Applying Change Impact Analysis Test to Migration Test Case Extraction based on IDAU and Graph Analysis Techniques

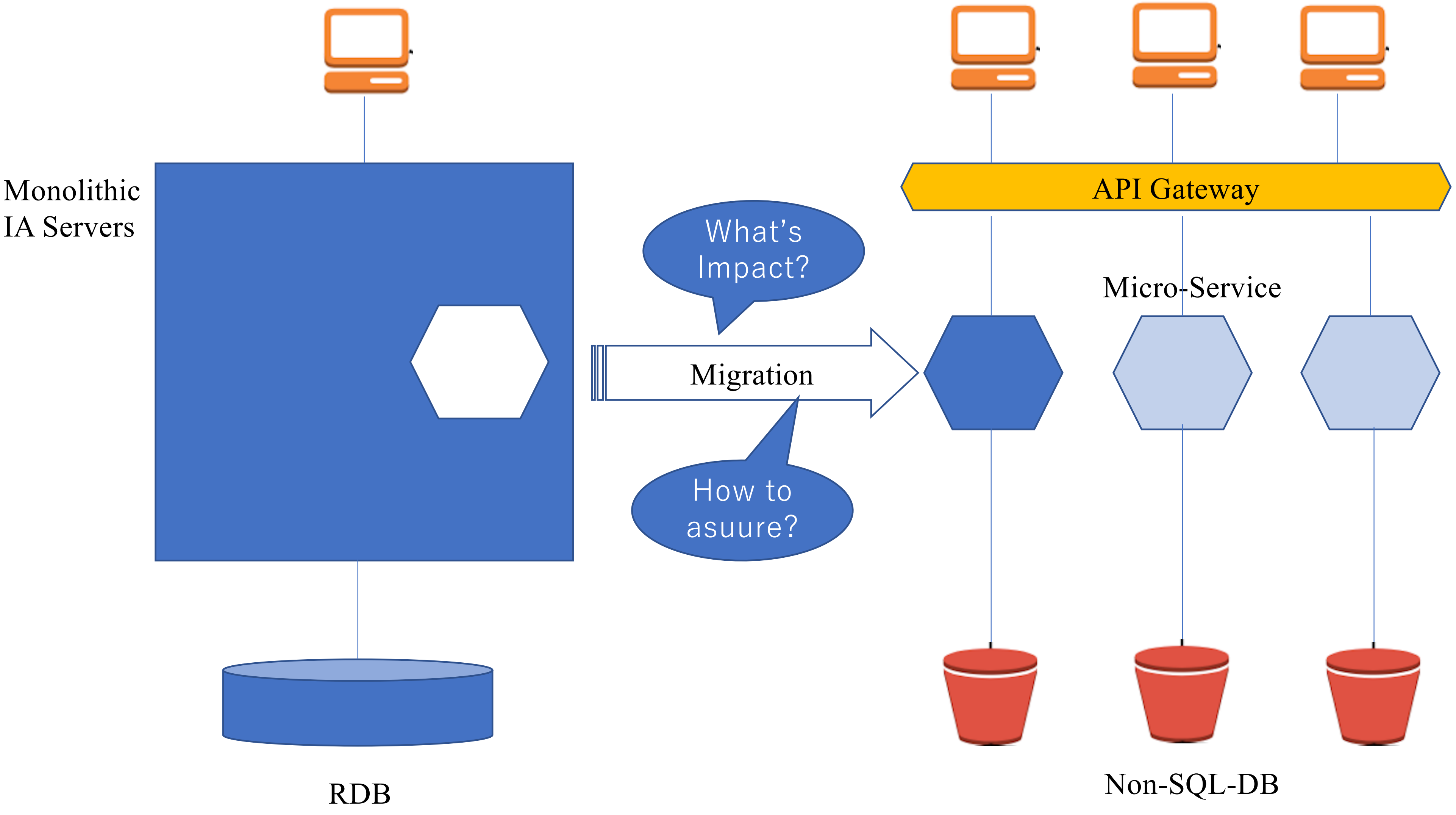
Tomohiro Takeda
University of Tsukuba, Tokyo, Japan

Contents

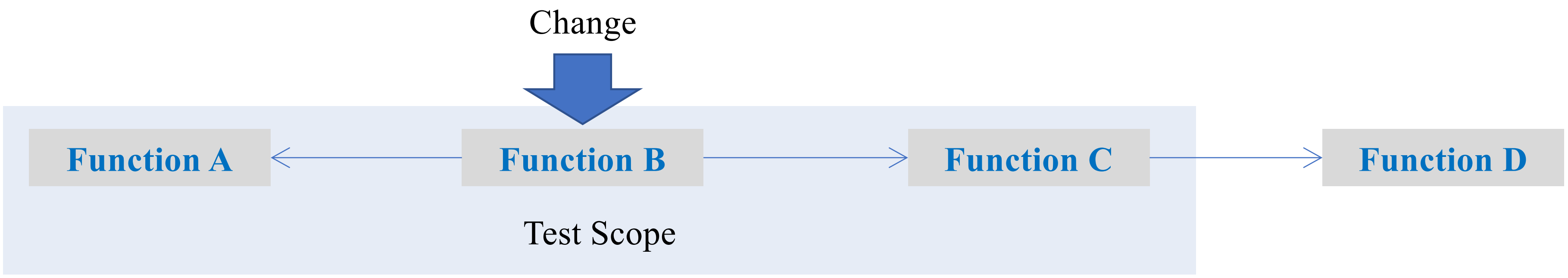
1. **Background**
2. **Previous Study**
3. **Code Based IDAU**
4. **Experiment**
5. **Conclusion**

1. Background

Recent Architecture Trend : Microservice



Impact Analysis Test



Change

Function A

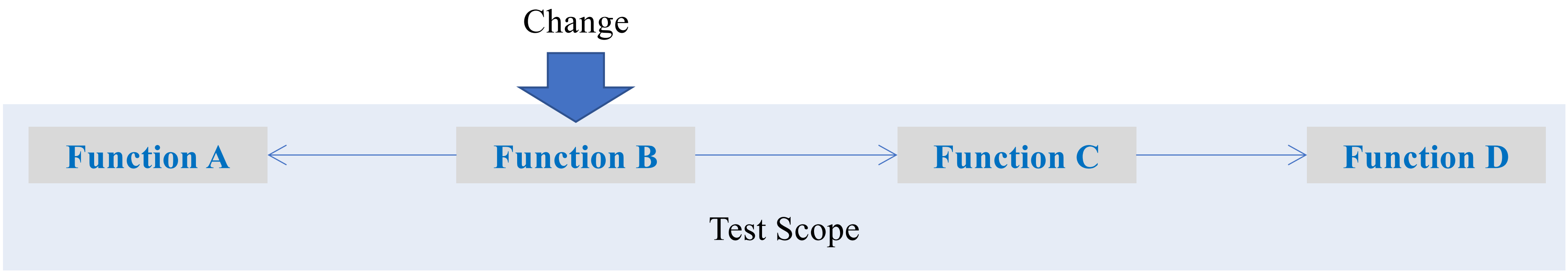
Function B

Function C

Function D

Test Scope

In this narrow range of test scope, there are much possible to lost any bugs.



