. Brand og SCA Molnlycke).

The development of Re2Spec started 1/1/2000.
Re2Spec has been used by a customer for a number of projects comprising 80.000 COBOL and PL1 programs, and by another customer for Euro analysis of 20.000 COBOL programs.
This customer has furthermore used Re2Spec to convert 10.000 MetaCobol programs and for analyzing 3000 Cool:Gen programs.

Re2Spec can, as the best corresponding tools on the market, perform very detailed analysis, because the tools 'knows' the individually components (programming language etc.) syntax and semantic.

3 principles distinguish Re2Spec from other tools.

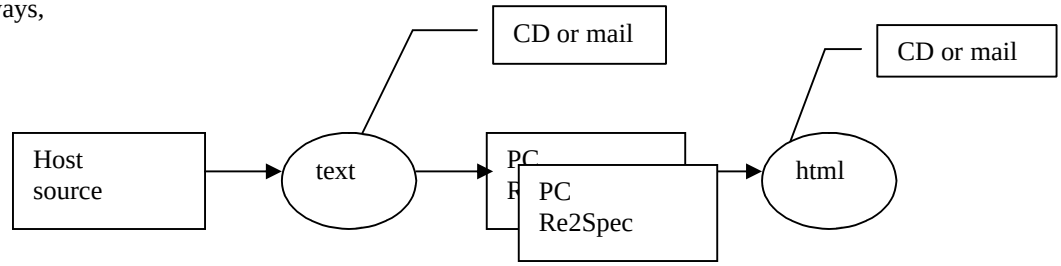
1) Re2Spec is not designed for a particular programming language, the tools basic components, compiler, simulator and transformer, are generic. New programming languages and analysis can be defined as plugins as needed.

2) Robustness. A system can refer to missing and/or illegal components. In a big system the number can be several thousands objects. Therefore Re2Spec is designed not to stop in situations like this. There will always be generated the documentation, which in any way is possible to establish from the more or less complete system.

3) Performance. Analysis of a system comprising 200.000 components (COBOL, PL1, CICS, SQL, JCL, DFH…) and generation of documentation consisting of 1.100.000 html documents lasts 20 hours.
Often one needs a number of iterations before the first useful documentation is available, due to lack of knowledge about which programming languages has been used, which libraries contains what etc. Therefore a part of Re2Spec is a Sniffer, which switch to the right compiler and reallocates objects if necessary. This means one literally can download all the code into a single library, start the analysis, and out comes a useful result after 1 (one) iteration.

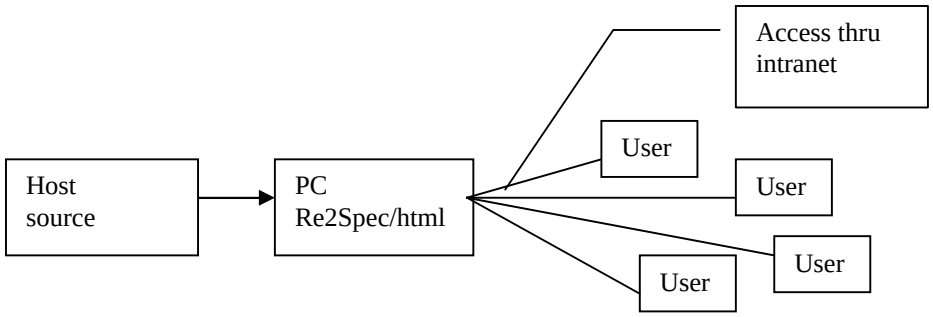
Re2Spec can be used in 3 ways,

1 – external/service:



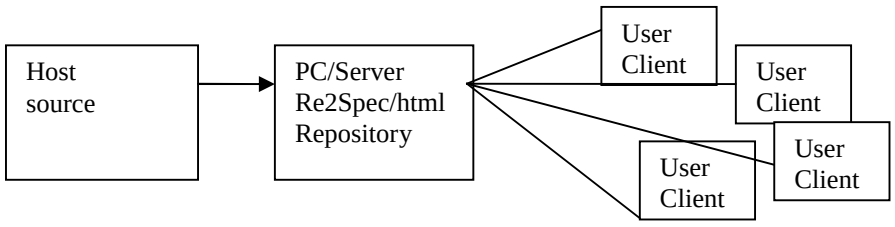
The system is transferred to a medium (CD or mail), posted to Specifik Software, who analysis the code, and generates the documentation in form of Html. This is returned, and can then be installed as wished. The Html documents can be read by any standard browser. No other software is necessary.

2 – internal/simple:



Re2Spec is installed on a PC. The system is transferred from the Host to the PC as needed. The analysis is performed as needed. If, after each analysis, the generated documentation is saved on a CD or another backup, a documentation history archive is established!.

