Traceability – An Introduction

Eine Ringvorlesung
für die Technische Universität Dresden
by

Harry M. Sneed
SoRing Kft. H-1221 Budapest

Institut für Software Technik
Dresden im November 2018

Two Forms of Software Tracing

There are two forms of software tracing:

1. **Dynamic tracing**, when you execute the software and follow the path of execution through the code.
2. **Static tracing**, when you begin with a software artifact and collect all the other artifacts related to that one, for instance, you begin with a change request or an error message and collect all artifacts, i.e. documents, code members and test specifications, related to it.

Origin of Dynamic Software Tracing

- Dynamic Tracing began with the Question „what path thru the system is followed by this test case?“
- The answer that question one has to instrument the code by placing markers at different points in the code, for instance after each decision.
- Dynamic tracing is used to debug and to document tests.
- In testing modules for Siemens at the Budapest Test Laboratory in 1978 this speaker had to document every test case since payment was based on test cases tested.
- The monthly test report to Siemens included the path of every test case tested.
- Dynamic tracing is used today to document tests of all kinds from unit testing to network testing.

Origin of Static Software Tracing

- Static Tracing began with the Question „What will it cost to make this change?“
- The answer was „that it all depends on what is affected by that change.“
- To find that out one had to trace the change thru the Software artifacts – documents, code and tests.
- This was referred to as „Impact Analysis“.
- Research on Impact Analysis began in the early 1990s.
- Steve Bohner wrote his dissertation for the University of Maryland on the subject of “Impact Analysis in the Software Change Process ”, in the year 1990, the same year that Robert Arnold finished his dissertation at the same University on “Reverse Engineering”.
- The two subjects are highly related.

My Work on Static Tracing

- My work on static tracing began in a banking project in Vienna in the years 2000 to 2003.
- I was responsible for calculating the costs of changing and extending a complex C++ system.
- To do this I had to identify the impact domain of the change in order to measure the size.
- That resulted in a repository-based tool to automatically calculate the costs of implementing specified changes.
- This work was reported on at the IEEE International Conference on Software Maintenance (ICSM) in Florence, Italy in Nov. 2001.