Change

Function A

Function B

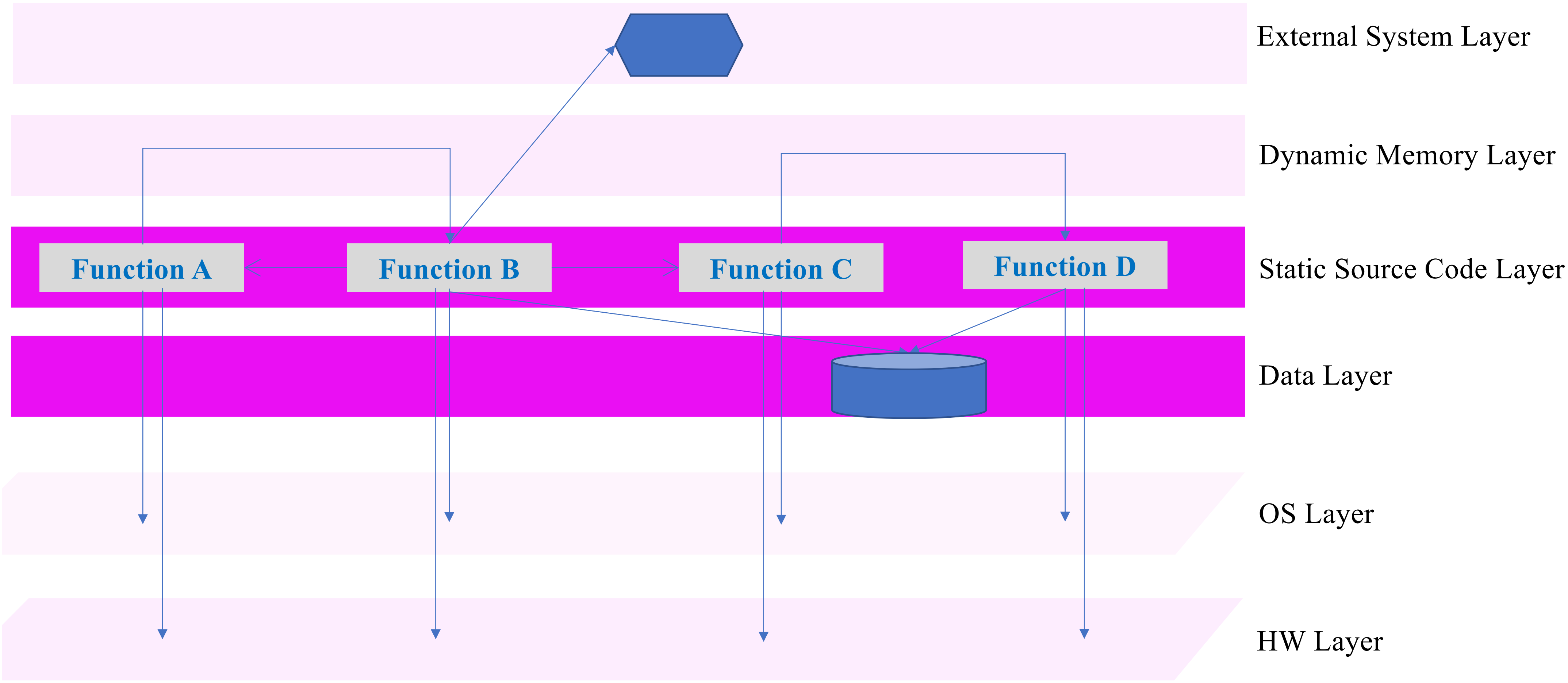
Function C

Function D

Test Scope

In this widen range of test case, this project test cost is increasing.

Why is impact analysis difficult?

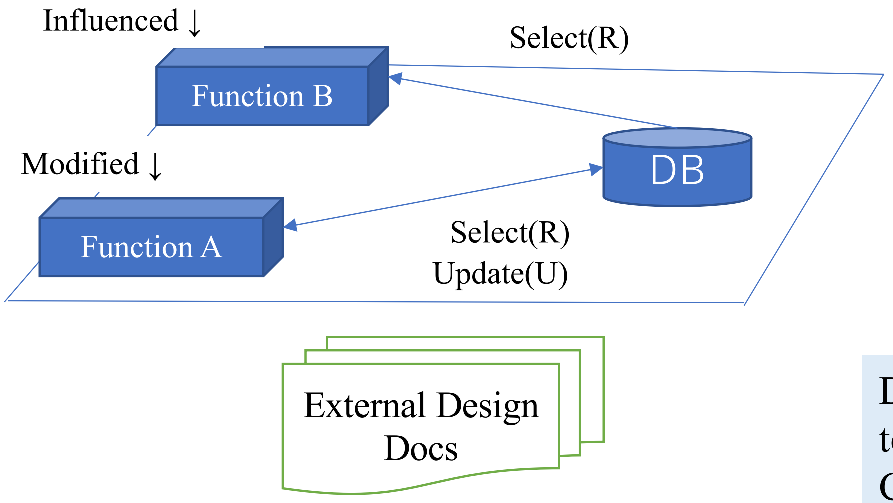


The application usually has several kinds of layer's dependencies.
This study based on IDAU test method focuses on the above Static Source Code and Data layer.

2. Previous Study

Impact Data All Used(IDAU) Test Methodology

Overview of IDAU



Create an expand CRUD table Based on the External design documents.

	Ds_1	Ds_2	So_1	So_2	So_3	So_4
Ta_1	R		Out			
Ta_2	RU	C	$InOut$			
Ta_3		R			Out	
Ta_5		R		Out		
Ta_6	CU					Out

Ta_2C	→	Ta_3R
Ta_2C	→	Ta_5R
Ta_2U	→	Ta_1R
Ta_2U	→	Ta_6U
Ta_6C	→	Ta_1R
Ta_6C	→	Ta_2R
Ta_6C	→	Ta_2U
Ta_6U	→	Ta_1R
Ta_6U	→	Ta_2R
Ta_6U	→	Ta_2U

Create the comprehensive combination of test case.

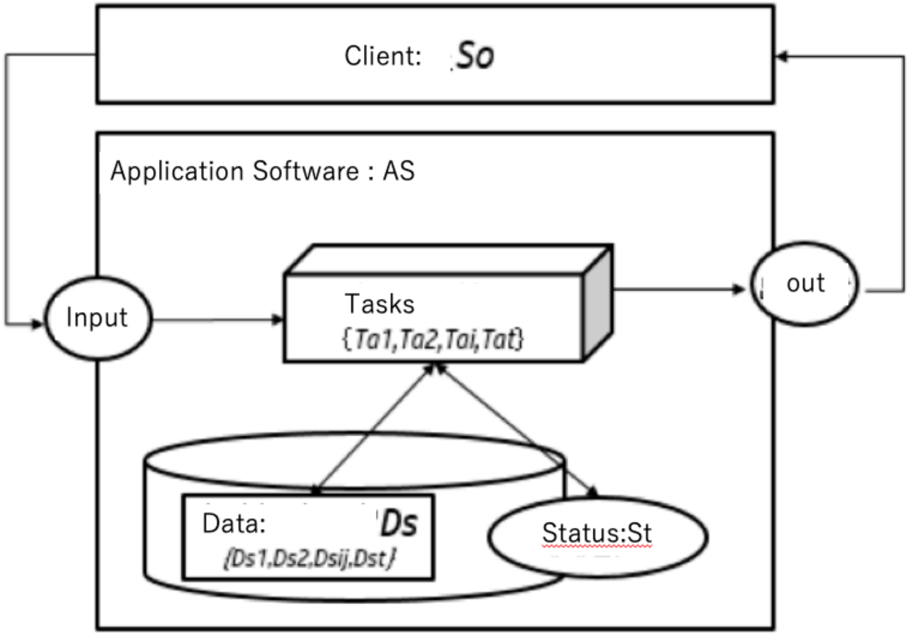
Decrease the number of test cases by the condition of CRUD Matrix.

		$P\{Ta\}$			
		C	R	U	D
$S\{Ta\}$	C	×	×	×	○
	R	○	-	○	-
	U	○	-	○	×
	D	○	-	○	×

Ta_2C	→	Ta_3R
Ta_2C	→	Ta_5R
Ta_2U	→	Ta_1R
Ta_2U	→	Ta_6U
Ta_6C	→	Ta_1R

Detect a a set of useful test cases.

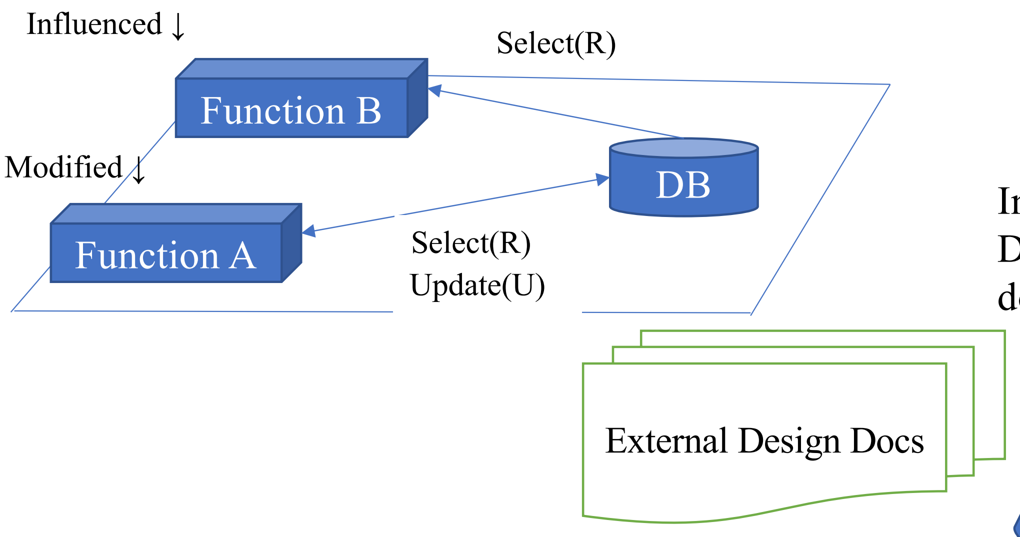
Restriction of IDAU



IDAU method has the strong restriction to be applied to the actual application as follows:

1. Each function is isolated to any other functions.
2. Each function only has dependencies via data.
3. Each function is called by external client application.
4. The call from Client application has input and output.

