Understanding the Architecture of Enterprise Applications Through Their Dependencies

Neeraj Sangal

President
Lattix, Inc.
http://www.lattix.com

TS-9038

Use Dependencies to Manage Architecture

Represent and manage large-scale architectures

Learn to manage the architecture of enterprise applications using dependencies that span domains such as services, applications, configurations, and databases. This session includes a demo and real-life examples.

Agenda

Representing Large-Scale Systems Using a Dependency Structure Matrix (DSM)

Patterns, Rules, and Algorithms

Demo: Java™ Technology and Spring Framework

Demo: Java Technology, Hibernate, and Oracle

Real-World Examples

Q&A

Agenda

Representing Large-Scale Systems Using a Dependency Structure Matrix (DSM)

Patterns, Rules, and Algorithms

Demo: Java™ Technology and Spring Framework

Demo: Java Technology, Hibernate, and Oracle

Real-World Examples

Q&A

The Approach

- Provides a precise big picture view of the architecture
- Not code-centric—architecture encompasses configuration files (e.g., Spring framework, Hibernate configurations), databases, and other elements
- Highly scalable—has been applied to systems with thousands of elements
- Enables explicit management of architectural evolution

What Is a DSM?

		1	2	3	4
Module A	1	▪		X	X
Module B	2		▪	X	
Module C	3	X		▪	X
Module D	4				▪

Fig 1: A Simple DSM

		1	2	3
Module D	1	▪		
Module A-C	2	X	▪	
Module B	3		X	▪

Fig 3: Lower Triangular

		1	2	3	4
Module D	1	▪			
Module A	2	X	▪	X	
Module C	3	X	X	▪	
Module B	4			X	▪

Fig 2: Block Triangular after Partitioning

			1	2	3	4
Module D		1	▪			
A-C	Module A	2	X	▪	X	
	Module C	3	X	X	▪	
Module B		4			X	▪

Fig 4: Hierarchical

What Is a Dependency?

- Module A depends on a module B if there are explicit references in A to syntactic elements of B

```
[illegible]="[illegible]en.model">
FK1_CREDIT_CARD_ID foreign key (CREDIT_CARD_ID)
{
references BILLING_DETAILS;
<property name="accountDao" ref="accountDao"/>
   <property name="categoryDao" ref="categoryDao"/>
alter table BILLING_DETAILS add constraint FK1_USER_ID
        System.out.println(“Input:” + sb.toString());
</bean>
foreign key (USER_ID) references USERS;
<bean id="customerLocation“
}
alter table BANK_ACCOUNT [illegible]="[illegible]SqlMapAccountDao">
FK1_BANK_ACCOUNT_ID foreign key (BANK_ACCOUNT_ID)
references BILLING_DETAILS;
<property name="sqlMapClient" ref="sqlMapClient"/>
</bean>
</many-to-one></class>
</hibernate-mapping>
```

Agenda

Representing Large-Scale Systems Using a Dependency Structure Matrix (DSM)

Patterns, Rules, and Algorithms

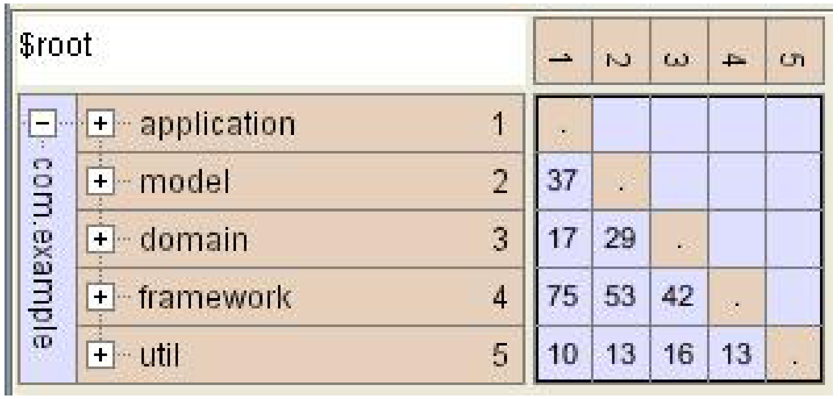
Demo: Java Technology and Spring Framework

Demo: Java Technology, Hibernate, and Oracle

Real-World Examples

Q&A

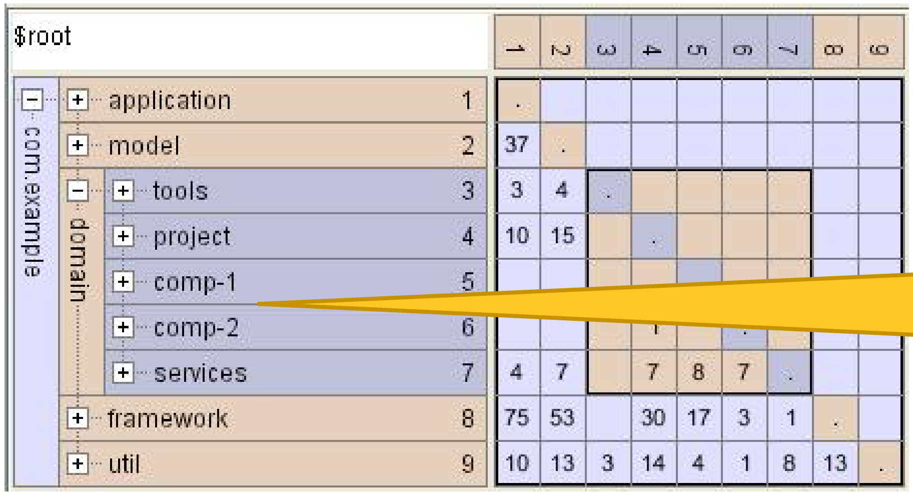
Architectural Patterns—I



$root

com.example		1	2	3	4	5
application	1	.				
model	2	37	.			
domain	3	17	29	.		
framework	4	75	53	42	.	
util	5	10	13	16	13	.

