

CRSRoot :

$File_{affectedRoot} = \{f\}, f \in File_{CRS} \wedge fanin(f) \geq 4$
 $\wedge fanout(f) \geq 4 \wedge Cochange(f) \geq 2$

UIFRoot :

$File_{affectedRoot} = \{f\}, f \in File_{UIF} \wedge StructImpact(f) \geq 1\%$
 $\wedge HistoryImpact(f) \geq 10 \wedge Cochange(f) \geq 2$

MVGRoot :

$File_{affectedRoot} = \{f\}, f \in File_{MVG} \wedge Cochange(f) \geq 2$

PKCRoot :

$File_{affectedRoot} = \{f_1, f_2, f_i, f_j\}, \{f_1, f_2, f_i, f_j\} \in File_{PKC}$
 $|depend(f_1, f_j) \wedge depend(f_2, f_i)|$
 $|\{f_1, f_i\} \in P_a \wedge \{f_2, f_j\} \in P_b|$

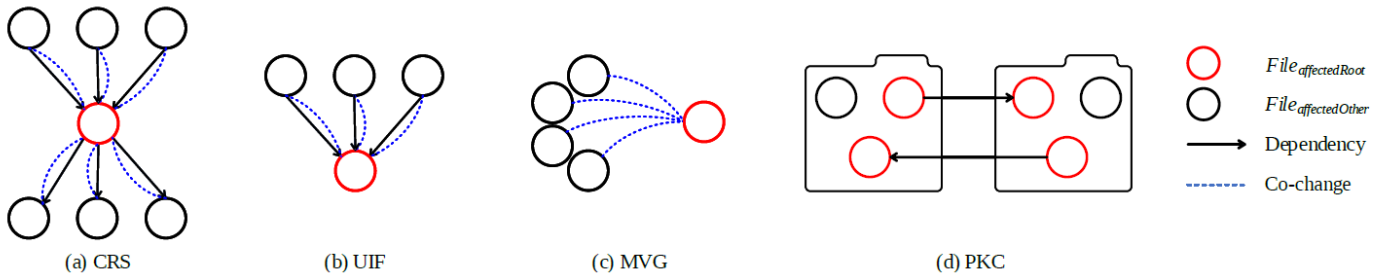


Fig. 2. Root files ($File_{affectedRoot}$) and non-root files ($File_{affectedOther}$) in each architecture anti-pattern

Description:

As we can see, in Figure 2(a), the file with both high fan-in and high fanout is the CRS root file. For UIF in Figure 2(b), the file with a large number of dependents is the UIF root file. In MVG shown in Figure 2(c), the file frequently co-changed with others structurally un-linked files is MVG root file. For PKC in Figure 2(d), the four files causing the cycle are PKC root files.

E.g., CRS root:

		1	2	3	4	5	6	7	8
other	airflow.utils.helpers_py	7	C2	C2,I	C...	C3	C5	C3	C4
	airflow.utils._init_py	C2	2	C2	C2	C	C2	C2	C2
	airflow._init_py	C2	C2,I	3	C...	C10	C7,C,I	C2	C6
	airflow.models_py	I...	C2,I	C...	4	C10	C11,I	C3	C10
root	airflow.plugins.manager_py	C3	C,I	C...	C10	5	C7	C	C2
	airflow.operators._init_py	C...	C2,I	C7,I	C...	C...	6	C2	C9
other	airflow.example_dags.example_short_circuit_operator_py	C3,I	C2,I	C2,I	I...	C	C2,I	7	C3
	airflow.operators.presto_check_operator_py	C4	C2,I	C6,I	C10	C2	C9,I	C3	8
	airflow.example_dags.example_branch_python_dop_operator_3_py	C3	C	C,I	I...	C2	C2,I	C4	C2
	airflow.example_dags.example_python_operator_py	C3	C	C,I	I...	C	C3,I	C5	C3
	airflow.example_dags.example_trigger_controller_dag_py	C3	C2	C2,I	C4	C	C2,I	C5	C3
	airflow.example_dags.example_bash_operator_py	C2	C	C2,I	I...	C	C2,I	C5	C4
	airflow.example_dags.example_trigger_target_dag_py	C3	C2	C2,I	I...	C	C2,I	C5	C3
	airflow.contrib.example_dags.example_twitter_dag_py	C2	C	C2,I	C...	C2	C2,I	C3	C3
	airflow.example_dags.example_branch_operator_py	C2	C	C,I	I...	C	C2,I	C5	C2
	airflow.example_dags.example_xcom_py	C3	C	C,I	C...	C2	C2,I	C5	C2
	airflow.example_dags.example_passing_params_via_test_command_py	C3	C	C,I	C...	C2	C2,I	C4	C2
	airflow.example_dags.example_http_operator_py	C2	C	C2,I	C...	C3	C4,I	C5	C3
	airflow.example_dags.example_latest_only_py	C	C	C,I	C...	C	C2,I	C3	C
	airflow.www.views_py	I...	C2,I	C...	S...	C4	C5,I	C3	C3
	airflow.contrib.operators.bigquery_check_operator_py	C4	C2,I	C2,I	C2	C	C2,I	C3	C4
	airflow.example_dags.example_latest_only_with_trigger_py	C	C,I	C,I	C...	C	C2,I	C3	C