First Problem of IDAU

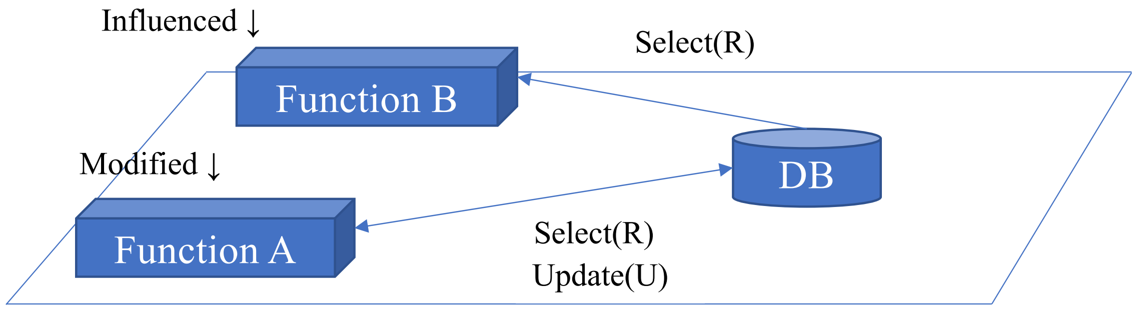


In the previous study, the relationships between Function and Data are only described on the external specification documents as a natural language.

		$P\{Ta\}$			
		C	R	U	D
$S\{Ta\}$	C	×	×	×	○
	R	○	-	○	-
	U	○	-	○	×
	D	○	-	○	×

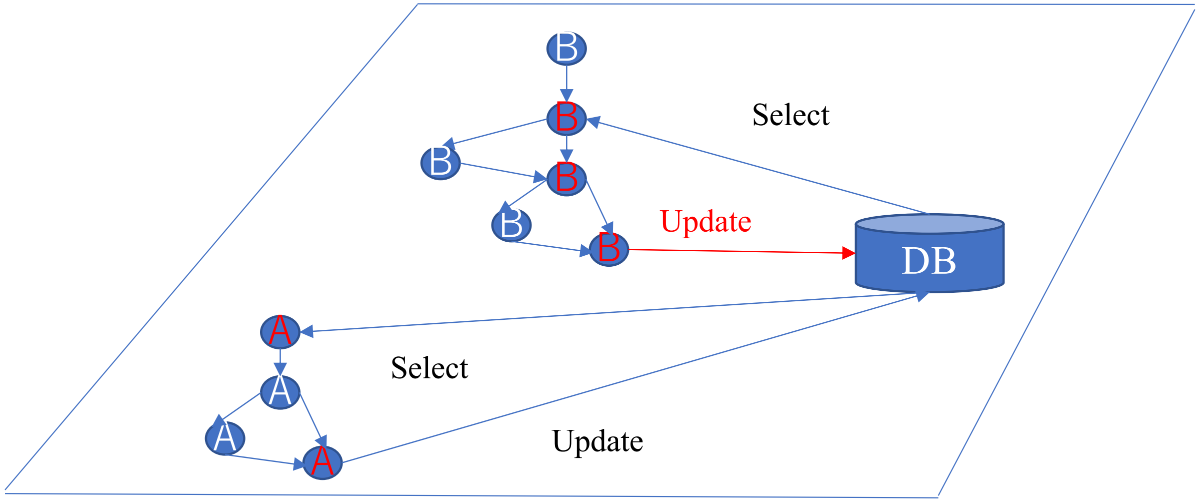
In the previous study, testers have to manually detect CRUD Information via the external design documents described as a natural languages. In this case, there are possible to lost any CRUD information.

Second Problem of IDAU



		$P\{Ta\}$			
		C	R	U	D
$S\{Ta\}$	C	×	×	×	○
	R	○	-	○	-
	U	○	-	○	×
	D	○	-	○	×

In the previous study, it is not possible to detect source code level CRUD information.



		$P\{Ta\}$			
		C	R	U	D
$S\{Ta\}$	C	×	×	×	○
	R	○	-	○	-
	U	○	-	○	×
	D	○	-	○	×

- In case of source code level CRUD analysis, we are able to detect U→R, U→U crud operations.

Solution & Experiments

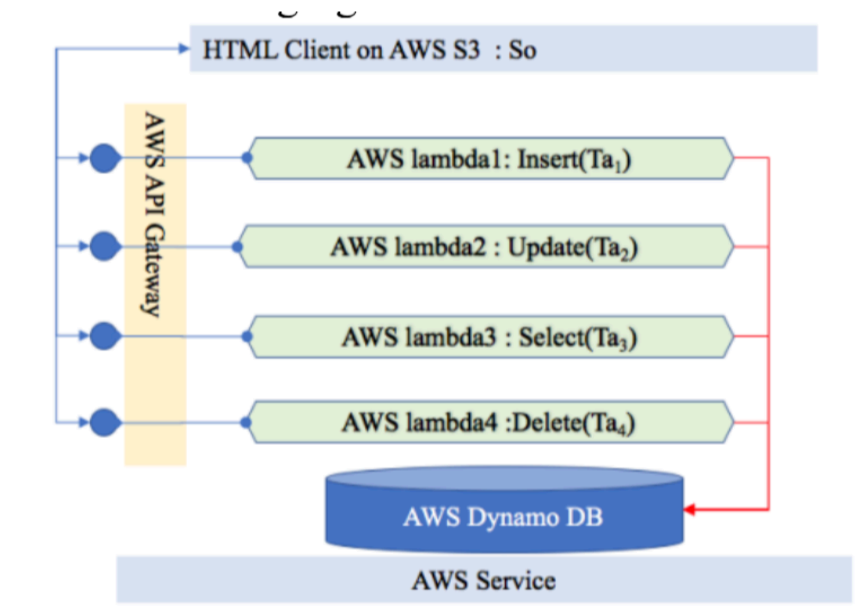
1. Lost of test cases by manual detection. → Automatic Test Case Detection Tool

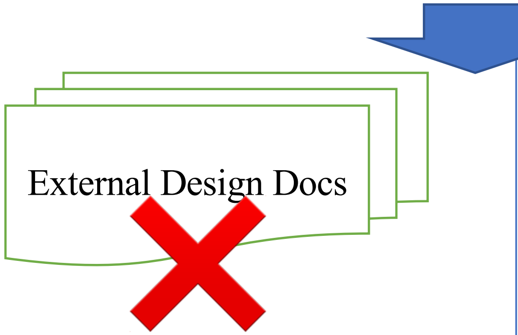
2. Lost of test cases by detection via the external design documents. → Code Based IDAU