Layered System



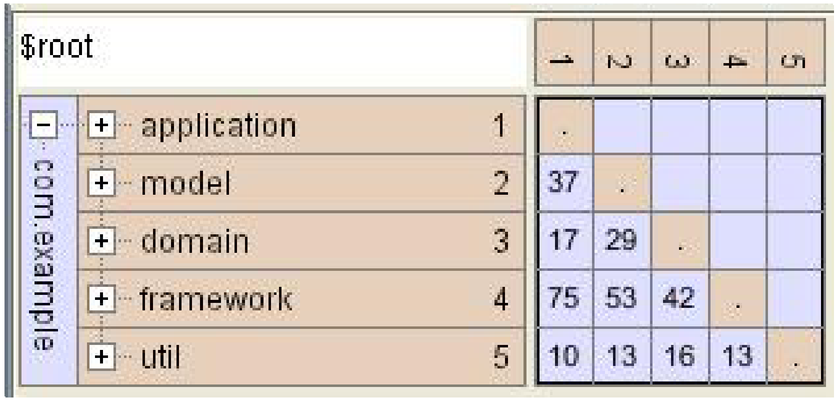
$root

com.example			1	2	3	4	5	6	7	8	9
application		1	.								
model		2	37	.							
domain	tools	3	3	4	.						
domain	project	4	10	15		.					
domain	comp-1	5					[illegible]				
domain	comp-2	6				1		.			
domain	services	7	4	7		7	8	7	.		
framework		8	75	53		30	17	3	1	.	
util		9	10	13	3	14	4	1	8	13	.

Private Components

Not Visible Outside “Domain”

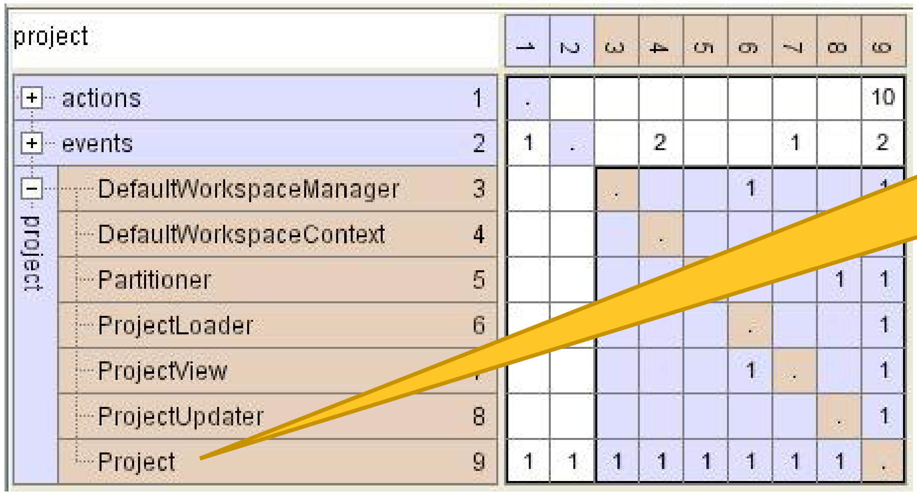
Architectural Patterns—II



$root

com.example		1	2	3	4	5
application	1	.				
model	2	37	.			
domain	3	17	29	.		
framework	4	75	53	42	.	
util	5	10	13	16	13	.

Imperfectly Layered System



project

		1	2	3	4	5	6	7	8	9
actions	1	.								10
events	2	1	.		2			1		2
project: DefaultWorkspaceManager	3			.			1			1
DefaultWorkspaceContext	4				.		[illegible]	[illegible]	[illegible]	[illegible]
Partitioner	5					[illegible]	[illegible]	[illegible]	1	1
ProjectLoader	6			[illegible]	[illegible]		.			1
ProjectView	7						1	.		1
ProjectUpdater	8								.	1
Project	9	1	1	1	1	1	1	1	1	.

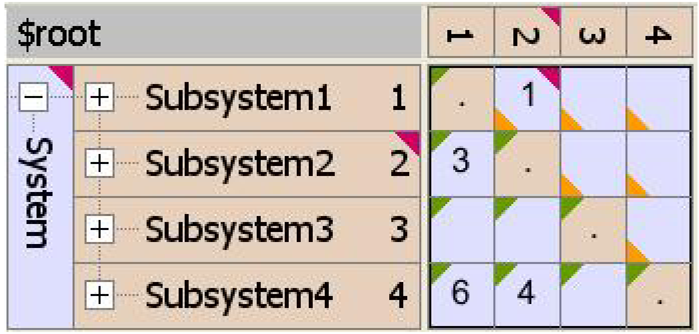
Change Propagator

Design Rules

- Succinct specification of acceptable and unacceptable dependencies between subsystems
- Each cell of the DSM represents design intent
- DSM offers a powerful way to visualize and specify design rules

Dependency Model = DSM + Design Rules

Design Rules



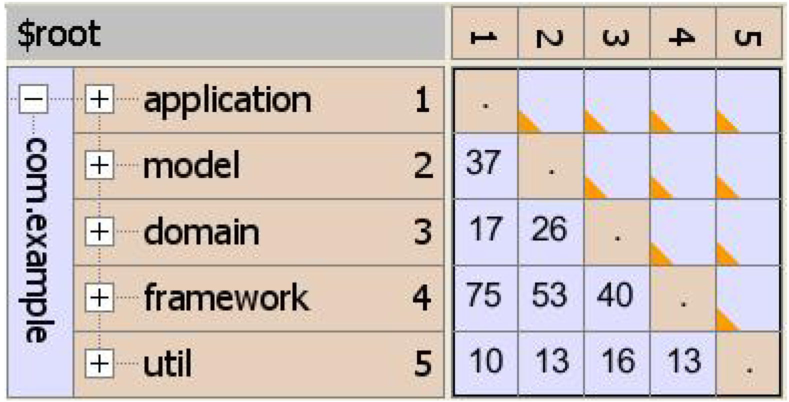
$root			1	2 🟥	3	4
System	Subsystem1 🟥	1	🟩 .	🟥 1 🟨	🟨	🟨
System	Subsystem2 🟥	2	🟩 3	🟩 .	🟨	🟨
System	Subsystem3	3	🟩	🟩	🟩 .	🟨
System	Subsystem4	4	🟩 6	🟩 4	🟩	🟩 .