3 – internal/advanced:



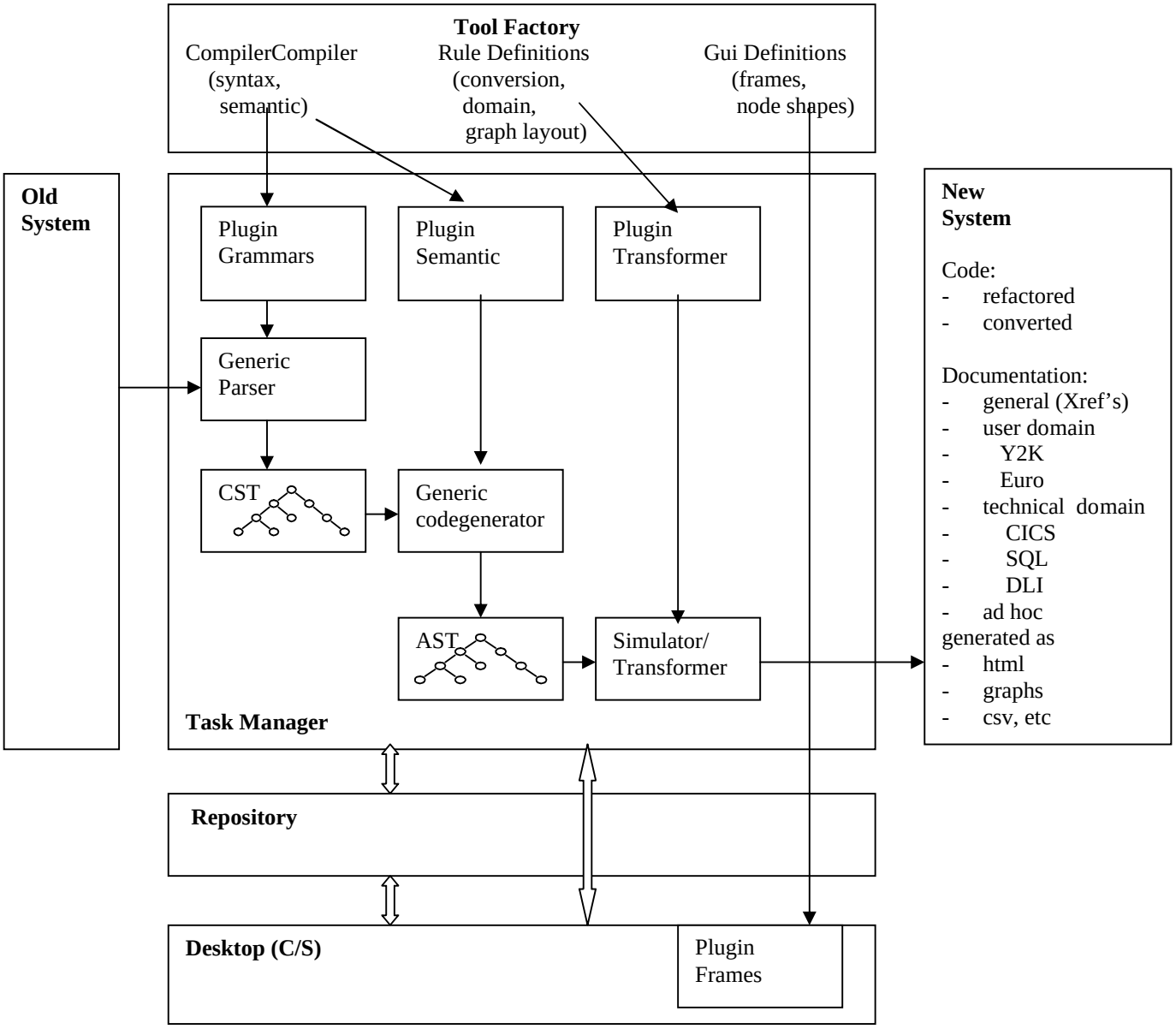
This option offers a number of facilities besides the Html documentation. The administrator and the users can via a Gui/desktop perform ad hoc queries (sql format), define jobs, projects, models, scopes etc. The administrator can define users and access rights.

Re2Spec/Design

Re2Spec consists of 4 main components,
Tool Factory, Task Manager, Repository and C/S Desktop

Re2Spec can be excecuted as a pure Client or in a Client/Server environment.