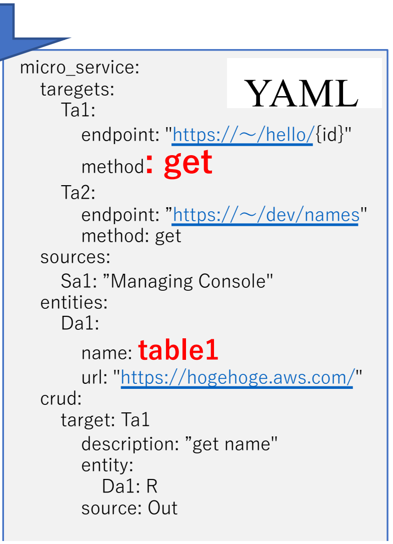
3. Experiment

3.1. Automatic Test Case Detection Tool by IDAU

Automatic Detection

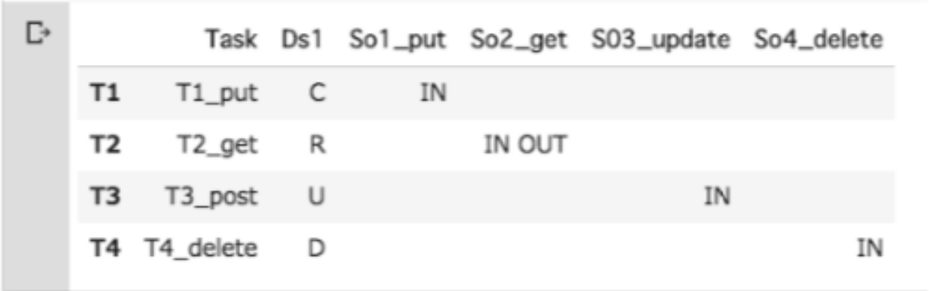




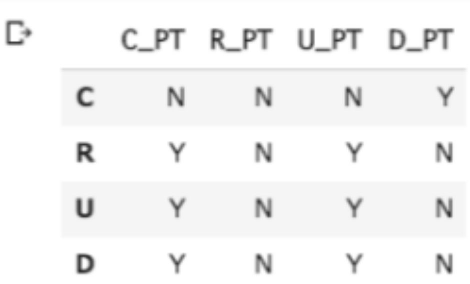


YAML

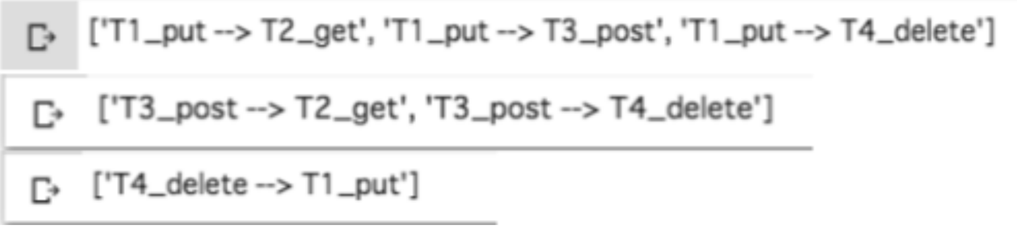
```
micro_service:
   taregets:
      Ta1:
         endpoint: "https://~/hello/{id}"
         method: get
      Ta2:
         endpoint: "https://~/dev/names"
         method: get
   sources:
      Sa1: "Managing Console"
   entities:
      Da1:
         name: table1
         url: "https://hogehoge.aws.com/"
   crud:
      target: Ta1
         description: "get name"
         entity:
            Da1: R
         source: Out
```



	Task	Ds1	So1_put	So2_get	S03_update	So4_delete
T1	T1_put	C	IN			
T2	T2_get	R		IN OUT		
T3	T3_post	U			IN	
T4	T4_delete	D				IN



	C_PT	R_PT	U_PT	D_PT
C	N	N	N	Y
R	Y	N	Y	N
U	Y	N	Y	N
D	Y	N	Y	N



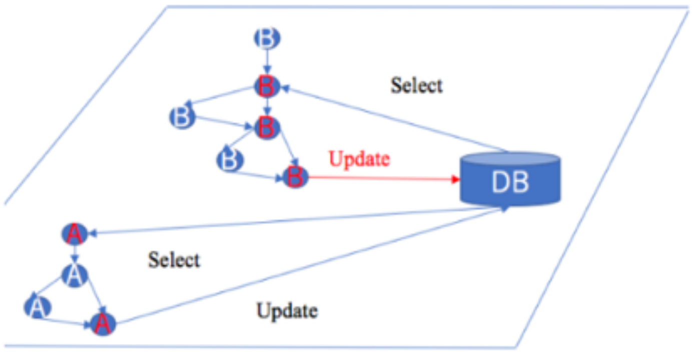
```
['T1_put --> T2_get', 'T1_put --> T3_post', 'T1_put --> T4_delete']
['T3_post --> T2_get', 'T3_post --> T4_delete']
['T4_delete --> T1_put']
```

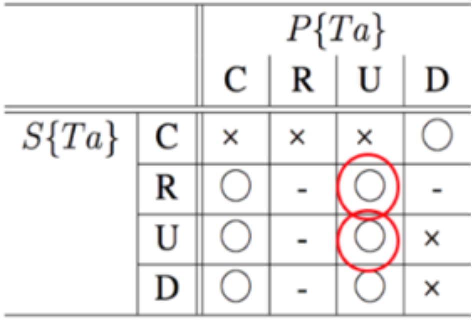
https://github.com/takedatmh/git/blob/master/ImpactAnalysis.ipynb

3. Experiment

3.2. Code Based IDAU

Process of Code Based IDAU



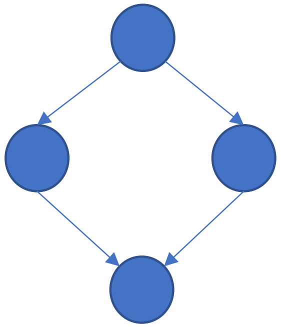


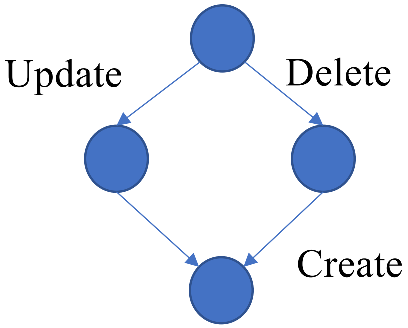
		P{Ta}			
		C	R	U	D
S{Ta}	C	×	×	×	○
	R	○	-	○	-
	U	○	-	○	×
	D	○	-	○	×

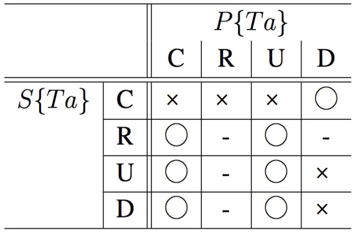
Source Code → Control Flow Graph → Detect CRUD Info → IDAU → Test Case

```
method(int i){
  if(i > 0){
    i = i + 100;
  } else {
    i = 0:
  }
}
```

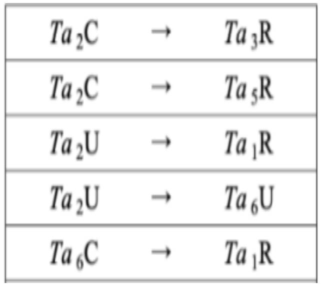
Jimple(Three Address)







