Object Oriented Design Quality Metrics

References

- Analyze java package metrics in a graph database
- Calculate metrics
- jqassistant
- notebook walks through examples for integrating various packages with Neo4j
- OO Design Quality Metrics
- py2neo

Artifacts

Table 1

- List all the artifacts this notebook is based on

	artifactName	packages	types
0	axon-messaging-4.8.0.jar	64	762
1	axon-eventsourcing-4.8.0.jar	9	130
2	axon-test-4.8.0.jar	8	87
3	axon-configuration-4.8.0.jar	1	39
4	axon-modelling-4.8.0.jar	10	150
5	axon-disruptor-4.8.0.jar	1	22

Incoming Dependencies

Incoming dependencies are also denoted as "Fan-in", "Afferent Couplings" or "in-degree". These are the ones that use the listed package.

If these packages get changed, the incoming dependencies might be affected by the change. The more incoming dependencies, the harder it gets to change the code without the need to adapt the dependent code (“rigid code”). Even worse, it might affect the behavior of the dependent code in an unwanted way (“fragile code”).

Table 2

- Show the top 20 packages with the most incoming dependencies

- Set the "incomingDependencies" properties on Package nodes.

	packageName	incomingDependencies	incomingDependenciesWeight	incomingDependentTypes	incomingDependentInterfaces	inco
0	org.axonframework.messaging	8829	34707	322	65	
1	org.axonframework.eventhandling	4937	31619	291	54	
2	org.axonframework.commandhandling	1539	7499	123	18	
3	org.axonframework.serialization	1240	6854	138	15	
4	org.axonframework.messaging.annotation	1056	5608	151	19	
5	org.axonframework.common	917	2195	329	12	
6	org.axonframework.common.transaction	310	1162	75	5	
7	org.axonframework.messaging.unitofwork	258	1410	83	5	
8	org.axonframework.modelling.command	254	995	75	8	
9	org.axonframework.modelling.saga	242	1482	57	11	
10	org.axonframework.eventsourcing.eventstore.jdbc	218	1404	26	14	
11	org.axonframework.messaging.deadletter	198	1542	34	8	
12	org.axonframework.deadline	188	1520	36	8	
13	org.axonframework.monitoring	188	638	39	6	
14	org.axonframework.tracing	187	783	69	4	
15	org.axonframework.queryhandling	182	958	49	10	
16	org.axonframework.eventsourcing.eventstore	169	814	60	5	
17	org.axonframework.eventsourcing	134	657	41	6	
18	org.axonframework.config	115	1491	34	0	
19	org.axonframework.commandhandling.gateway	97	395	33	1	

Outgoing Dependencies

Outcoming dependencies are also denoted as "Fan-out", "Efferent Couplings" or "out-degree". These are the ones that are used by the listed package.

Code from other packages and libraries you’re depending on (outgoing) might change over time. The more outgoing changes, the more likely and frequently code changes are needed. This involves time and effort which can be reduced by automation of tests and version updates. Automated tests are crucial to reveal updates, that change the behavior of the code unexpectedly (“fragile code”). As soon as more effort is required, keeping up becomes difficult (“rigid code”). Not being able to use a newer version might not only restrict features, it can get problematic if there are security issues. This might force you to take “fast but ugly” solutions into account which further increases technical dept.

Table 3

- Show the top 20 packages with the most outgoing dependencies
- Set the "outgoingDependencies" properties on Package nodes.

	packageName	outgoingDependencies	outgoingDependentTypes	outgoingDependentInterfaces	outgoingDependentPackages	outg
0	org.axonframework.config	7974	213	84	46	
1	org.axonframework.test.aggregate	2223	92	34	16	
2	org.axonframework.eventhandling	1557	151	54	16	
3	org.axonframework.disruptor.commandhandling	1487	85	31	14	
4	org.axonframework.test.saga	1375	79	26	17	
5	org.axonframework.eventsourcing.eventstore.jdbc	1340	51	27	11	
6	org.axonframework.queryhandling	1116	80	28	11	
7	org.axonframework.eventhandling.pooled	1022	57	26	12	
8	org.axonframework.eventsourcing	976	91	31	16	
9	org.axonframework.modelling.command	827	91	33	15	
10	org.axonframework.modelling.command.inspection	781	73	28	10	
11	org.axonframework.commandhandling.distributed	752	72	27	12	
12	org.axonframework.commandhandling	642	70	28	9	
13	org.axonframework.eventsourcing.eventstore	603	64	25	16	
14	org.axonframework.eventhandling.deadletter.jdbc	581	49	20	8	
15	org.axonframework.deadline.dbscheduler	503	38	17	9	
16	org.axonframework.deadline.quartz	481	38	18	10	
17	org.axonframework.commandhandling.gateway	443	56	11	10	
18	org.axonframework.deadline.jobrunr	433	35	17	10	
19	org.axonframework.modelling.saga	398	59	22	9	

Instability

$$Instability = \frac{Outgoing\ Dependencies}{Outgoing\ Dependencies + Incoming\ Dependencies}$$

Instability is expressed as the ratio of the number of outgoing dependencies of a module (i.e., the number of packages that depend on it) to the total number of dependencies (i.e., the sum of incoming and outgoing dependencies).