DSM with Rules View

Green Triangle: Dependency Acceptable

Yellow Triangle: Dependency Unacceptable

Red Triangle: Rule Violation Discovered



$root			1	2	3	4	5
com.example	application	1	.	🟨	🟨	🟨	🟨
com.example	model	2	37	.	🟨	🟨	🟨
com.example	domain	3	17	26	.	🟨	🟨
com.example	framework	4	75	53	40	.	🟨
com.example	util	5	10	13	16	13	.

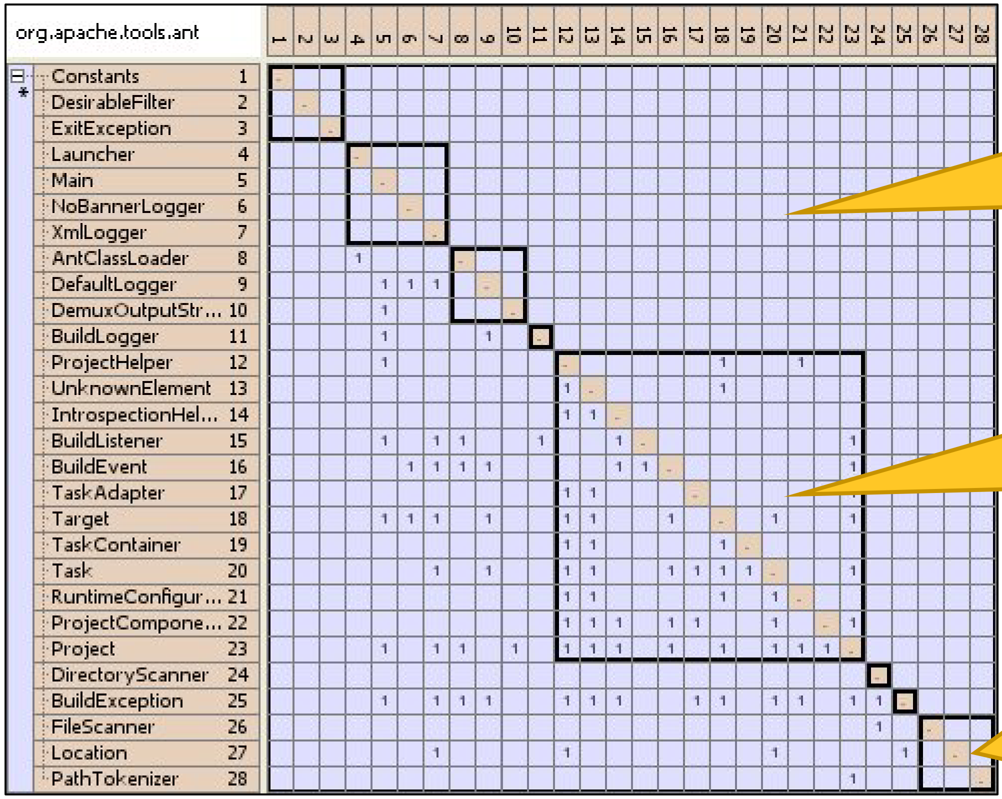
Rules for Layering

1. $root can-use $root
2. model cannot-use application
3. domain cannot-use application, model
4. framework cannot use application, model, domain
5. util cannot-use application, model, domain, framework

Types of DSM Algorithms

- Partitioning
  - Warfield, Gebala, Eppinger
- Clustering
  - Hartigan, Provider-Proximity
- Hierarchy improvements
  - Lattix

Algorithms: Partitioning



org.apache.tools.ant

		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28
Constants	1	-																											
DesirableFilter	2		-																										
ExitException	3			-																									
Launcher	4				-																								
Main	5					-																							
NoBannerLogger	6						-																						
XmlLogger	7							-																					
AntClassLoader	8				1				-																				
DefaultLogger	9					1	1	1		-																			
DemuxOutputStr…	10					1					-																		
BuildLogger	11					1				1		-																	
ProjectHelper	12					1							-						1			1							
UnknownElement	13												1	-					1										
IntrospectionHel…	14												1	1	-														
BuildListener	15					1		1	1			1			1	-								1					
BuildEvent	16						1	1	1	1					1	1	-							1					
TaskAdapter	17												1	1				-						1					
Target	18					1	1	1		1			1	1			1		-		1			1					
TaskContainer	19												1	1					1	-									
Task	20							1		1			1	1			1	1	1	1	-			1					
RuntimeConfigur…	21												1	1					1		1	-							
ProjectCompone…	22												1	1	1		1	1			1		-	1					
Project	23					1		1	1		1		1	1	1		1		1		1	1	1	-					
DirectoryScanner	24																								-				
BuildException	25					1		1	1	1			1	1	1			1	1		1	1		1	1	-			
FileScanner	26																								1		-		
Location	27							1					1								1					1		-	
PathTokenizer	28																							1					-

Partitioning Is Central to DSM

DEMO

Enterprise Architecture:
Java Technology and Spring Framework
Java Technology, Hibernate, and Oracle