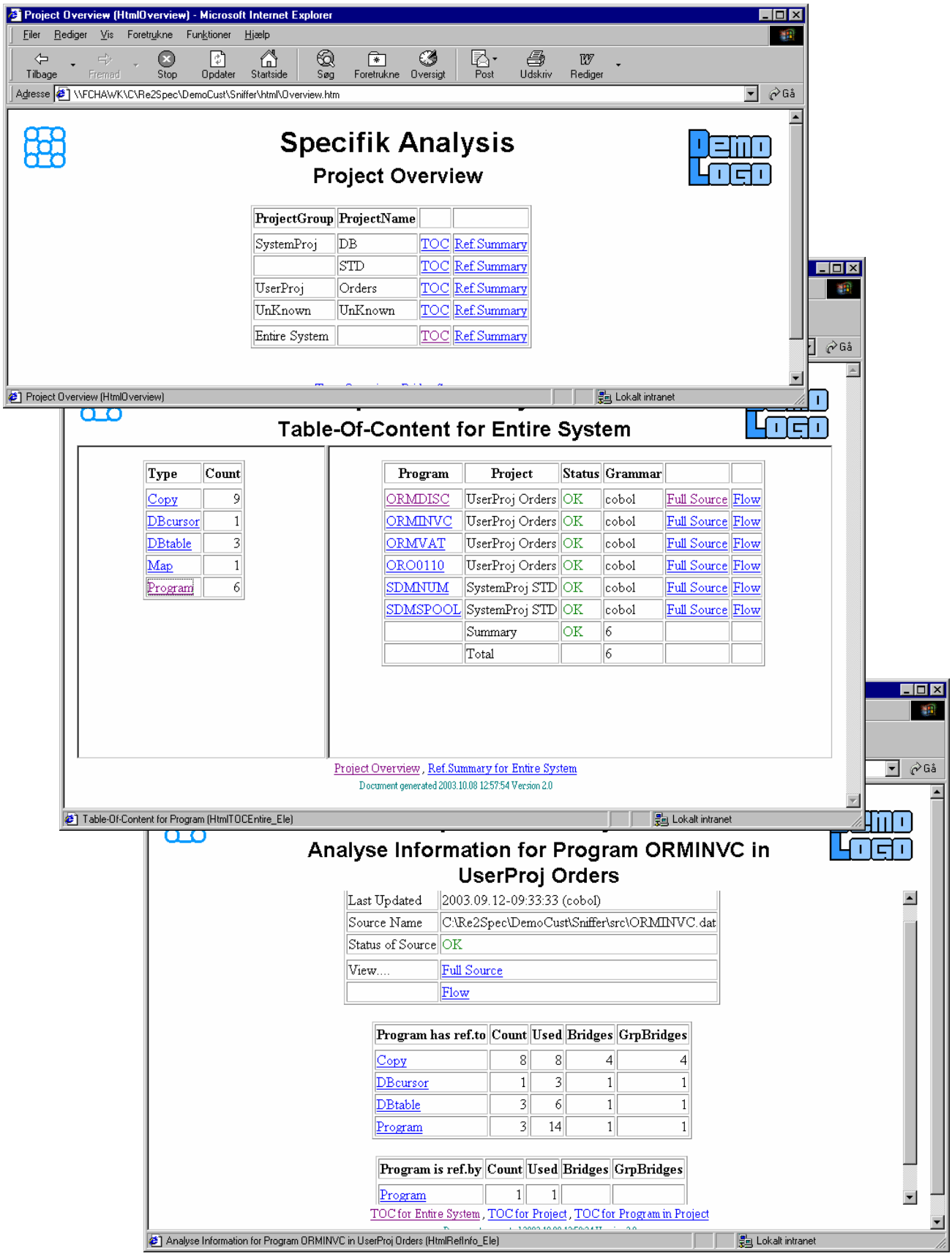
Re2Spec can be executed with or without the Repository activated.



The following languages/components are implemented. COBOL, C, Java, PL1, CICS, DLI, SQL, JCL, DFH….
New languages/components can be implemented as plugins as needed.
Languages/components, which are not defined as plugins, is attempted to be recognized (ASM, C, Easytrieve, etc.), so these can be catalogued and integrated in the total documentation.

Just a few examples of the approximately 20 different “common” types of HTML generated by Re2Spec.

Most of these common types occurs in different “flavours” depending on the number of projects specified, component-types found in the analysis (might be >20) and of course the number of components analysed causing the total number of “flavours” to be several hundreds and the total number of documents to be as many as millions.



Project Overview (HtmlOverview) - Microsoft Internet Explorer

Specifik Analysis

Project Overview

ProjectGroup	ProjectName		
SystemProj	DB	TOC	Ref.Summary
	STD	TOC	Ref.Summary
UserProj	Orders	TOC	Ref.Summary
UnKnown	UnKnown	TOC	Ref.Summary
Entire System		TOC	Ref.Summary

Table-Of-Content for Entire System

Type	Count
Copy	9
DBcursor	1
DBtable	3
Map	1
Program	6

Program	Project	Status	Grammar		
ORMDISC	UserProj Orders	OK	cobol	Full Source	Flow
ORMINVC	UserProj Orders	OK	cobol	Full Source	Flow
ORMVAT	UserProj Orders	OK	cobol	Full Source	Flow
ORO0110	UserProj Orders	OK	cobol	Full Source	Flow
SDMNUM	SystemProj STD	OK	cobol	Full Source	Flow
SDMSPOOL	SystemProj STD	OK	cobol	Full Source	Flow
	Summary	OK	6		
	Total		6		