		P{Ta}			
		C	R	U	D
S{Ta}	C	×	×	×	○
	R	○	-	○	-
	U	○	-	○	×
	D	○	-	○	×

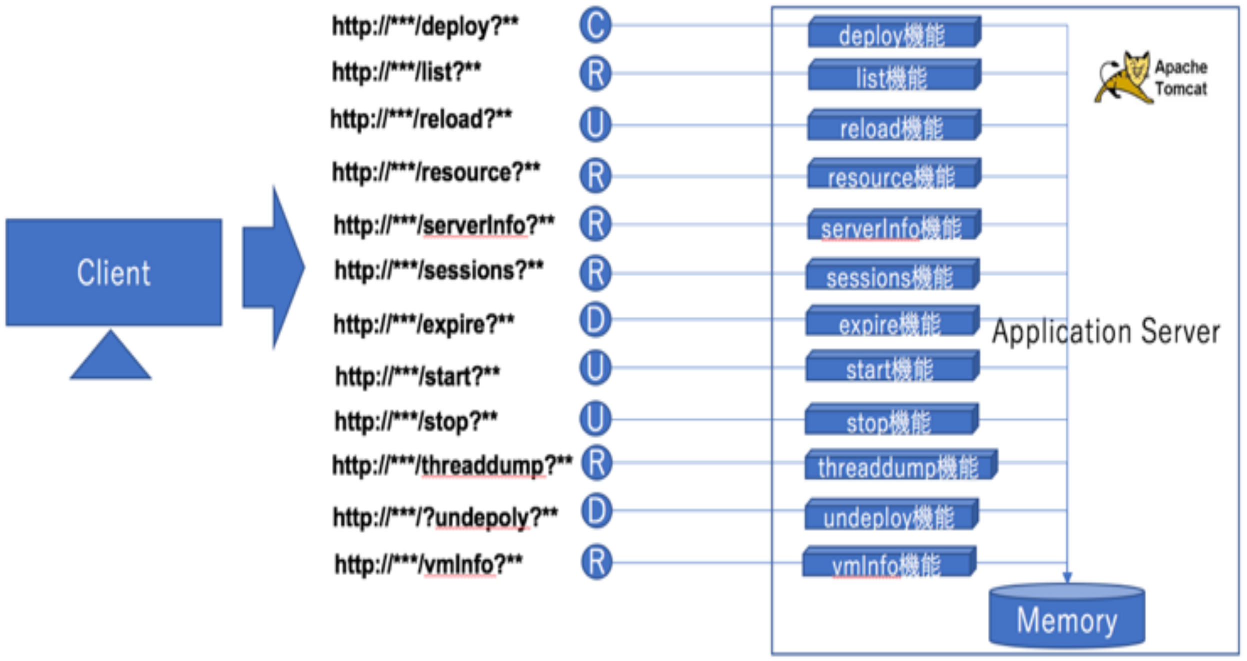


Ta_2C	→	Ta_3R
Ta_2C	→	Ta_5R
Ta_2U	→	Ta_1R
Ta_2U	→	Ta_6U
Ta_6C	→	Ta_1R

Source Code ⇨ Control Flow Graph ⇨ Detect CRUD Info ⇨ IDAU ⇨ Test Case

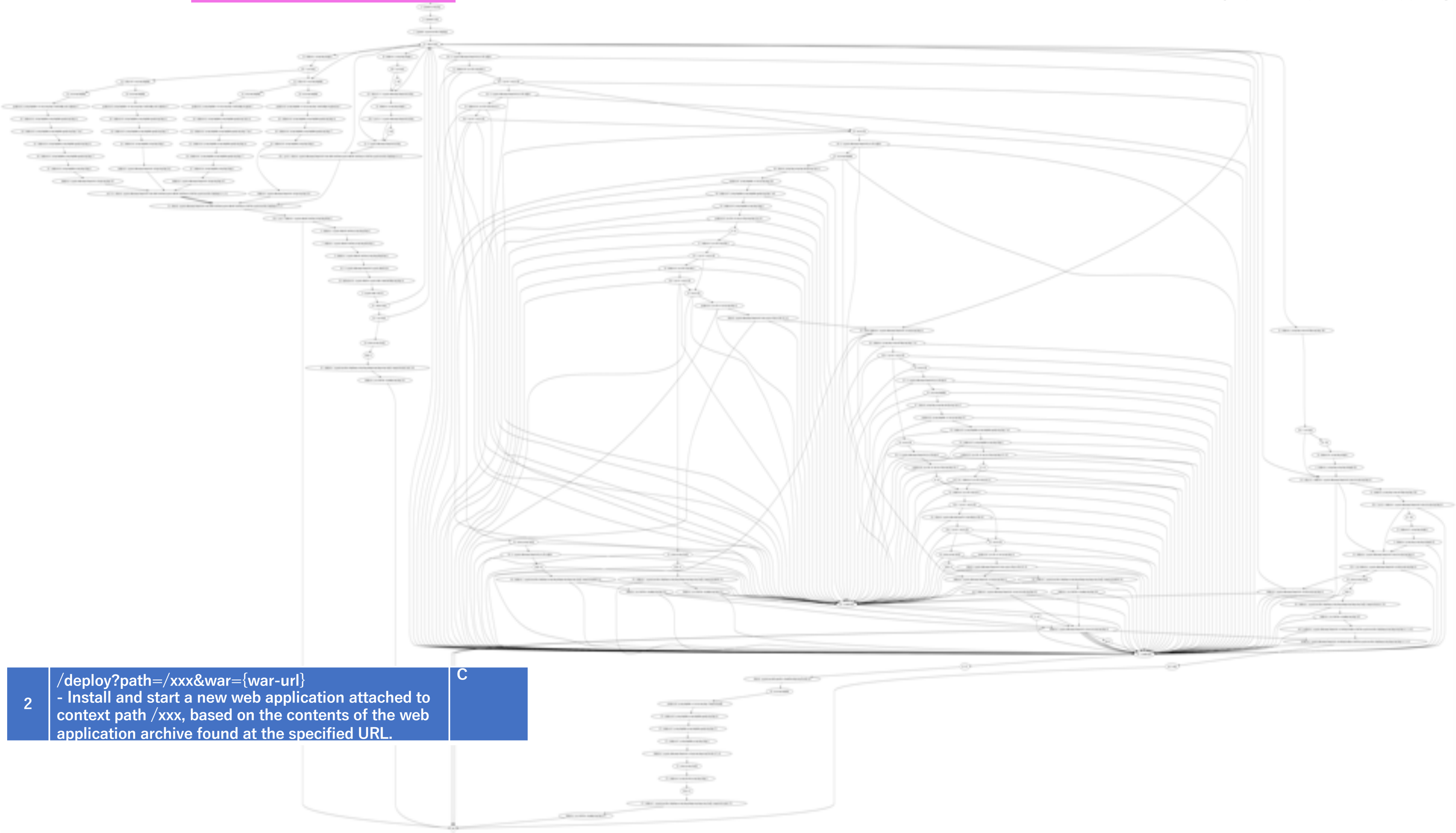
Target Application

ID	API External Specification	CRUD
1	/list - List the context paths of all currently installed web applications for this virtual host. Each context will be listed with the following format path:status: sessions. Where path is the context path. Status is either running or stopped. Sessions are the number of active Sessions.	R
2	/deploy?path=/xxx&war={war-url} - Install and start a new web application attached to context path /xxx, based on the contents of the web application archive found at the specified URL.	C
3	/reload?path=/xxx - Reload the Java classes and resources for the application at the specified path.	U
4	/resources?type=xxxx - Enumerate the available global JNDI resources, optionally limited to those of the specified type (fully qualified Java class name), if available.	R
5	/server info - Display system OS and JVM properties.	C
6	/sessions - Deprecated. Use expire.	R
7	/expire?path=/xxx - List session idle time information about the web application attached to context path /xxx for this virtual host.	D
8	/start?path=/xxx - Start the web application attached to context path /xxx for this virtual host.	U
9	/stop?path=/xxx - Stop the web application attached to context path /xxx for this virtual host.	U
10	/undeploy?path=/xxx - Shutdown and remove the web application attached to context path /xxx for this virtual host, and remove the underlying WAR file or document base directory.	D
11	/findLeak	R
12	/vminfo - Write some VM info.	R
13	/thread dump - Write a JVM thread dump.	C



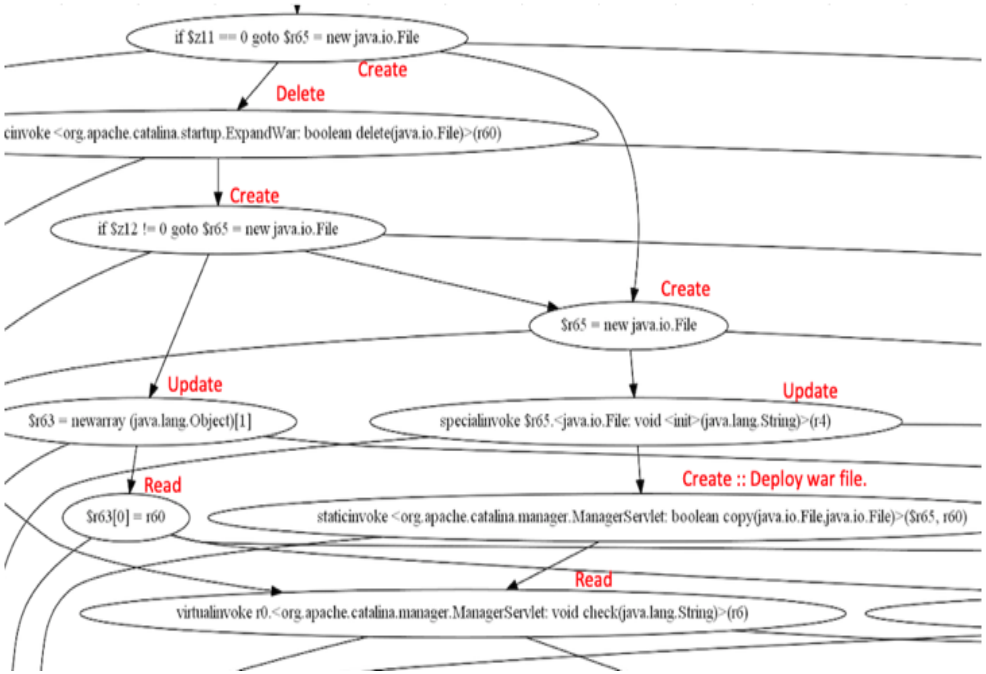
Source Code ⇨ **Control Flow Graph** ⇨ Detect CRUD Info ⇨ IDAU ⇨ Test Case