Impact Analysis

Agenda

Representing Large-Scale Systems Using a Dependency Structure Matrix (DSM)

Patterns, Rules, and Algorithms

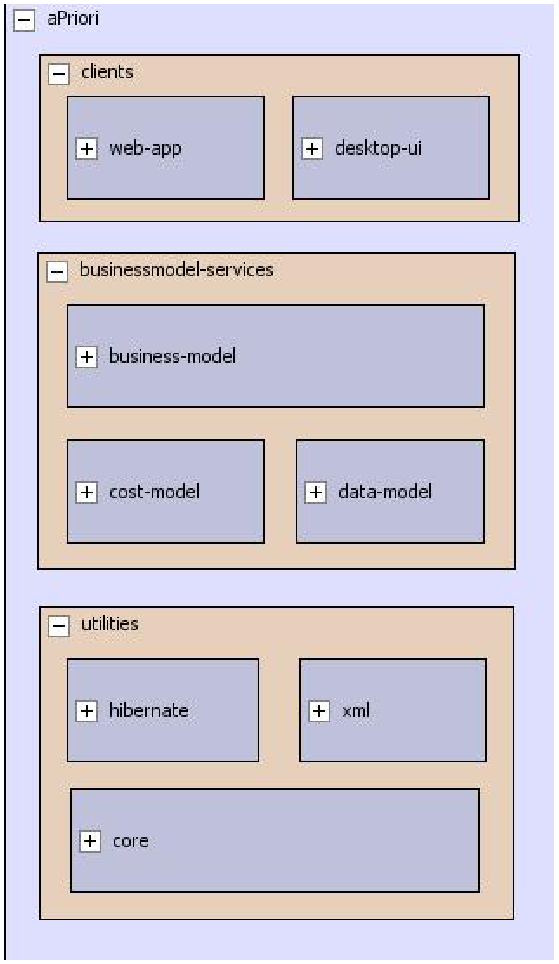
Demo: Java Technology and Spring Framework

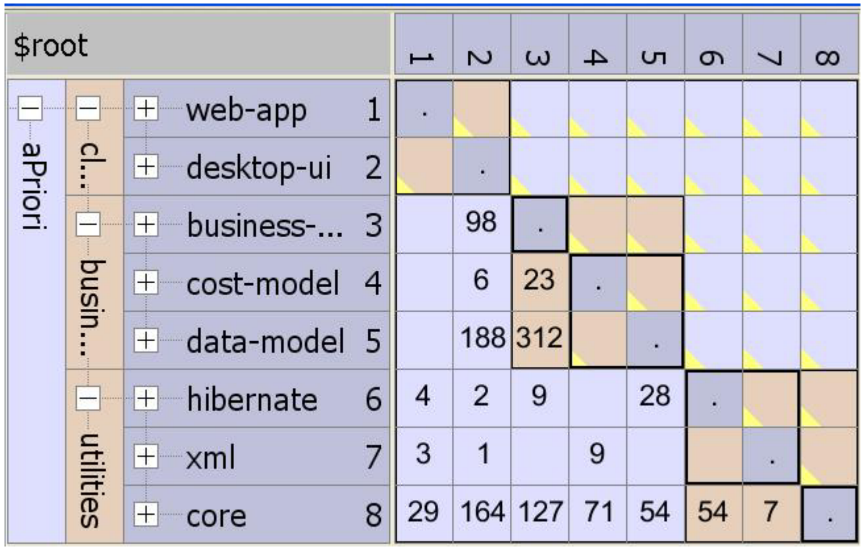
Demo: Java Technology, Hibernate, and Oracle