Project Overview , Ref.Summary for Entire System

Document generated 2003.10.08 12:57:54 Version 2.0

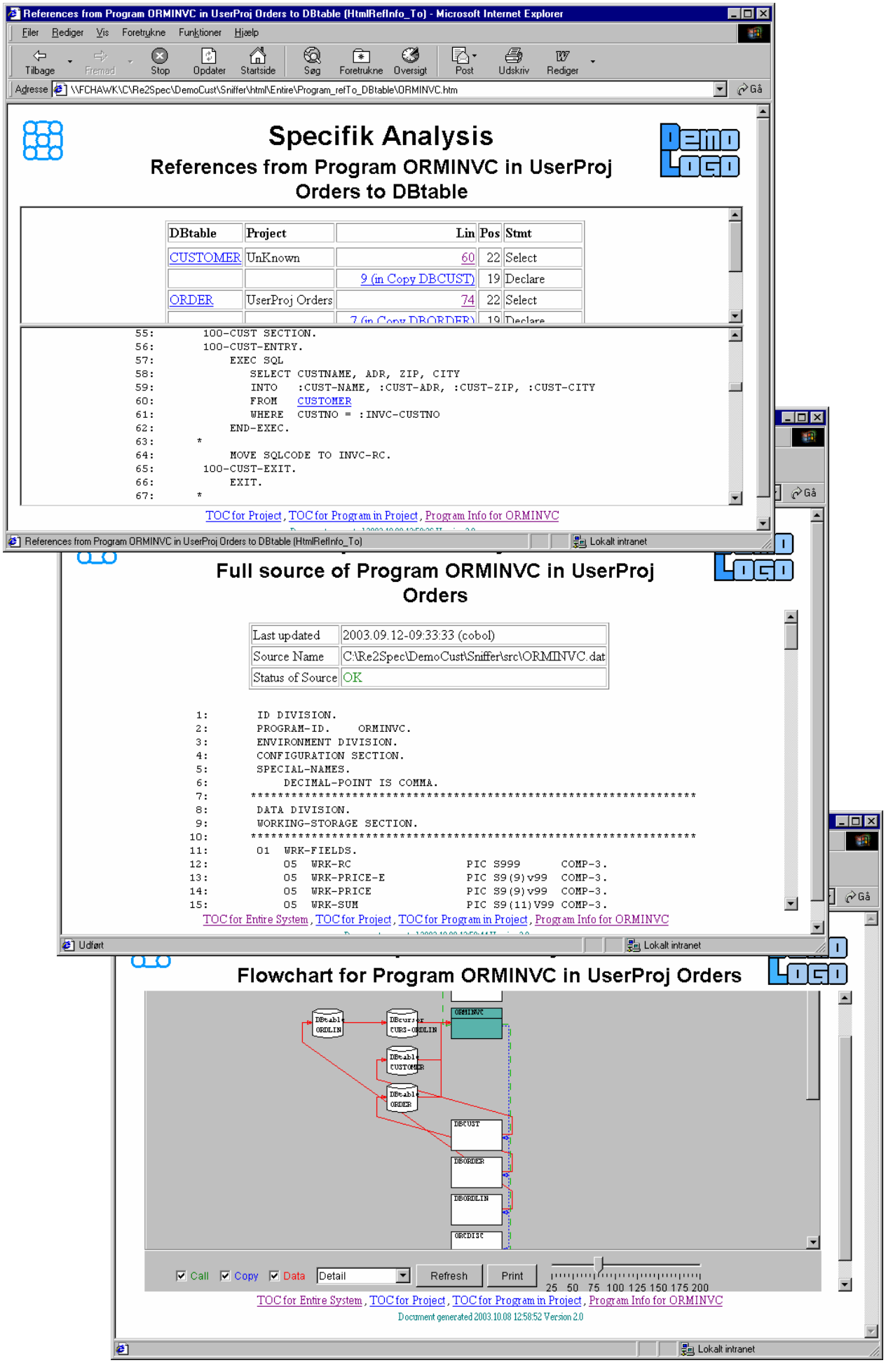
Analyse Information for Program ORMINVC in UserProj Orders

Last Updated	2003.09.12-09:33:33 (cobol)
Source Name	C:\Re2Spec\DemoCust\Sniffer\src\ORMINVC.dat
Status of Source	OK
View....	Full Source
	Flow

Program has ref.to	Count	Used	Bridges	GrpBridges
Copy	8	8	4	4
DBcursor	1	3	1	1
DBtable	3	6	1	1
Program	3	14	1	1

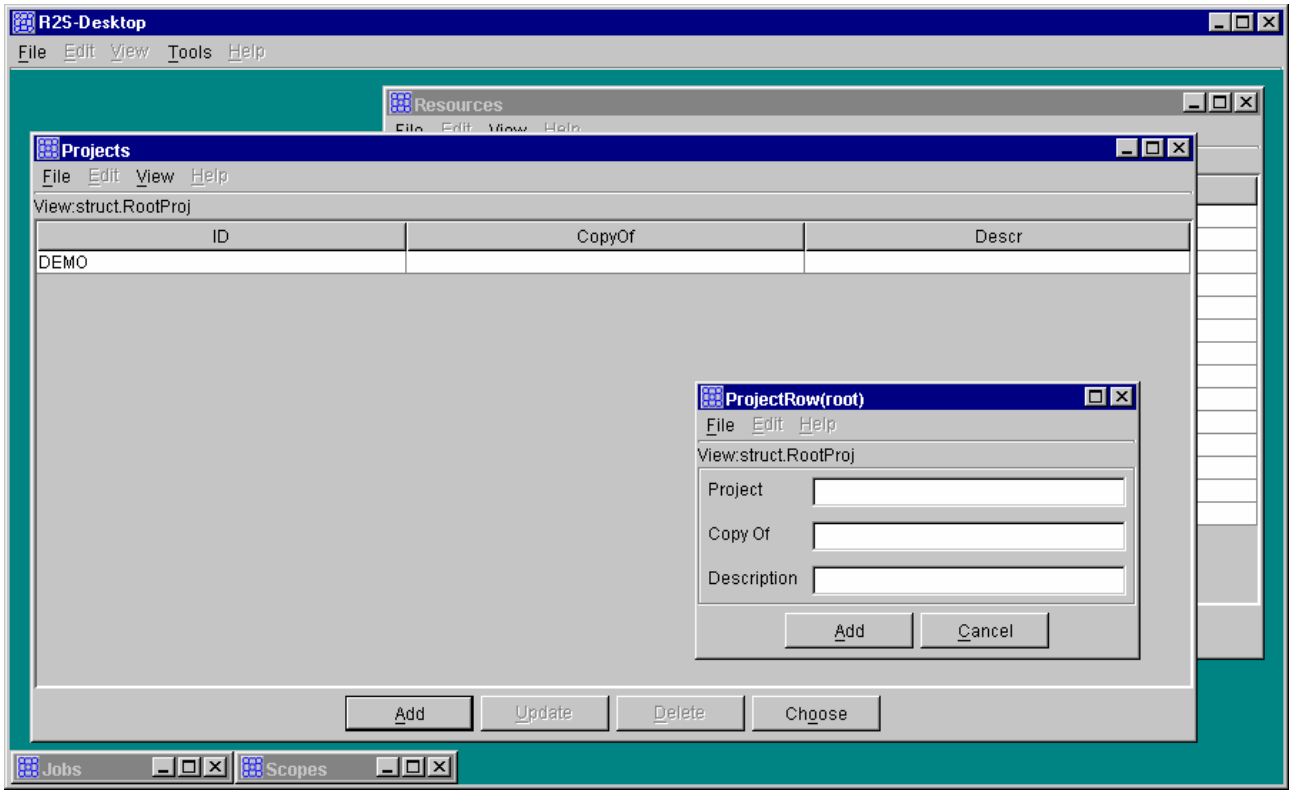
Program is ref.by	Count	Used	Bridges	GrpBridges
Program	1	1		