Impact Analysis of Maintenance Tasks
for a Distributed Object-oriented System

Harry M. Sneed
Software Daten Service GmbH
Vienna, Austria

GEOS System Model Levels

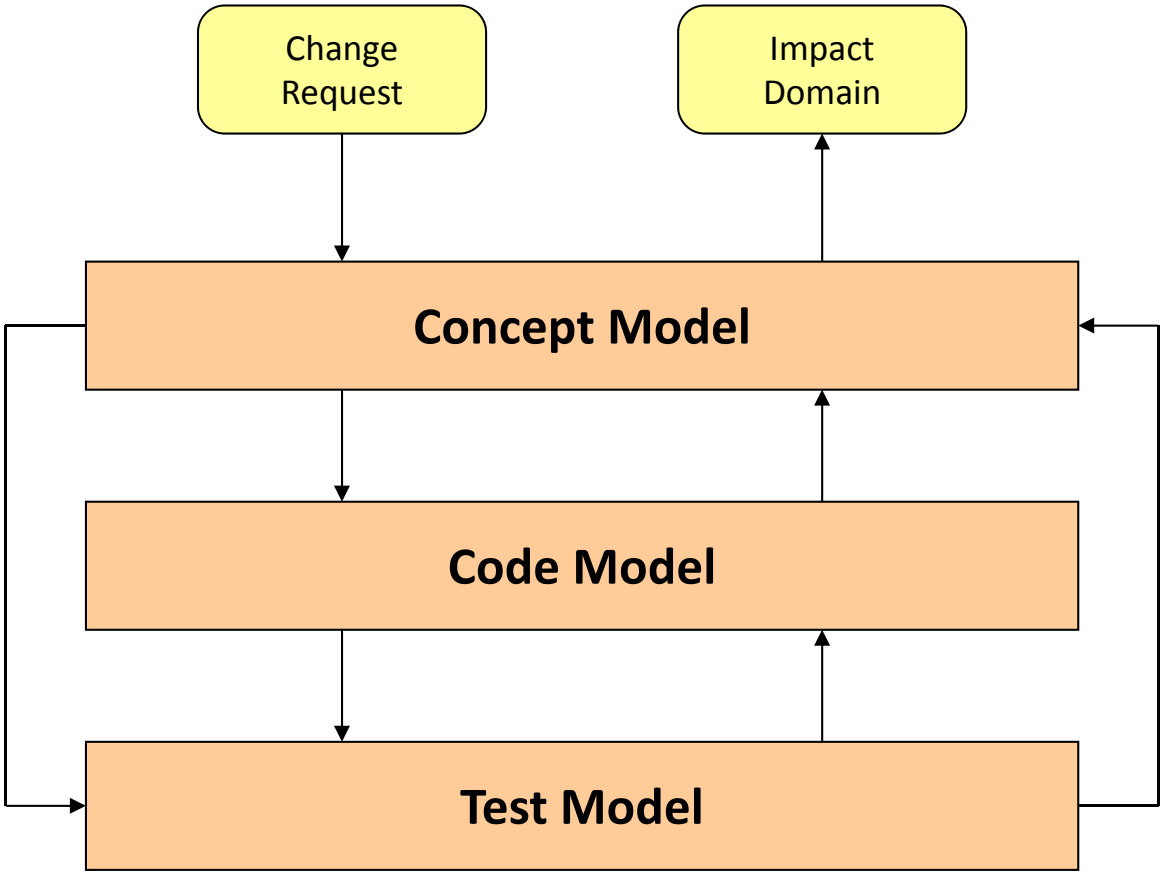

Figure 1: Repository Levels

Architecture Impact Analysis

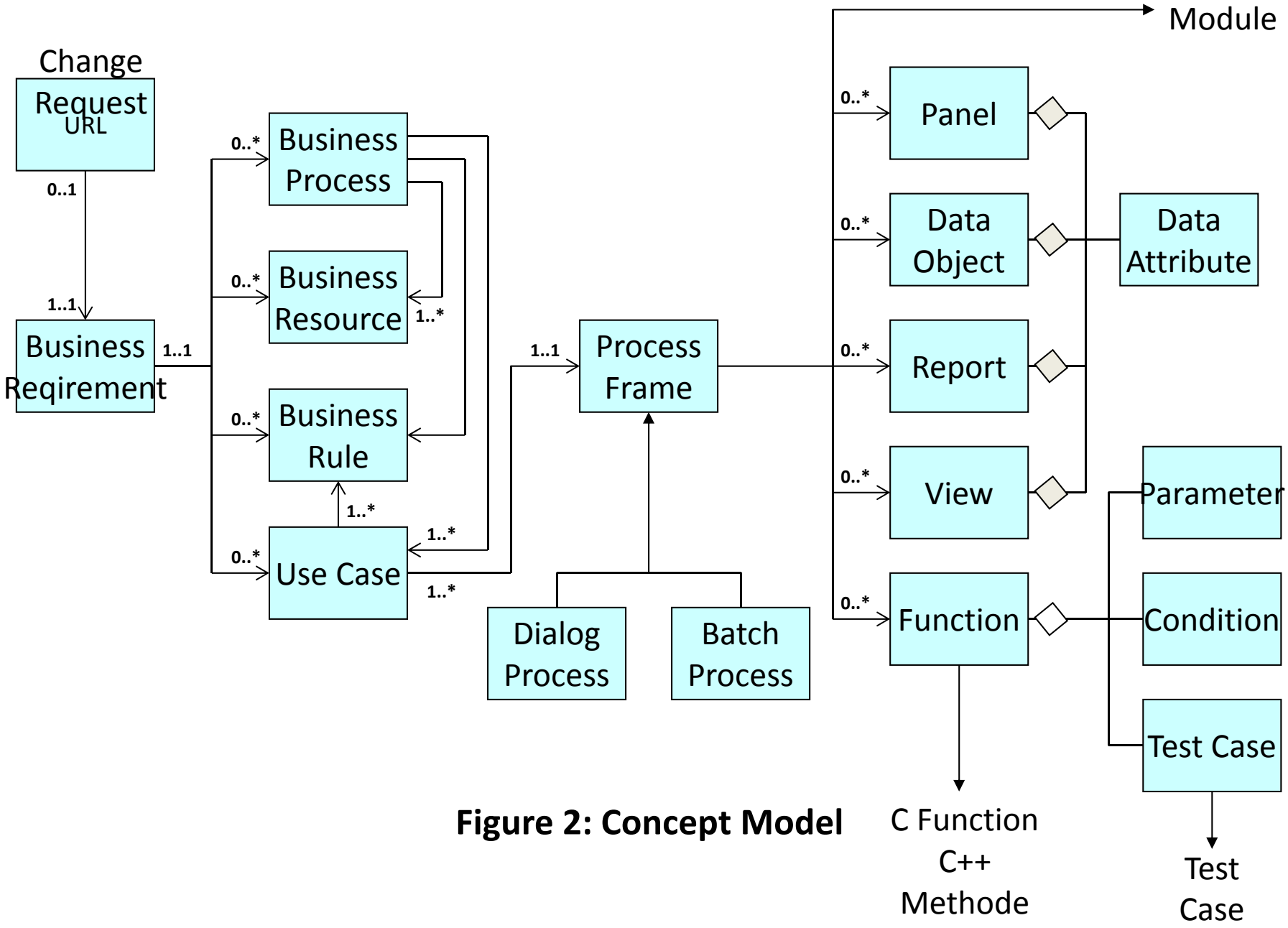

Figure 2: Concept Model

Code Impact Analysis

```
+-------------------------------------------------------------------+
|                                                                   |
|                      SDS Repository Impact Analysis               |
| Product     : GEOS                                                |
| System      : NOSTRO                                              |
| Project     : Sneed                                 Date : 20.10.2001 |
| Repository :  D:\tools\maintain\softrepo\tables                   |
+-------------------------------------------------------------------+
| Base Element : GAF                                                |
| Element type : COMP                                               |
| Search Direction : F                                              |
+-------------------------------------------------------------------+
| Lev Impacted Elements ( Forward   )     Element Type Module_Id    |
+-------------------------------------------------------------------+
|  1 GAF                                  COMP    -> 0000           |
|  1 gafausfgh                            MOD     -> 0016           |
|  2 GASVC_Steuer_Kontr                   CLASS   -> 0016           |
|  2 gafkdzutcpp                          MOD     -> 0017           |
|  3 gafausfg.h                           INC     -> 0021           |
|  2 KD_Ausfg_Warten                      FUNCT   -> 0021           |
|  2 GetKontoSparte                       FUNCT   -> 0021           |
|  3 Execute                              FUNCT   -> 0036           |
|  3 GAMAusfg_Kd_ZuteilungDlg             FUNCT   -> 0032           |
|  2 gafktzutcpp                          MOD     -> 0018           |
|  3 gafausfg.h                           INC     -> 0021           |
|  3 gafzutei.h                           INC     -> 0021           |
+-------------------------------------------------------------------+
/                Total Number of Elements impacted =      12