Real-World Examples

Q&A

Architectural Analysis: Extract Components

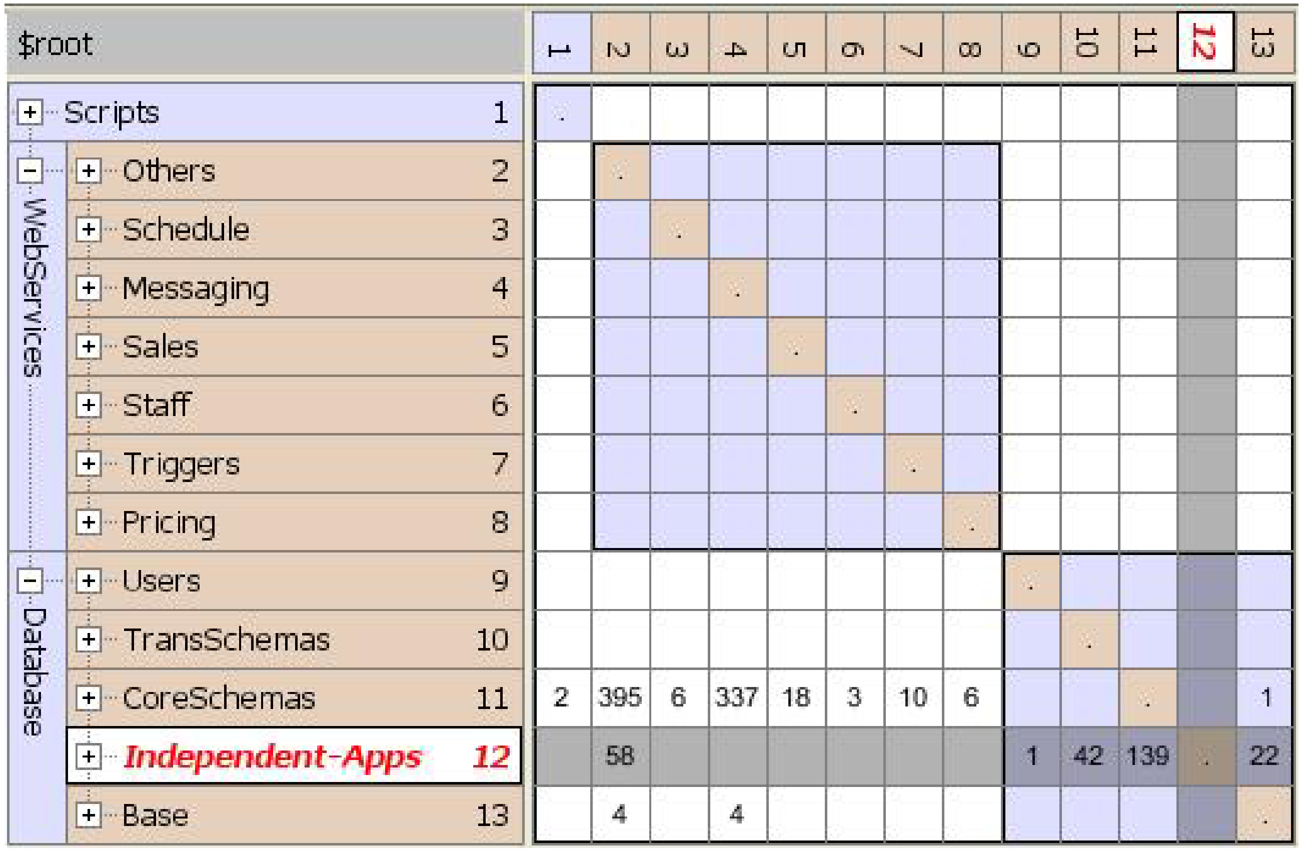




$root				1	2	3	4	5	6	7	8
aPriori	cl...	web-app	1	.							
		desktop-ui	2		.						
	busin...	business-...	3		98	.					
		cost-model	4		6	23	.				
		data-model	5		188	312		.			
	utilities	hibernate	6	4	2	9		28	.		
		xml	7	3	1		9			.	
		core	8	29	164	127	71	54	54	7	.

Extracting Components Requires Splitting Up Business Logic

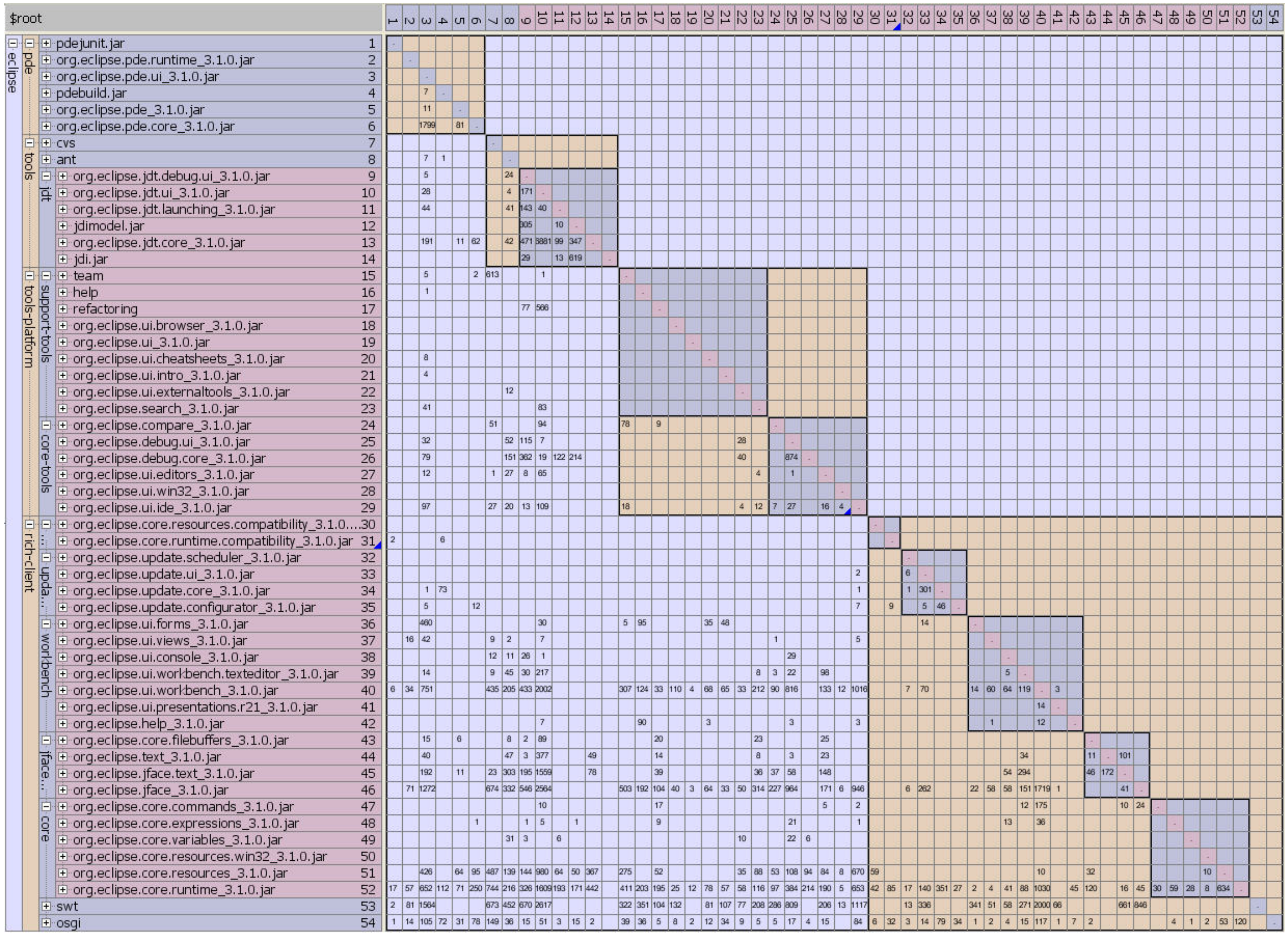
Risk Management: Impact Analysis



$root			1	2	3	4	5	6	7	8	9	10	11	12	13
Scripts		1	.												
WebServices	Others	2		.											
	Schedule	3			.										
	Messaging	4				.									
	Sales	5					.								
	Staff	6						.							
	Triggers	7							.						
	Pricing	8								.					
Database	Users	9									.				
	TransSchemas	10										.			
	CoreSchemas	11	2	395	6	337	18	3	10	6			.		1
	Independent-Apps	*12*		58							1	42	139	.	22
	Base	13		4		4									.