Control Flow Graph



2	/deploy?path=/xxx&war={war-url} - Install and start a new web application attached to context path /xxx, based on the contents of the web application archive found at the specified URL.	C

Source Code ⇨ Control Flow Graph ⇨ **Detect CRUD Info** ⇨ IDAU ⇨ Test Case


if $z11 == 0 goto $r65 = new java.io.File
Create
Delete
cinvoke <org.apache.catalina.startup.ExpandWar: boolean delete(java.io.File)>(r60)
Create
if $z12 != 0 goto $r65 = new java.io.File
Create
$r65 = new java.io.File
Update
Update
$r63 = newarray (java.lang.Object)[1]
specialinvoke $r65.<java.io.File: void <init>(java.lang.String)>(r4)
Read
Create :: Deploy war file.
$r63[0] = r60
staticinvoke <org.apache.catalina.manager.ManagerServlet: boolean copy(java.io.File,java.io.File)>($r65, r60)
Read
virtualinvoke r0.<org.apache.catalina.manager.ManagerServlet: void check(java.lang.String)>(r6)

External Specification Level			Source Code Level	
Task	CRUD	DS	CRUD	DS
deploy	C	ApplicatioinServer	C	LOG
deploy			U	config
deploy			R	config.length()
deploy			R	war
deploy			R	war.length()
deploy			R	debug
deploy			C	LOG
deploy			R	cn
deploy			R	cn.getPath
deploy			R	RequestUtil.filter
deploy			C	writter.println
deploy			R	sm.getString
deploy			R	path
deploy			R	cn.getName()
deploy			R	cn.getBaseName
deploy			R	cn.getDisplayName
deploy			R	host.findChild
deploy			R	context
deploy			R	update
deploy			R	smClient.getString
deploy			R	displayPath
deploy			R	config.startsWith("file:")
deploy			R	config.substring("file:".length()
deploy			U	config
deploy			R	war.startsWith("file:")
deploy			R	war.substring("file:".length()
deploy			U	war

Source Code ⇨ Control Flow Graph ⇨ Detect CRUD Info ⇨ IDAU ⇨ Test Case

ID	API	IDAU CRUD	CB-IDAU CRUD
1	/list	R	C,R,U
2	/deploy	C	C,R,U,D
3	/reload	U	C, R, U
4	/resources	R	C, R
5	/serverinfo	C	C, R
6	/sessions	R	C, R, D
7	/expire	D	C, R, D
8	/start	U	C, R, U
9	/stop	U	C, R, U
10	/undeploy	D	C, R, U, D
11	/findLeak	R	C, R
12	/vminfo	R	C, R
13	/threaddump	C	C, R

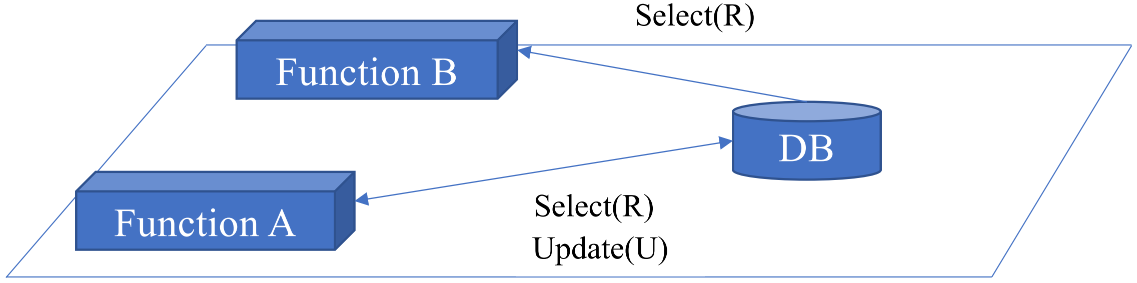
		$P\{Ta\}$			
		C	R	U	D
$S\{Ta\}$	C	×	×	×	○
	R	○	-	○	-
	U	○	-	○	×
	D	○	-	○	×

Method	Test Case Number
IDAU	65
CB-IDAU	78

4. Conclusion

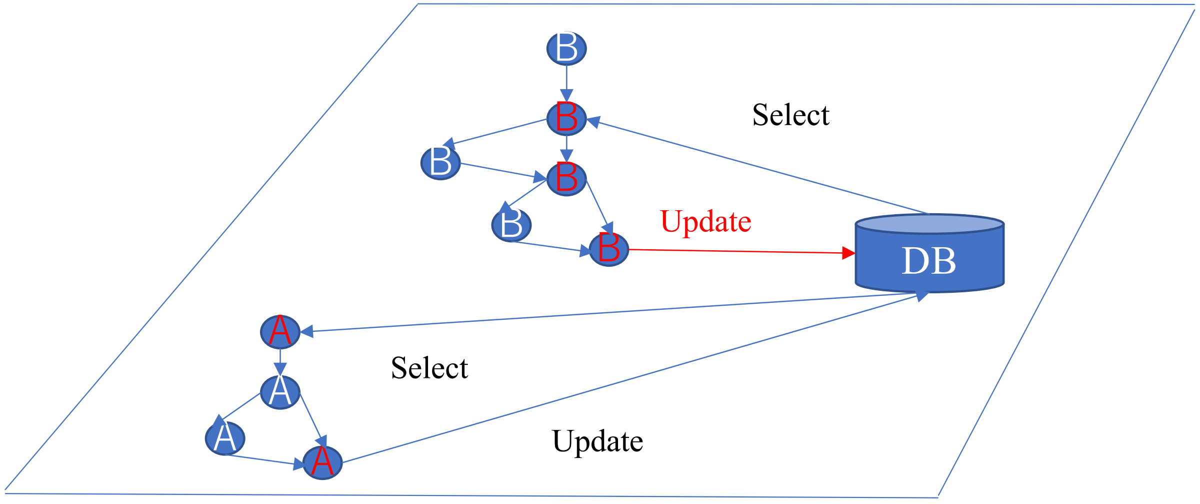
Result

■ Previous IDAU Method.



