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-4.7.5.jar	61	729
1	axon-modelling-4.7.5.jar	10	149
2	axon-disruptor-4.7.5.jar	1	22
3	axon-eventsourcing-4.7.5.jar	9	130
4	axon-configuration-4.7.5.jar	1	39
5	axon-test-4.7.5.jar	8	85

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	incomingDependent
0	org.axonframework.messaging	8507	33749	310	
1	org.axonframework.eventhandling	4409	27987	280	
2	org.axonframework.commandhandling	1539	7503	123	
3	org.axonframework.serialization	1078	5606	126	
4	org.axonframework.messaging.annotation	1050	5603	149	
5	org.axonframework.common	876	2059	313	
6	org.axonframework.common.transaction	276	1060	67	
7	org.axonframework.modelling.command	254	995	75	
8	org.axonframework.messaging.unitofwork	251	1384	81	
9	org.axonframework.modelling.saga	242	1482	57	
10	org.axonframework.eventsourcing.eventstore.jdbc	218	1404	26	
11	org.axonframework.monitoring	189	641	39	
12	org.axonframework.queryhandling	175	935	47	
13	org.axonframework.tracing	173	712	65	
14	org.axonframework.eventsourcing.eventstore	169	814	60	
15	org.axonframework.deadline	165	1367	33	
16	org.axonframework.messaging.deadletter	135	978	28	
17	org.axonframework.eventsourcing	134	657	41	
18	org.axonframework.config	115	1483	34	
19	org.axonframework.commandhandling.gateway	101	409	35	

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	outgoingDependent
0	org.axonframework.config	7942	212	84	
1	org.axonframework.test.aggregate	2223	92	34	
2	org.axonframework.eventhandling	1557	151	54	
3	org.axonframework.disruptor.commandhandling	1487	85	31	
4	org.axonframework.test.saga	1375	79	26	
5	org.axonframework.eventsourcing.eventstore.jdbc	1340	51	27	
6	org.axonframework.queryhandling	1108	78	28	
7	org.axonframework.eventhandling.pooled	1022	57	26	
8	org.axonframework.eventsourcing	976	91	31	
9	org.axonframework.modelling.command	827	91	33	
10	org.axonframework.modelling.command.inspection	781	73	28	
11	org.axonframework.commandhandling	642	70	28	
12	org.axonframework.commandhandling.distributed	603	67	23	
13	org.axonframework.eventsourcing.eventstore	603	64	25	
14	org.axonframework.deadline.quartz	481	38	18	
15	org.axonframework.commandhandling.gateway	447	58	11	
16	org.axonframework.modelling.saga	386	58	21	
17	org.axonframework.eventsourcing.eventstore.leg...	375	47	17	
18	org.axonframework.deadline.jobrunr	348	31	15	
19	org.axonframework.deadline	347	43	21	

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	instabilityArtifac
0	org.axonframework.messaging	messaging	0.015394	0.101449	0.189873	0.107143	0.1428
1	org.axonframework.common.transaction	transaction	0.021277	0.056338	0.000000	0.040000	0.2000
2	org.axonframework.common	common	0.025584	0.045732	0.000000	0.013333	0.1428
3	org.axonframework.monitoring	monitoring	0.100000	0.152174	0.333333	0.230769	0.2000
4	org.axonframework.eventhandling.scheduling	scheduling	0.111111	0.166667	0.000000	0.250000	0.2500
5	org.axonframework.common.annotation	annotation	0.120000	0.120000	0.000000	0.166667	0.2500
6	org.axonframework.lifecycle	lifecycle	0.138889	0.259259	0.000000	0.214286	0.2500
7	org.axonframework.serialization	serialization	0.140351	0.267442	0.318182	0.230769	0.2000
8	org.axonframework.common.stream	stream	0.147059	0.166667	0.000000	0.125000	0.2500
9	org.axonframework.messaging.annotation	annotation	0.222798	0.310185	0.419355	0.218750	0.1428
10	org.axonframework.eventhandling	eventhandling	0.260979	0.350348	0.509434	0.266667	0.1666
11	org.axonframework.common.jpa	jpa	0.272727	0.250000	1.000000	0.300000	0.2000
12	org.axonframework.commandhandling	commandhandling	0.294360	0.362694	0.608696	0.333333	0.1428
13	org.axonframework.common.legacyjpa	legacyjpa	0.300000	0.277778	1.000000	0.333333	0.2500
14	org.axonframework.serialization.upcasting	upcasting	0.312500	0.083333	0.000000	0.333333	0.5000
15	org.axonframework.messaging.unitofwork	unitofwork	0.328877	0.198020	0.583333	0.128205	0.1428
16	org.axonframework.common.lock	lock	0.352113	0.363636	0.500000	0.222222	0.2000
17	org.axonframework.messaging.correlation	correlation	0.358974	0.230769	0.400000	0.333333	0.3333
18	org.axonframework.eventhandling.tokenstore	tokenstore	0.378378	0.342105	0.571429	0.333333	0.3333
19	org.axonframework.common.property	property	0.394737	0.380952	1.000000	0.285714	0.3333