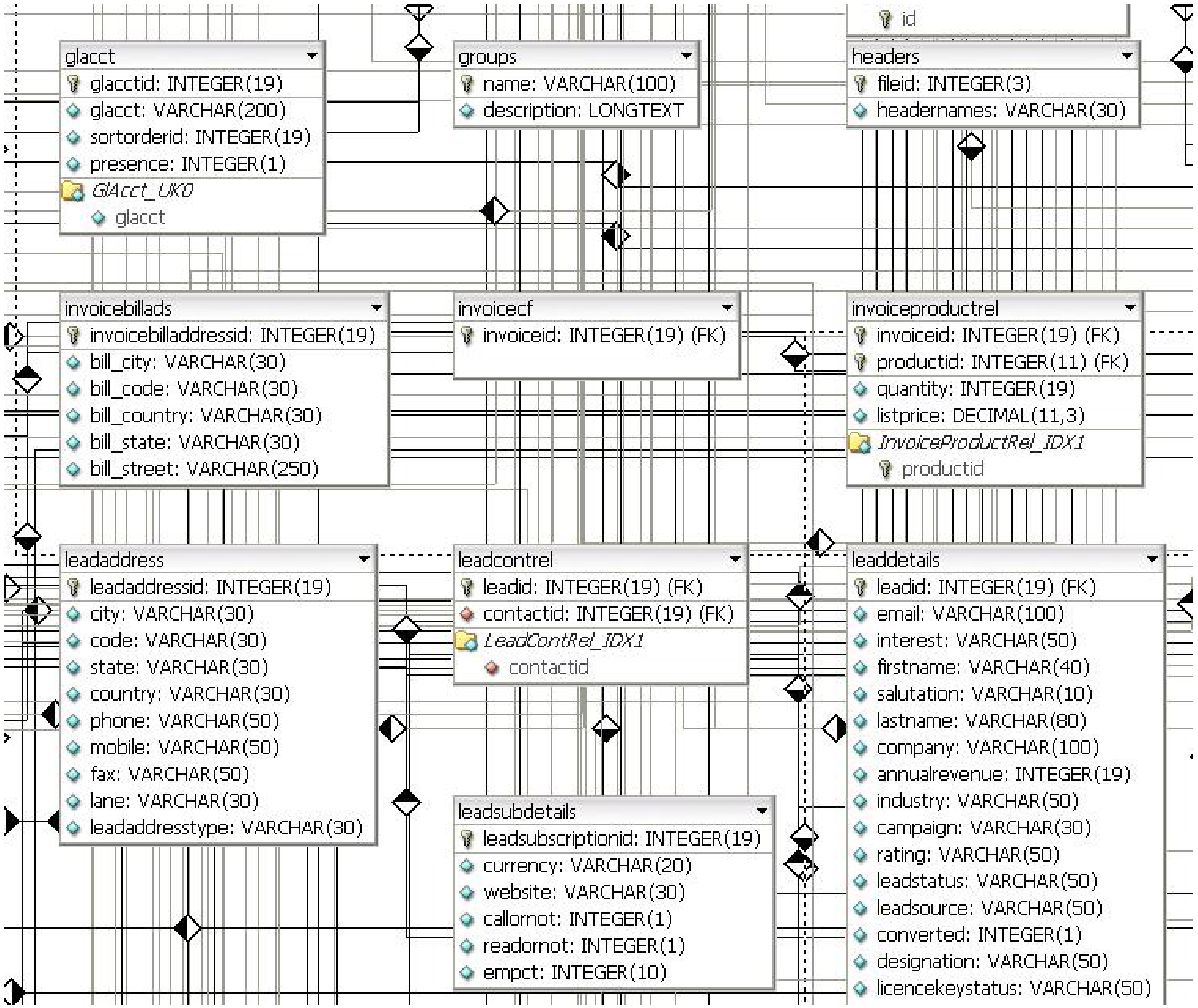
Impact DSM: Subset Showing What’s Affected