65 Test Cases

■ Code Based IDAU Method

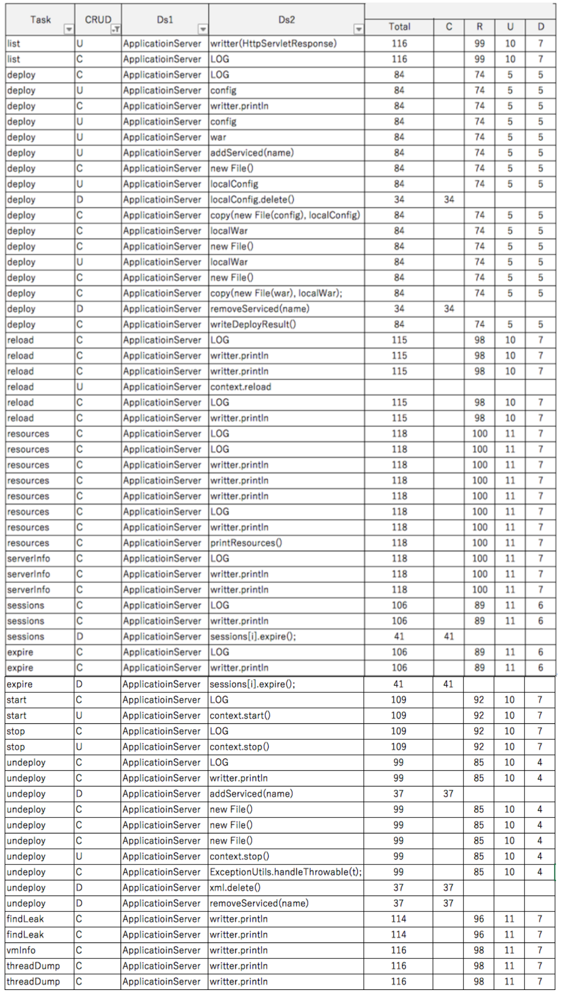


78 Test Cases

5. Future Work

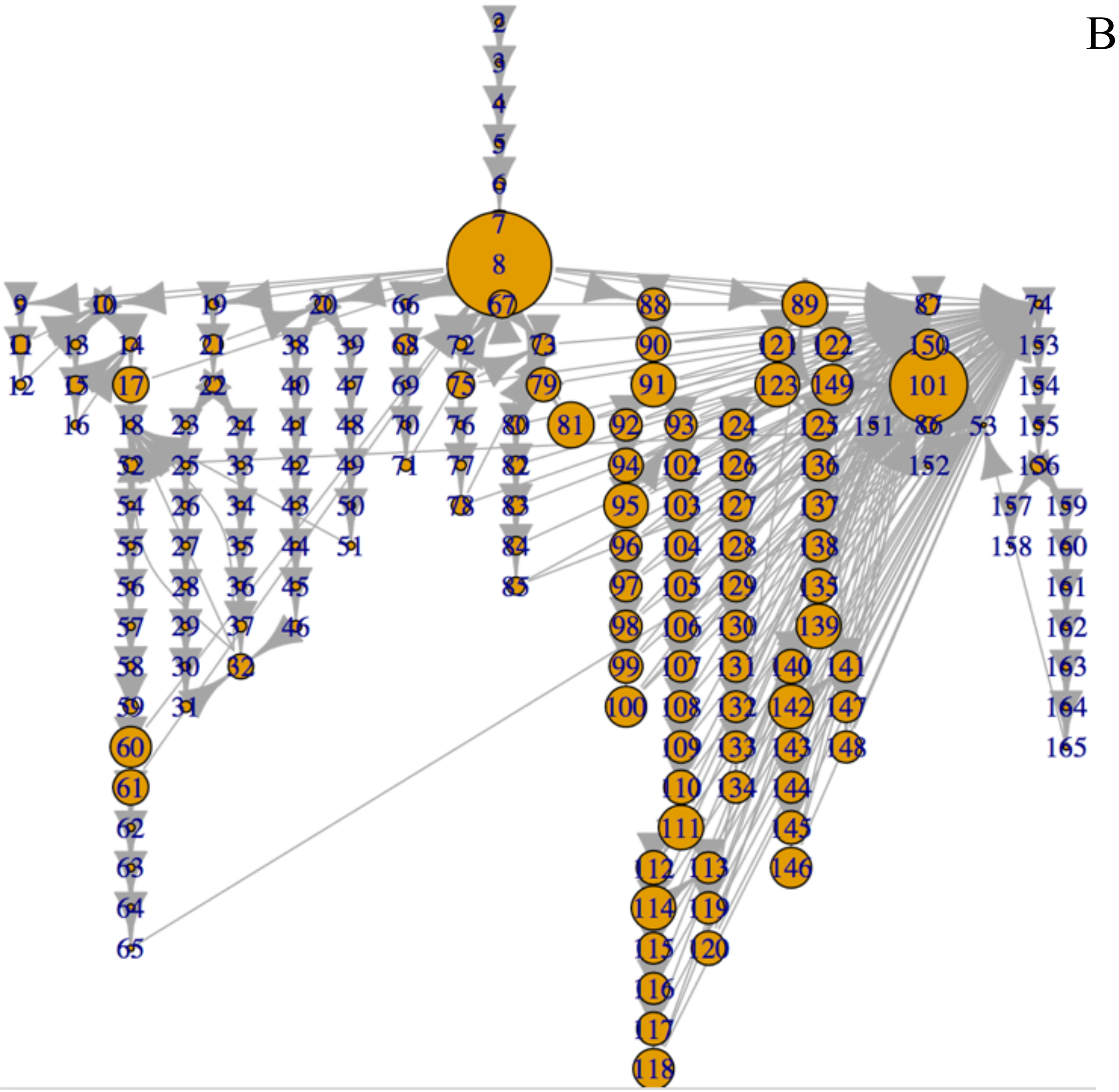
Graph Mining

Step Base CRUD count by Code Based IDAU



Task	CRUD	Ds1	Ds2	Total	C	R	U	D
list	U	ApplicatioinServer	writter(HttpServletResponse)	116		99	10	7
list	C	ApplicatioinServer	LOG	116		99	10	7
deploy	C	ApplicatioinServer	LOG	84		74	5	5
deploy	U	ApplicatioinServer	config	84		74	5	5
deploy	C	ApplicatioinServer	writter.println	84		74	5	5
deploy	U	ApplicatioinServer	config	84		74	5	5
deploy	U	ApplicatioinServer	war	84		74	5	5
deploy	U	ApplicatioinServer	addServiced(name)	84		74	5	5
deploy	C	ApplicatioinServer	new File()	84		74	5	5
deploy	U	ApplicatioinServer	localConfig	84		74	5	5
deploy	D	ApplicatioinServer	localConfig.delete()	34	34			
deploy	C	ApplicatioinServer	copy(new File(config), localConfig)	84		74	5	5
deploy	C	ApplicatioinServer	localWar	84		74	5	5
deploy	C	ApplicatioinServer	new File()	84		74	5	5
deploy	U	ApplicatioinServer	localWar	84		74	5	5
deploy	C	ApplicatioinServer	new File()	84		74	5	5
deploy	C	ApplicatioinServer	copy(new File(war), localWar);	84		74	5	5
deploy	D	ApplicatioinServer	removeServiced(name)	34	34			
deploy	C	ApplicatioinServer	writeDeployResult()	84		74	5	5
reload	C	ApplicatioinServer	LOG	115		98	10	7
reload	C	ApplicatioinServer	writter.println	115		98	10	7
reload	C	ApplicatioinServer	writter.println	115		98	10	7
reload	U	ApplicatioinServer	context.reload					
reload	C	ApplicatioinServer	LOG	115		98	10	7
reload	C	ApplicatioinServer	writter.println	115		98	10	7
resources	C	ApplicatioinServer	LOG	118		100	11	7
resources	C	ApplicatioinServer	LOG	118		100	11	7
resources	C	ApplicatioinServer	writter.println	118		100	11	7
resources	C	ApplicatioinServer	writter.println	118		100	11	7
resources	C	ApplicatioinServer	writter.println	118		100	11	7
resources	C	ApplicatioinServer	LOG	118		100	11	7
resources	C	ApplicatioinServer	writter.println	118		100	11	7
resources	C	ApplicatioinServer	printResources()	118		100	11	7
serverInfo	C	ApplicatioinServer	LOG	118		100	11	7
serverInfo	C	ApplicatioinServer	writter.println	118		100	11	7
serverInfo	C	ApplicatioinServer	writter.println	118		100	11	7
sessions	C	ApplicatioinServer	LOG	106		89	11	6
sessions	C	ApplicatioinServer	writter.println	106		89	11	6
sessions	D	ApplicatioinServer	sessions[i].expire();	41	41			
expire	C	ApplicatioinServer	LOG	106		89	11	6
expire	C	ApplicatioinServer	writter.println	106		89	11	6
expire	D	ApplicatioinServer	sessions[i].expire();	41	41			
start	C	ApplicatioinServer	LOG	109		92	10	7
start	U	ApplicatioinServer	context.start()	109		92	10	7
stop	C	ApplicatioinServer	LOG	109		92	10	7
stop	U	ApplicatioinServer	context.stop()	109		92	10	7
undeploy	C	ApplicatioinServer	LOG	99		85	10	4
undeploy	C	ApplicatioinServer	writter.println	99		85	10	4
undeploy	D	ApplicatioinServer	addServiced(name)	37	37			
undeploy	C	ApplicatioinServer	new File()	99		85	10	4
undeploy	C	ApplicatioinServer	new File()	99		85	10	4
undeploy	C	ApplicatioinServer	new File()	99		85	10	4
undeploy	U	ApplicatioinServer	context.stop()	99		85	10	4
undeploy	C	ApplicatioinServer	ExceptionUtils.handleThrowable(t);	99		85	10	4
undeploy	D	ApplicatioinServer	xml.delete()	37	37			
undeploy	D	ApplicatioinServer	removeServiced(name)	37	37			
findLeak	C	ApplicatioinServer	writter.println	114		96	11	7
findLeak	C	ApplicatioinServer	writter.println	114		96	11	7
vmInfo	C	ApplicatioinServer	writter.println	116		98	11	7
threadDump	C	ApplicatioinServer	writter.println	116		98	11	7
threadDump	C	ApplicatioinServer	writter.println	116		98	11	7