TOC for Entire System , TOC for Project , TOC for Program in Project

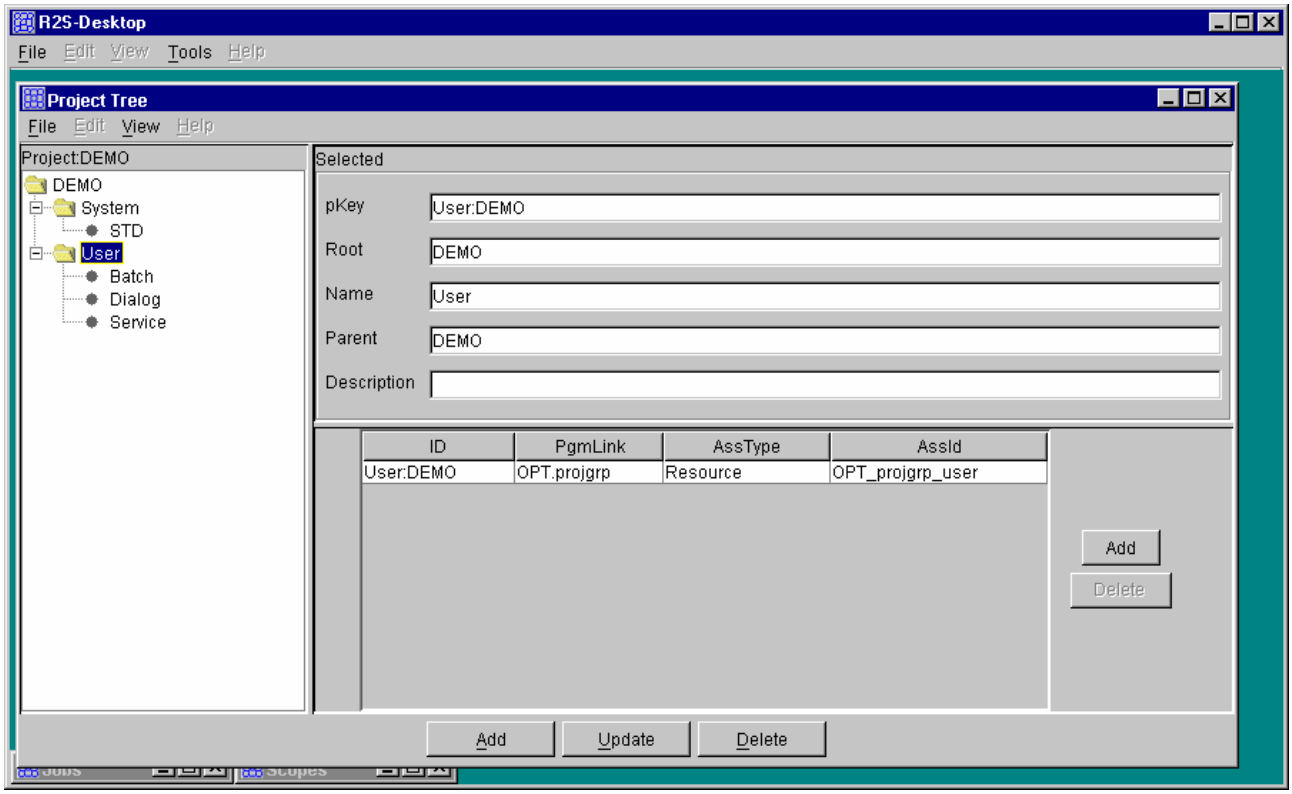

References from Program ORMINVC in UserProj Orders to DBtable (HtmlRefInfo_To) - Microsoft Internet Explorer
Specifik Analysis
References from Program ORMINVC in UserProj Orders to DBtable
DBtable
Project
Lin
Pos
Stmt
CUSTOMER
UnKnown
60
22
Select
9 (in Copy DBCUST)
19
Declare
ORDER
UserProj Orders
74
22
Select
55: 100-CUST SECTION.
56: 100-CUST-ENTRY.
57: EXEC SQL
58: SELECT CUSTNAME, ADR, ZIP, CITY
59: INTO :CUST-NAME, :CUST-ADR, :CUST-ZIP, :CUST-CITY
60: FROM CUSTOMER
61: WHERE CUSTNO = :INVC-CUSTNO
62: END-EXEC.
63: *
64: MOVE SQLCODE TO INVC-RC.
65: 100-CUST-EXIT.
66: EXIT.
67: *
TOC for Project , TOC for Program in Project , Program Info for ORMINVC
Full source of Program ORMINVC in UserProj Orders
Last updated
2003.09.12-09:33:33 (cobol)
Source Name
C:\Re2Spec\DemoCust\Sniffer\src\ORMINVC.dat