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.tracing.attributes	attributes	0.000000	0	6
5	org.axonframework.serialization.converters	converters	0.000000	0	5
6	org.axonframework.commandhandling.callbacks	callbacks	0.000000	0	4
7	org.axonframework.deadline.quartz	quartz	0.000000	0	4
8	org.axonframework.eventhandling.deadletter	deadletter	0.000000	0	4
9	org.axonframework.eventhandling.scheduling.java	java	0.000000	0	4
10	org.axonframework.eventhandling.tokenstore.jpa	jpa	0.000000	0	4
11	org.axonframework.deadline.jobrunr	jobrunr	0.000000	0	3
12	org.axonframework.eventhandling.scheduling.job...	jobrunr	0.000000	0	3
13	org.axonframework.util	util	0.000000	0	3
14	org.axonframework.modelling.saga.repository.le...	legacyjpa	0.000000	0	3
15	org.axonframework.eventhandling.tokenstore.inm...	inmemory	0.000000	0	2
16	org.axonframework.eventhandling.tokenstore.leg...	legacyjpa	0.000000	0	2
17	org.axonframework.messaging.interceptors.legac...	legacyvalidation	0.000000	0	2
18	org.axonframework.modelling.command.legacyjpa	legacyjpa	0.000000	0	2
19	org.axonframework.modelling.saga.repository.in...	inmemory	0.000000	0	2
20	org.axonframework.eventsourcing.eventstore.inm...	inmemory	0.000000	0	2
21	org.axonframework.test.server	server	0.000000	0	2
22	org.axonframework.common.digest	digest	0.000000	0	1
23	org.axonframework.common.io	io	0.000000	0	1
24	org.axonframework.eventhandling.interceptors	interceptors	0.000000	0	1
25	org.axonframework.disruptor.commandhandling	commandhandling	0.045455	1	22
26	org.axonframework.eventhandling.deadletter.jpa	jpa	0.111111	1	9
27	org.axonframework.eventhandling.tokenstore.jdbc	jdbc	0.111111	1	9
28	org.axonframework.modelling.saga.repository.jdbc	jdbc	0.111111	1	9
29	org.axonframework.test.matchers	matchers	0.125000	3	24