Small values near zero indicate low *Instability*. With no outgoing but some incoming dependencies the Instability is zero which is denoted as maximally stable. Such code units are more rigid and difficult to change without impacting other parts of the system. If they are changed less because of that, they are considered stable.

Conversely, high values approaching one indicate high *Instability*. With some outgoing dependencies but no incoming ones the *Instability* is denoted as maximally unstable. Such code units are easier to change without affecting other modules, making them more flexible and less prone to cascading changes throughout the system. If they are changed more often because of that, they are considered unstable.

Table 4

- Show the top 20 packages with the lowest *Instability*

	p.fqn	p.name	instability	instabilityTypes	instabilityInterfaces	instabilityPackages	instabilityArtifacts	p.outgoingDe
0	org.axonframework.messaging	messaging	0.014840	0.098039	0.187500	0.101695	0.142857	
1	org.axonframework.common.transaction	transaction	0.018987	0.050633	0.000000	0.034483	0.200000	
2	org.axonframework.common	common	0.024468	0.043605	0.000000	0.012658	0.142857	
3	org.axonframework.eventhandling.scheduling	scheduling	0.090909	0.142857	0.000000	0.222222	0.250000	
4	org.axonframework.monitoring	monitoring	0.100478	0.152174	0.333333	0.230769	0.200000	
5	org.axonframework.common.annotation	annotation	0.120000	0.120000	0.000000	0.166667	0.250000	
6	org.axonframework.lifecycle	lifecycle	0.123457	0.233333	0.000000	0.176471	0.250000	
7	org.axonframework.serialization	serialization	0.124294	0.250000	0.318182	0.214286	0.200000	
8	org.axonframework.common.stream	stream	0.147059	0.166667	0.000000	0.125000	0.250000	
9	org.axonframework.messaging.annotation	annotation	0.221813	0.307339	0.406250	0.218750	0.142857	
10	org.axonframework.eventhandling	eventhandling	0.239760	0.341629	0.500000	0.258065	0.166667	
11	org.axonframework.common.jpa	jpa	0.272727	0.250000	1.000000	0.300000	0.200000	
12	org.axonframework.commandhandling	commandhandling	0.294360	0.362694	0.608696	0.333333	0.142857	
13	org.axonframework.common.legacyjpa	legacyjpa	0.300000	0.277778	1.000000	0.333333	0.250000	
14	org.axonframework.serialization.upcasting	upcasting	0.312500	0.083333	0.000000	0.333333	0.500000	
15	org.axonframework.messaging.unitofwork	unitofwork	0.322835	0.194175	0.583333	0.121951	0.142857	
16	org.axonframework.common.digest	digest	0.333333	0.333333	0.000000	0.333333	0.500000	
17	org.axonframework.messaging.deadletter	deadletter	0.346535	0.403509	0.500000	0.454545	0.333333	
18	org.axonframework.common.lock	lock	0.352113	0.363636	0.500000	0.222222	0.200000	
19	org.axonframework.messaging.correlation	correlation	0.358974	0.230769	0.400000	0.333333	0.333333	

Abstractness

$$Abstractness = \frac{abstract\ classes\ in\ category}{total\ number\ of\ classes\ in\ category}$$

Package *Abstractness* is expressed as the ratio of the number of abstract classes and interfaces to the total number of classes of a package.

Zero *Abstractness* means that there are no abstract types or interfaces in the package. On the other hand, a value of one means that there are only abstract types.