+-------------------------------------------------------------------+
| Base Element : GetSKONTR                                          |
| Element type : FUNCT                                              |
| Search Direction : F                                              |
+-------------------------------------------------------------------+
| Lev Impacted Elements ( Forward   )     Element Type Module_Id    |
+-------------------------------------------------------------------+
|  1 GetSKONTR                            FUNCT   -> 0017           |
+-------------------------------------------------------------------+
/                Total Number of Elements impacted =        1
+-------------------------------------------------------------------+
```

Artifact Impact Table

Requester:	Bank of Austria	System:	GEOS
Change Request:	Alter_Order_Map	Date:	2001.01.22

Level	Type	Artifact	Lines	Stmts	Fct. Pts	Obj. Pts.	Test Pts.	Comp	Qual	Impact Rate
Concept	Panel	Order_Map	39	39	7	21	-	0.420	0.602	10%
Concept	Dialog	Order_Entry	67	67	13	39	-	0.491	0.598	10%
Concept	Frame	Order_Process	208	208	20	112	-	0.523	0.570	5%
Concept	Funct	Status_check	92	92	6	28	-	0.318	0.635	20%
Concept	TestCase	TF_OE_011	9	9	-	-	4	-	-	50%
Concept	TestCase	TF_OE_013	7	7	-	-	3	-	-	50%
Code	Comp	Procord	2420	1680	51	803	-	0.561	0.501	2%
Code	Module	Procord1	860	560	19	240	-	0.573	0.499	5%
Code	Class	Order	97	42	-	28	-	0.519	0.538	10%
Code	Method	Check_Status	36	21	-	-	-	-	-	40%
Test	TestCase	TF_OE_011	12	8	-	-	4	-	-	50%
Test	TestCase	TF_OE_013	10	6	-	-	3	-	-	50%
Test	TestObj	Procord	219	150	-	-	72	-	-	2%
Test	Scenario	Test_Orders	66	45	-	-	21	0.592	0.542	10%
Test	Script	Test_Status	212	158	-	-	69	0.571	0.531	20%
		15	4354	3092	116	1271	176	0.507	0.557	3%

Figure 6: Impact Table

Code Change Request Cost Estimation

Code based Cost Report

Product : GEOS
System : NOSTRO
Project : CR2 Date : 11.09.01
Repository : d:\maintain\softrepo\tables

Metric	Quantity		Change Rate		Complexity		Quality		Adjusted Quantity		Monthly Product		Person Months
Stmnts:	333	X	0.500	X	1.028	X	0.595	=	101	/	500	=	0.20
Dat_Pt:	65	X	0.500	X	1.028	X	0.595	=	19	/	150	=	0.13
Obj_Pt:	425	X	0.500	X	1.028	X	0.595	=	129	/	140	=	0.93
Fct_Pt:	108	X	0.500	X	1.028	X	0.595	=	33	/	32	=	1.03
Median:	0	X	0.500	X	1.028	X	0.595	=	0	/	00	=	0.57