ID	Method	Test Case Number
1	CB-IDAU Step Base	5755



Bonachich Power Centrality Analysis

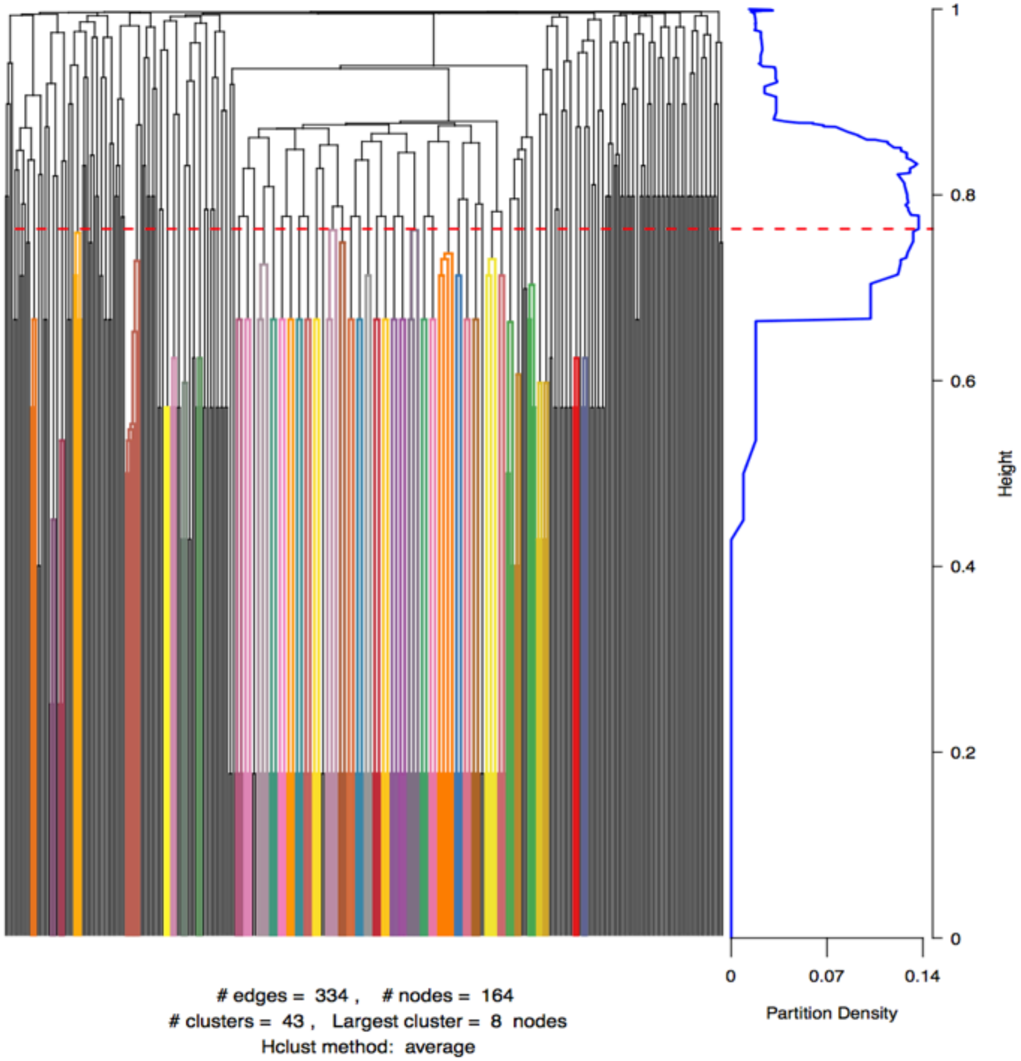
Node 8

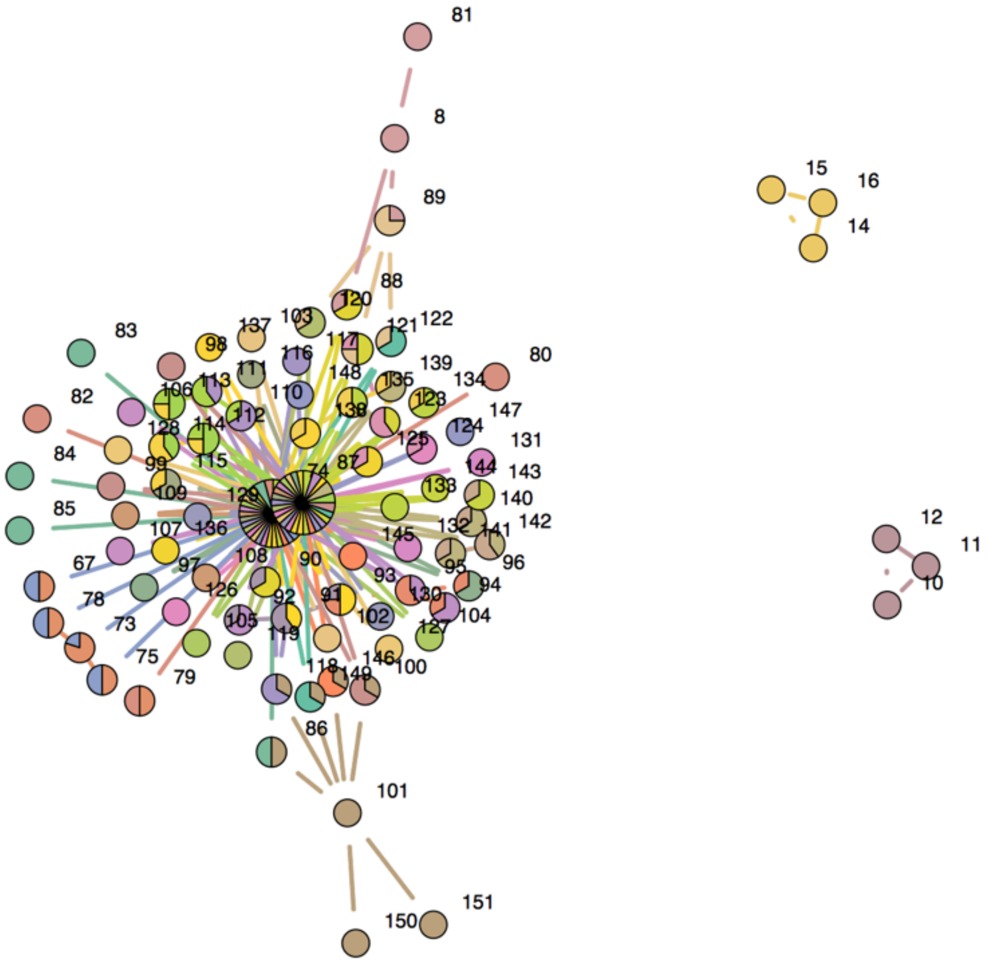
Null-Check varidation step

Node 101

Servlet method execution step

Link-Community Analysis





Result of Link-Community Analysis

Node 93, 123, 124, 125

File I/O steps

		3	4	5	7	8	9	10	12	14	15	16	17	18	19	20	21	22	23	24	25	Σ
Exception catch	74				■				■	■	■	■	■	■	■	■	■	■	■	■	■	30
Exception catch	87									■	■	■	■	■	■	■	■	■	■	■	■	27
Exception catch	121	■						■														3
If(String#endwith)	112		■	■																		3
If(String#endwith)	113		■	■																		3
new File()	93					■	■															3
new File()	123	■																				2
new File()	124	■																	■			2
new File()	125	■																				2
If(File#eixst)	110		■																			2
	Σ	4	3	2	1	1	1	1	1	2	2	2	2	2	2	2	2	2	3	2	2	

Thanks

Name: Tomohiro Takeda
Position: Student of Tsukuba University
Email: s1740116@s.tsukubai.ac.jp, takedatmh@gmail.com