Table 5

- Show the top 30 packages with the lowest *Abstractness*

	fullQualifiedPackageName	packageName	abstractness	numberAbstractTypes	numberTypes
0	org.axonframework.eventsourcing.eventstore.leg...	legacyjpa	0.000000	0	10
1	org.axonframework.commandhandling.distributed....	commandfilter	0.000000	0	7
2	org.axonframework.serialization.json	json	0.000000	0	7
3	org.axonframework.serialization.xml	xml	0.000000	0	7
4	org.axonframework.deadline.dbscheduler	dbscheduler	0.000000	0	6
5	org.axonframework.eventhandling.scheduling.dbs...	dbscheduler	0.000000	0	6
6	org.axonframework.tracing.attributes	attributes	0.000000	0	6
7	org.axonframework.serialization.converters	converters	0.000000	0	5
8	org.axonframework.commandhandling.callbacks	callbacks	0.000000	0	4
9	org.axonframework.deadline.quartz	quartz	0.000000	0	4
10	org.axonframework.eventhandling.deadletter	deadletter	0.000000	0	4
11	org.axonframework.eventhandling.scheduling.java	java	0.000000	0	4
12	org.axonframework.eventhandling.tokenstore.jpa	jpa	0.000000	0	4
13	org.axonframework.test.server	server	0.000000	0	4
14	org.axonframework.eventhandling.scheduling.job...	jobrunr	0.000000	0	3
15	org.axonframework.util	util	0.000000	0	3
16	org.axonframework.modelling.saga.repository.le...	legacyjpa	0.000000	0	3
17	org.axonframework.eventhandling.tokenstore.inm...	inmemory	0.000000	0	2
18	org.axonframework.eventhandling.tokenstore.leg...	legacyjpa	0.000000	0	2
19	org.axonframework.messaging.interceptors.legac...	legacyvalidation	0.000000	0	2
20	org.axonframework.eventsourcing.eventstore.inm...	inmemory	0.000000	0	2
21	org.axonframework.modelling.command.legacyjpa	legacyjpa	0.000000	0	2
22	org.axonframework.modelling.saga.repository.in...	inmemory	0.000000	0	2
23	org.axonframework.common.digest	digest	0.000000	0	1
24	org.axonframework.common.io	io	0.000000	0	1
25	org.axonframework.eventhandling.interceptors	interceptors	0.000000	0	1
26	org.axonframework.disruptor.commandhandling	commandhandling	0.045455	1	22
27	org.axonframework.modelling.saga.repository.jdbc	jdbc	0.100000	1	10
28	org.axonframework.eventhandling.deadletter.jpa	jpa	0.111111	1	9
29	org.axonframework.eventhandling.tokenstore.jdbc	jdbc	0.111111	1	9

Distance from the main sequence

The *main sequence* is a imaginary line that represents a good compromise between *Abstractness* and *Instability*. A high distance to this line may indicate problems. For example is very *stable* (rigid) code with low abstractness hard to change.

Read more details on that in OO Design Quality Metrics and Calculate metrics.

Table 6

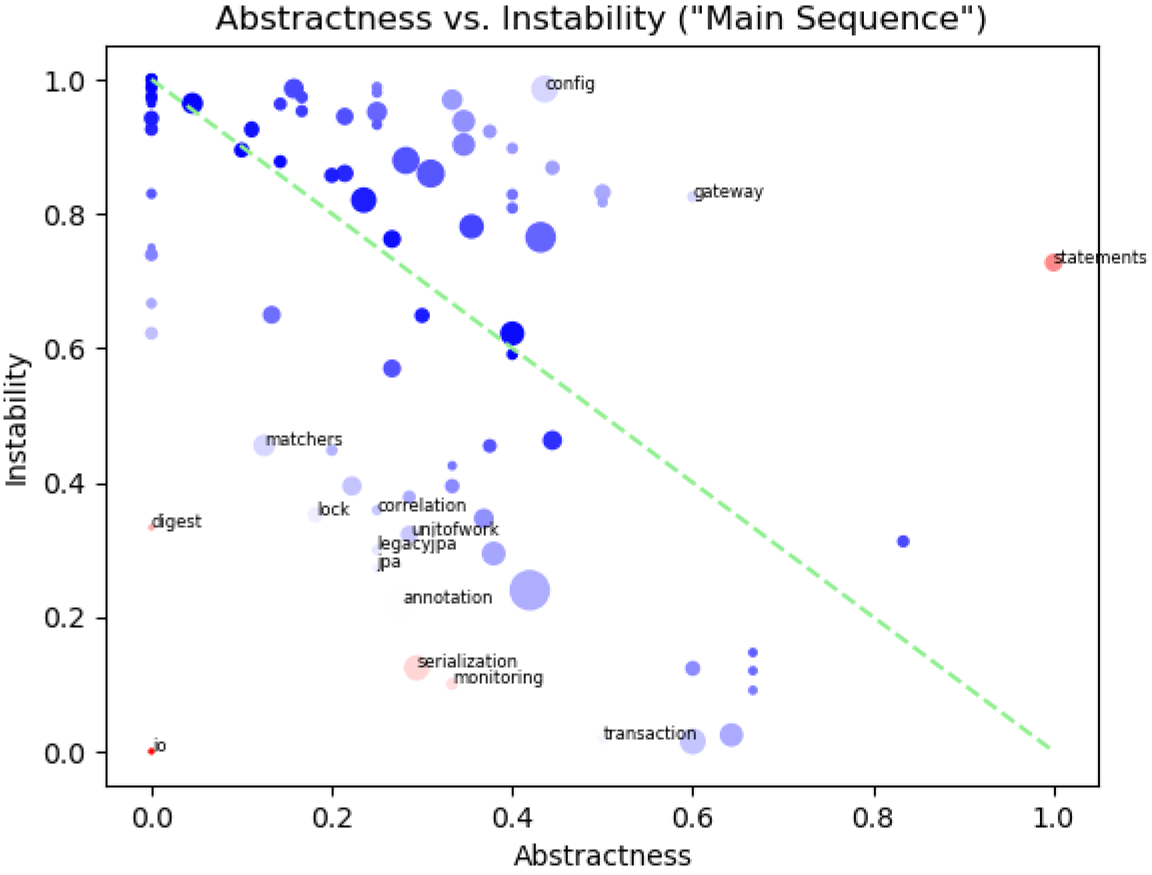
- Show the top 20 packages with the highest distance from the "main sequence"