Dependency Model View of Eclipse Platform

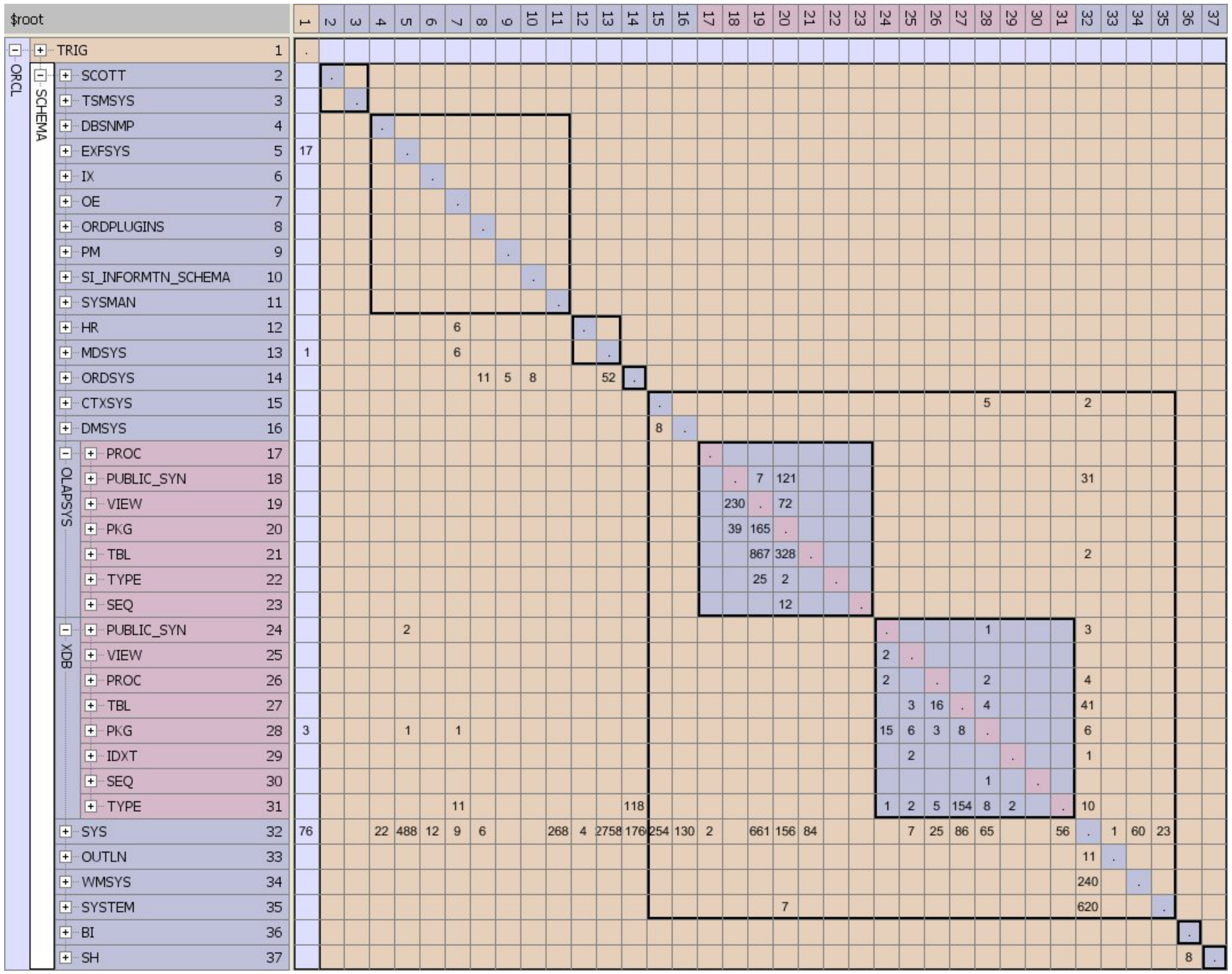


	$root		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54
eclipse / pde	pdejunit.jar	1	-																																																					
	org.eclipse.pde.runtime_3.1.0.jar	2		-																																																				
	org.eclipse.pde.ui_3.1.0.jar	3			-																																																			
	pdebuild.jar	4		7	-																																																		
	org.eclipse.pde_3.1.0.jar	5		11		-																																																	
	org.eclipse.pde.core_3.1.0.jar	6		1799		81	-																																																
tools	cvs	7							-																																																
	ant	8		7	1				-																																														
jdt	org.eclipse.jdt.debug.ui_3.1.0.jar	9		5					24	-																																														
	org.eclipse.jdt.ui_3.1.0.jar	10		28					4	171	-																																													
	org.eclipse.jdt.launching_3.1.0.jar	11		44					41	143	40	-																																											
	jdimodel.jar	12									305		10	-																																										
	org.eclipse.jdt.core_3.1.0.jar	13		191		11	62		42	471	3881	99	347	-																																										
	jdi.jar	14									29		13	619		-																																									
tools-platform / support-tools	team	15		5			2	613			1					-																																							
	help	16		1													-																																							
	refactoring	17									77	566							-																																						
	org.eclipse.ui.browser_3.1.0.jar	18																		-																																					
	org.eclipse.ui_3.1.0.jar	19																			-																																				
	org.eclipse.ui.cheatsheets_3.1.0.jar	20		8																		-																																			
	org.eclipse.ui.intro_3.1.0.jar	21		4																				-																																		
	org.eclipse.ui.externaltools_3.1.0.jar	22								12																-																																	
	org.eclipse.search_3.1.0.jar	23		41							83														-																																
core-tools	org.eclipse.compare_3.1.0.jar	24							51			94					78		9							-																															
	org.eclipse.debug.ui_3.1.0.jar	25		32					52	115	7												28			-																														
	org.eclipse.debug.core_3.1.0.jar	26		79					151	362	19	122	214											40			874	-																													
	org.eclipse.ui.editors_3.1.0.jar	27		12				1	27	8	65													4		1		-																												
	org.eclipse.ui.win32_3.1.0.jar	28																												-																											
	org.eclipse.ui.ide_3.1.0.jar	29		97				27	20	13	109					18							4	12	7	27		16	4	-																										
rich-client / …	org.eclipse.core.resources.compatibility_3.1.0....	30																														-																									
	org.eclipse.core.runtime.compatibility_3.1.0.jar	31	2			6																										-																								
upda...	org.eclipse.update.scheduler_3.1.0.jar	32																																-																							
	org.eclipse.update.ui_3.1.0.jar	33																													2			6	-																						
	org.eclipse.update.core_3.1.0.jar	34		1	73																									1			1	301	-																					
	org.eclipse.update.configurator_3.1.0.jar	35		5			12																							7		9		5	46	-																				
workbench	org.eclipse.ui.forms_3.1.0.jar	36		460							30					5	95				35	48												14			-																			
	org.eclipse.ui.views_3.1.0.jar	37	16	42				9	2		7														1					5							-																		
	org.eclipse.ui.console_3.1.0.jar	38						12	11	26	1																29														-																	
	org.eclipse.ui.workbench.texteditor_3.1.0.jar	39		14				9	45	30	217													8	3	22		98													5	-																
	org.eclipse.ui.workbench_3.1.0.jar	40	6	34	751				435	205	433	2002					307	124	33	110	4	68	65	33	212	90	816		133	12	1016			7	70			14	60	64	119	-	3													
	org.eclipse.ui.presentations.r21_3.1.0.jar	41																																									14	-												
	org.eclipse.help_3.1.0.jar	42										7						90				3					3				3							1			12		-											
jface...	org.eclipse.core.filebuffers_3.1.0.jar	43		15		6			8	2	89							20							23				25																	-										
	org.eclipse.text_3.1.0.jar	44		40					47	3	377			49				14							8		3		23															34				11	-	101									
	org.eclipse.jface.text_3.1.0.jar	45		192		11		23	303	195	1559			78				39							36	37	58		148													54	294				46	172	-								
	org.eclipse.jface_3.1.0.jar	46	71	1272				674	332	546	2564					503	192	104	40	3	64	33	50	314	227	964		171	6	946			6	262			22	58	58	151	1719	1				41	-							
core	org.eclipse.core.commands_3.1.0.jar	47										10							17												5		2											12	175					10	24	-						
	org.eclipse.core.expressions_3.1.0.jar	48						1			1	5		1					9										21				1										13		36										-					
	org.eclipse.core.variables_3.1.0.jar	49								31	3		6												10			22	6																										-				
	org.eclipse.core.resources.win32_3.1.0.jar	50																																																				-			
	org.eclipse.core.resources_3.1.0.jar	51			426		64	95	487	139	144	980	64	50	367		275		52					35	88	53	108	94	84	8	670	59										10			32							10	-			
	org.eclipse.core.runtime_3.1.0.jar	52	17	57	652	112	71	250	744	216	326	1609	193	171	442		411	203	195	25	12	78	57	58	116	97	384	214	190	5	653	42	85	17	140	351	27	2	4	41	88	1030		45	120		16	45	30	59	28	8	634	-		
	swt	53	2	81	1564				673	452	670	2617					322	351	104	132		81	107	77	208	286	809		206	13	1117			13	336			341	51	58	271	2000	66				661	846							-	
	osgi	54	1	14	105	72	31	78	149	36	15	51	3	15	2		39	36	5	8	2	12	34	9	5	5	17	4	15		84	6	32	3	14	79	34	1	2	4	15	117	1	7	2					4	1	2	53	120		-