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

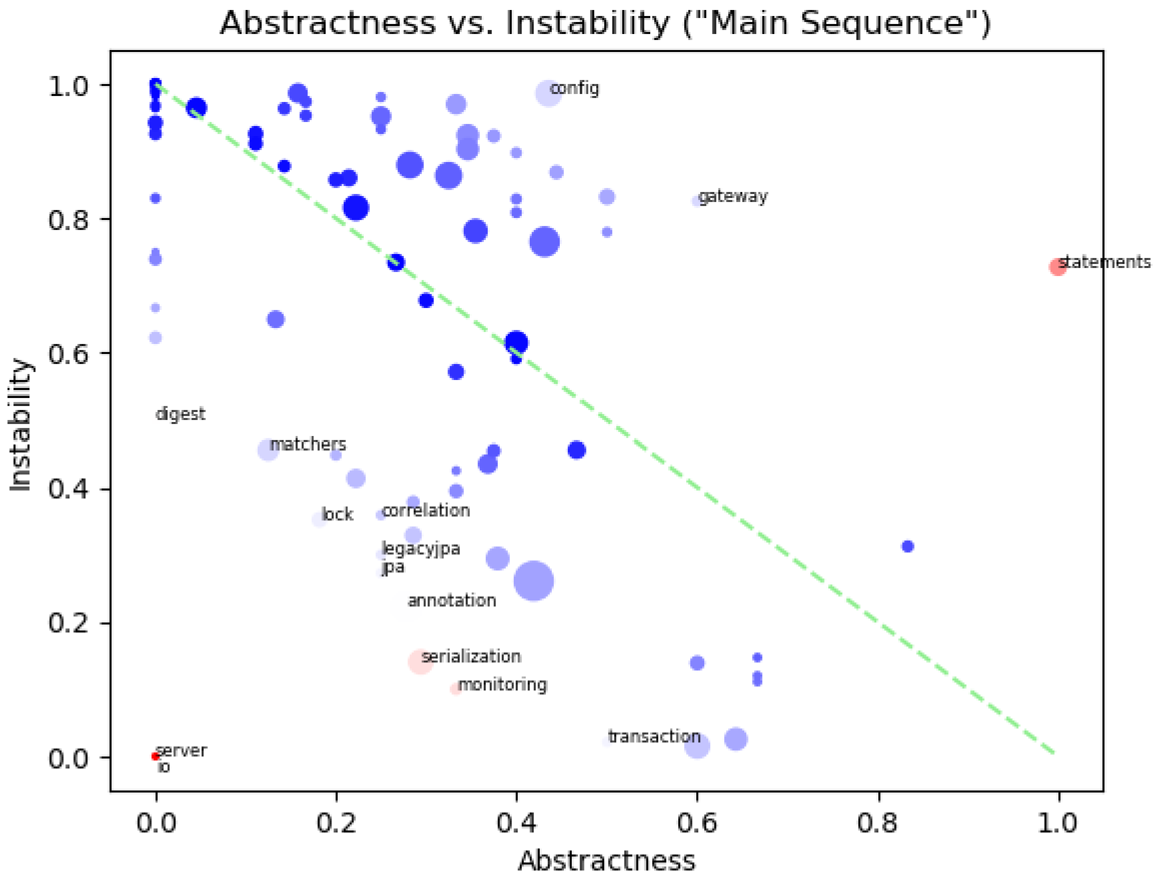
- Show the top 20 packages with the highest distance from the "main sequence"

	artifactName	fullQualifiedPackageName	packageName	distance	abstractness	instability	typesInPackage
0	axon-test-4.7.5	org.axonframework.test.server	server	1.000000	0.000000	0.000000	2
1	axon-messaging-4.7.5	org.axonframework.common.io	io	1.000000	0.000000	0.000000	1
2	axon-eventsourcing-4.7.5	org.axonframework.eventsourcing.eventstore.jdb...	statements	0.727273	1.000000	0.727273	15
3	axon-messaging-4.7.5	org.axonframework.monitoring	monitoring	0.566667	0.333333	0.100000	6
4	axon-messaging-4.7.5	org.axonframework.serialization	serialization	0.565531	0.294118	0.140351	34
5	axon-messaging-4.7.5	org.axonframework.common.digest	digest	0.500000	0.000000	0.500000	1
6	axon-messaging-4.7.5	org.axonframework.messaging.annotation	annotation	0.499424	0.277778	0.222798	54
7	axon-messaging-4.7.5	org.axonframework.common.transaction	transaction	0.478723	0.500000	0.021277	4
8	axon-messaging-4.7.5	org.axonframework.common.jpa	jpa	0.477273	0.250000	0.272727	4
9	axon-messaging-4.7.5	org.axonframework.common.lock	lock	0.466069	0.181818	0.352113	11
10	axon-messaging-4.7.5	org.axonframework.common.legacyjpa	legacyjpa	0.450000	0.250000	0.300000	4
11	axon-messaging-4.7.5	org.axonframework.eventhandling.gateway	gateway	0.425397	0.600000	0.825397	5
12	axon-configuration-4.7.5	org.axonframework.config	config	0.421624	0.435897	0.985727	39
13	axon-test-4.7.5	org.axonframework.test.matchers	matchers	0.419643	0.125000	0.455357	24
14	axon-messaging-4.7.5	org.axonframework.messaging.correlation	correlation	0.391026	0.250000	0.358974	4
15	axon-messaging-4.7.5	org.axonframework.messaging.unitofwork	unitofwork	0.385409	0.285714	0.328877	14
16	axon-messaging-4.7.5	org.axonframework.messaging	messaging	0.384606	0.600000	0.015394	35
17	axon-messaging-4.7.5	org.axonframework.serialization.xml	xml	0.377778	0.000000	0.622222	7
18	axon-messaging-4.7.5	org.axonframework.tracing	tracing	0.364218	0.222222	0.413559	18
19	axon-test-4.7.5	org.axonframework.test	test	0.351724	0.200000	0.448276	5

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
digest
matchers
lock
correlation
legacyjpa
jpa
annotation
serialization
monitoring
transaction
server
io