	artifactName	fullQualifiedPackageName	packageName	distance	abstractness	instability	typesInPackage
0	axon-messaging-4.8.0	org.axonframework.common.io	io	1.000000	0.000000	0.000000	1
1	axon-eventsourcing-4.8.0	org.axonframework.eventsourcing.eventstore.jdb...	statements	0.727273	1.000000	0.727273	15
2	axon-messaging-4.8.0	org.axonframework.common.digest	digest	0.666667	0.000000	0.333333	1
3	axon-messaging-4.8.0	org.axonframework.serialization	serialization	0.581589	0.294118	0.124294	34
4	axon-messaging-4.8.0	org.axonframework.monitoring	monitoring	0.566188	0.333333	0.100478	6
5	axon-messaging-4.8.0	org.axonframework.messaging.annotation	annotation	0.500409	0.277778	0.221813	54
6	axon-messaging-4.8.0	org.axonframework.common.transaction	transaction	0.481013	0.500000	0.018987	4
7	axon-messaging-4.8.0	org.axonframework.common.jpa	jpa	0.477273	0.250000	0.272727	4
8	axon-messaging-4.8.0	org.axonframework.common.lock	lock	0.466069	0.181818	0.352113	11
9	axon-messaging-4.8.0	org.axonframework.common.legacyjpa	legacyjpa	0.450000	0.250000	0.300000	4
10	axon-messaging-4.8.0	org.axonframework.eventhandling.gateway	gateway	0.425397	0.600000	0.825397	5
11	axon-configuration-4.8.0	org.axonframework.config	config	0.421681	0.435897	0.985783	39
12	axon-test-4.8.0	org.axonframework.test.matchers	matchers	0.419643	0.125000	0.455357	24
13	axon-messaging-4.8.0	org.axonframework.messaging.unitofwork	unitofwork	0.391451	0.285714	0.322835	14
14	axon-messaging-4.8.0	org.axonframework.messaging.correlation	correlation	0.391026	0.250000	0.358974	4
15	axon-messaging-4.8.0	org.axonframework.messaging	messaging	0.385160	0.600000	0.014840	35
16	axon-messaging-4.8.0	org.axonframework.tracing	tracing	0.382956	0.222222	0.394822	18
17	axon-messaging-4.8.0	org.axonframework.serialization.xml	xml	0.377778	0.000000	0.622222	7
18	axon-test-4.8.0	org.axonframework.test	test	0.351724	0.200000	0.448276	5
19	axon-messaging-4.8.0	org.axonframework.eventhandling	eventhandling	0.340885	0.419355	0.239760	93

Abstractness vs. *Instability* Plot with "Main Sequence" line as reference

Figure 1

- Plot *Abstractness* vs. *Instability* of all packages
- Draw the "main sequence" as dashed green line
- Scale the packages by the number of types they contain
- Color the packages by their distance to the "main sequence" (blue=near, red=far)


Abstractness vs. Instability ("Main Sequence")
Instability
Abstractness
0.0
0.2
0.4
0.6
0.8
1.0
0.0
0.2
0.4
0.6
0.8
1.0
config
gateway
statements
matchers
lock
correlation
digest
unitofwork
legacyjpa
jpa
annotation
serialization
monitoring
transaction
io