ER Diagram (Part of a CRM System)



DSM for an Oracle Database (Default)



	$root		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37
ORCL	TRIG	1	.																																				
SCHEMA	SCOTT	2		.																																		
	TSMSYS	3			.																																	
	DBSNMP	4				.																																
	EXFSYS	5	17				.																															
	IX	6						.																														
	OE	7							.																													
	ORDPLUGINS	8								.																												
	PM	9									.																											
	SI_INFORMTN_SCHEMA	10										.																										
	SYSMAN	11											.																									
	HR	12							6					.																								
	MDSYS	13	1						6						.																							
	ORDSYS	14								11	5	8			52	.																						
	CTXSYS	15															.													5				2					
	DMSYS	16															8	.																				
OLAPSYS	PROC	17																	.																			
	PUBLIC_SYN	18																		.	7	121												31					
	VIEW	19																		230	.	72																
	PKG	20																		39	165	.																
	TBL	21																			867	328	.											2					
	TYPE	22																			25	2		.														
	SEQ	23																				12			.													
XDB	PUBLIC_SYN	24					2																			.				1				3					
	VIEW	25																								2	.											
	PROC	26																								2		.		2				4					
	TBL	27																									3	16	.	4				41					
	PKG	28	3				1		1																15	6	3	8	.				6					
	IDXT	29																									2				.			1					
	SEQ	30																													1		.						
	TYPE	31							11							118											1	2	5	154	8	2		.	10					
	SYS	32	76			22	488	12	9	6			268	4	2758	176	254	130	2		661	156	84				7	25	86	65			56	.	1	60	23		
	OUTLN	33																																	11	.				
	WMSYS	34																																	240		.			
	SYSTEM	35																				7												620			.		
	BI	36																																				.	
	SH	37																																				8	.

Summary: Big Picture View That Scales Across Multiple Domains

- Highly scalable—represent massive systems to give you a precise big picture view
- Dependency analysis is not only for code, but can also be applied to a variety of domains
- Large parts of dependency extraction can be automated allowing you to test your architecture and to prevent it from eroding
- Critical visibility of the architecture is achieved very quickly—try it out on your own software!

For More Information

- Steven D. Eppinger, “Innovation at the Speed of Information”, Harvard Business Review, January 2001.
- Baldwin, C.Y. and Clark K.B., The Power of Modularity Volume 1, MIT Press, Cambridge, MA, 2000
- Herb Simon, The Sciences of the Artificial, 3rd Edition, MIT Press, Cambridge, MA 1996
- Neeraj Sangal, Ev Jordan, Vineet Sinha, Daniel Jackson, “Using Dependency Models to Manage Complex Software Architecture”, OOPSLA 2005 http://sdg.lcs.mit.edu/pubs/2005/oopsla05-dsm.pdf
- DSM Website: http://www.dsmweb.org
- Lattix: http://www.lattix.com

Q&A

Neeraj Sangal

Frank Waldman

Understanding the Architecture of Enterprise Applications Through Their Dependencies

Neeraj Sangal

President
Lattix, Inc.
http://www.lattix.com

TS-9038