Status of Source
OK
1: ID DIVISION.
2: PROGRAM-ID. ORMINVC.
3: ENVIRONMENT DIVISION.
4: CONFIGURATION SECTION.
5: SPECIAL-NAMES.
6: DECIMAL-POINT IS COMMA.
7: *****
8: DATA DIVISION.
9: WORKING-STORAGE SECTION.
10: *****
11: 01 WRK-FIELDS.
12: 05 WRK-RC PIC S999 COMP-3.
13: 05 WRK-PRICE-E PIC S9(9)v99 COMP-3.
14: 05 WRK-PRICE PIC S9(9)v99 COMP-3.
15: 05 WRK-SUM PIC S9(11)V99 COMP-3.
TOC for Entire System , TOC for Project , TOC for Program in Project , Program Info for ORMINVC
Flowchart for Program ORMINVC in UserProj Orders
ORMINVC
DBCUST
DBORDER
DBORDLIN
ORDISC
Call
Copy
Data
Detail
Refresh
Print
25 50 75 100 125 150 175 200
TOC for Entire System , TOC for Project , TOC for Program in Project , Program Info for ORMINVC
Document generated 2003.10.08 12:58:52 Version 2.0
Lokalt intranet

Some examples from the GUI:

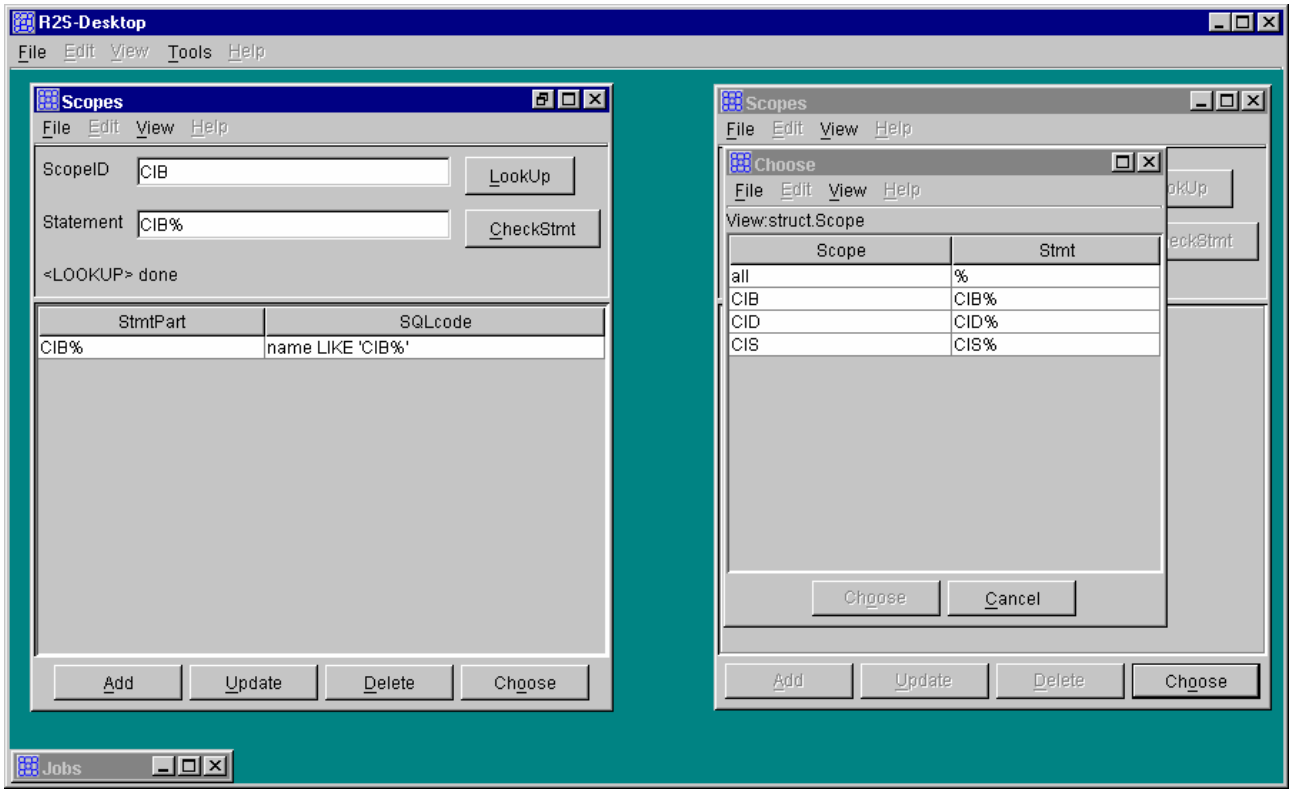
Defining a Project.



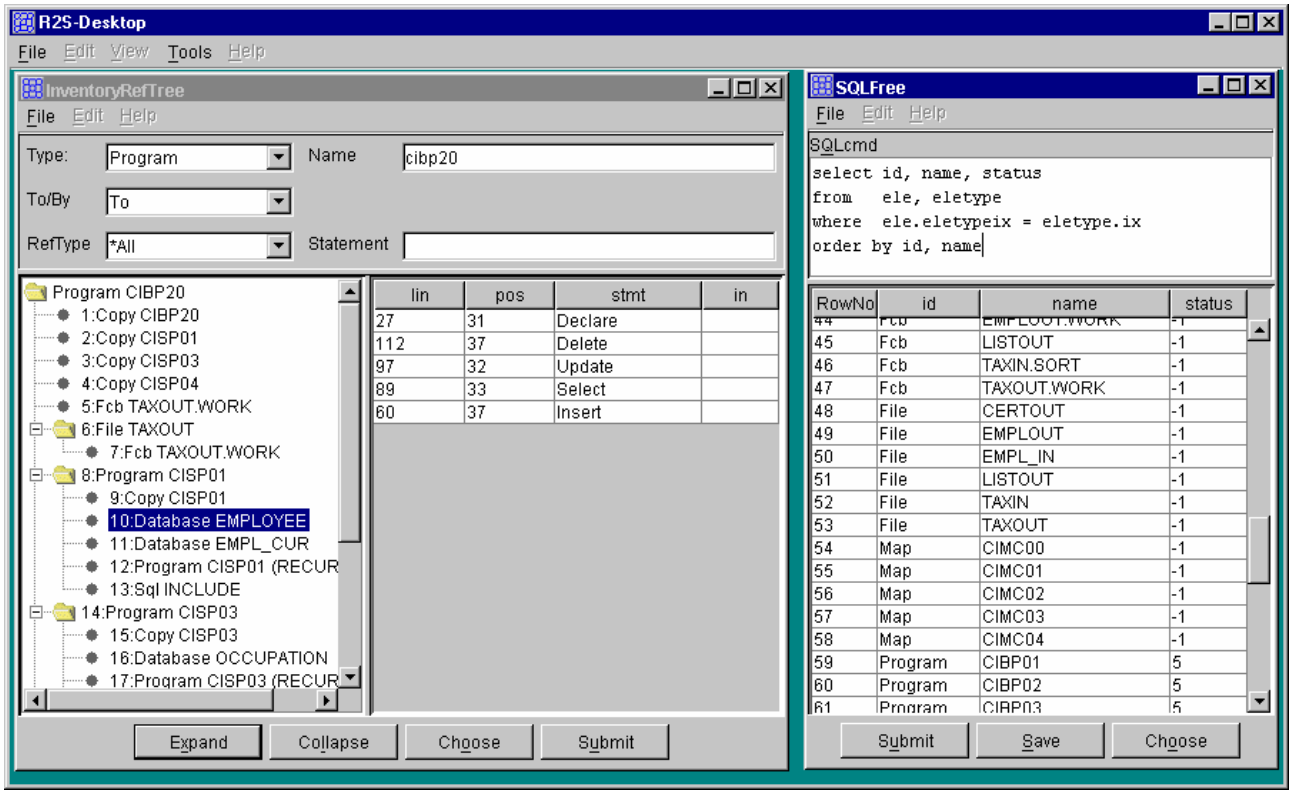
Defining the project structure and the usage of resources.

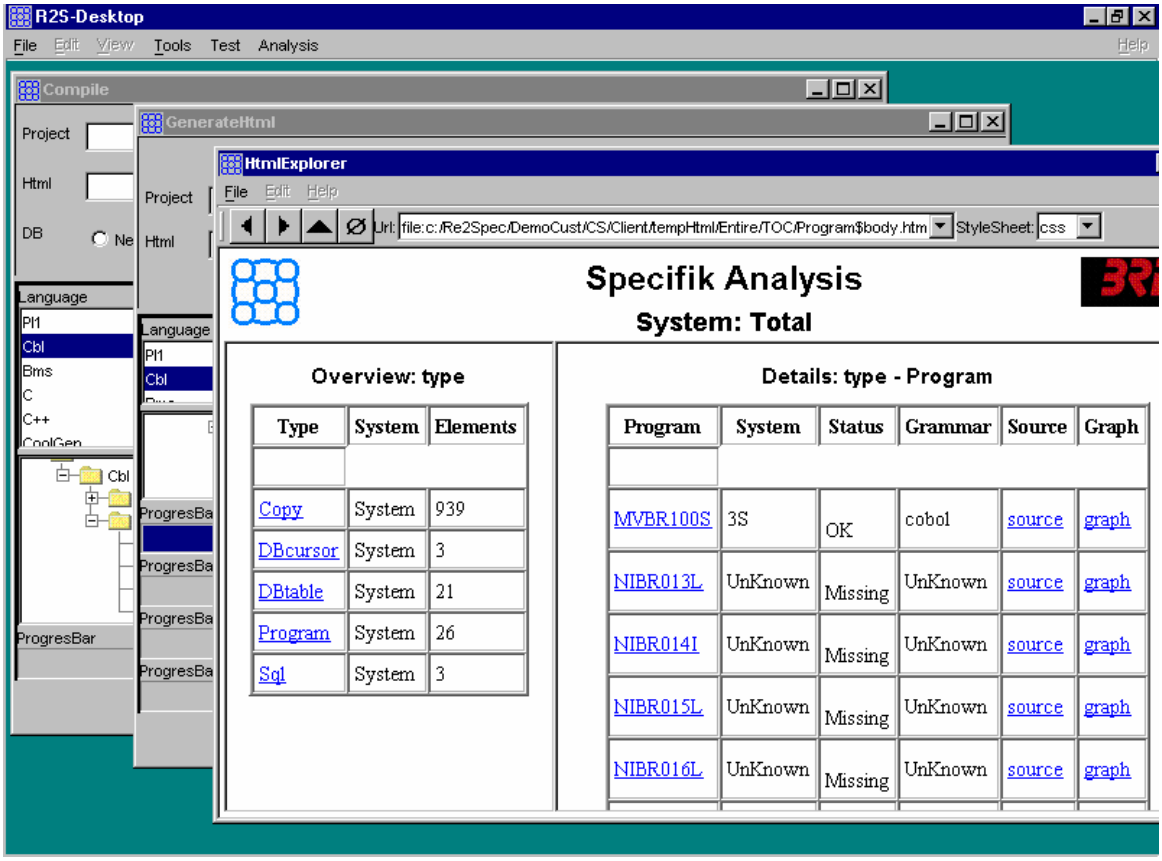


Defining the namespaces (“scopes”) for the project structure.

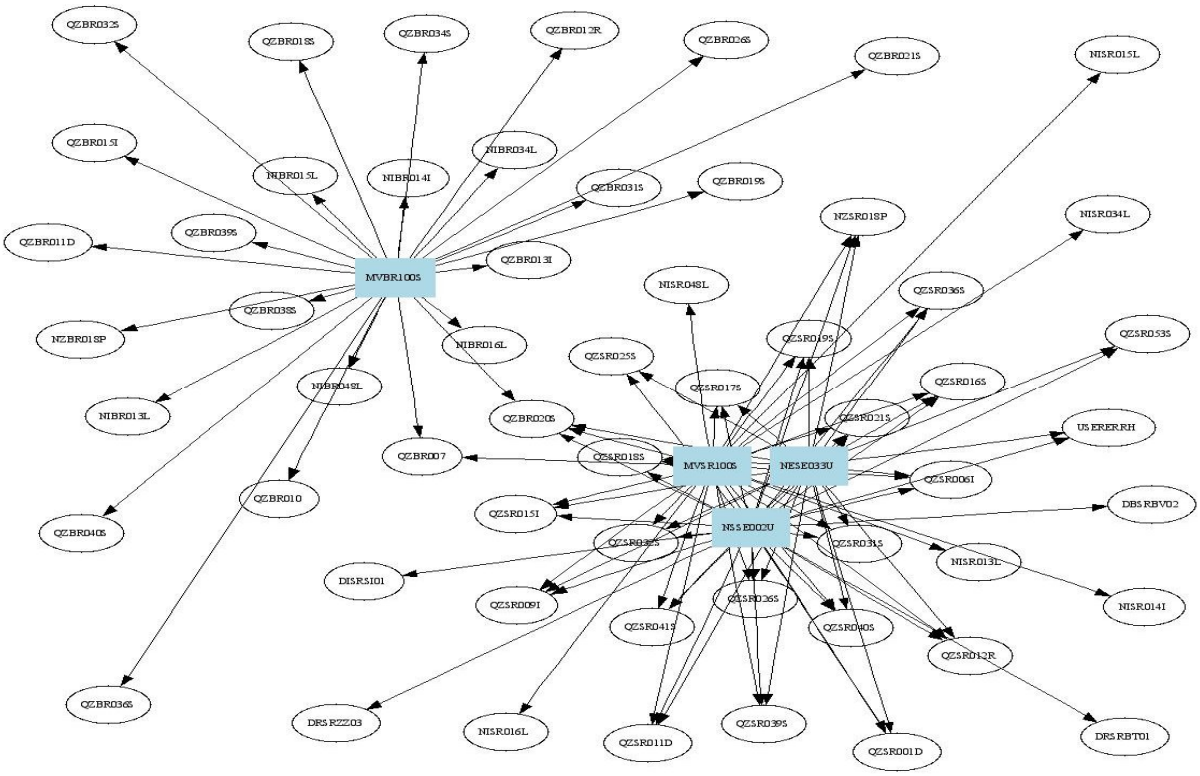


AdHoc-requests using reference tree and “free form” SQL.





Bulitin HtmlExplorer:



Graphs: