Case Management

Case management refers to the retrieval of some statistics related to variants, cases and events in the considered log.

In this page, we wish to provide some insights on how to retrieve statistics about variants, cases and events on logs and Pandas dataframes.

If not differently specified, the same methods apply for both logs and Pandas dataframes.

Loading a log or a Pandas dataframe

The following instructions could be used to read the **receipt.xes** log:

```
import os
from pm4py.objects.log.importer.xes import
factory as xes_importer

log =
xes_importer.import_log(os.path.join("tests","inp
```

And the following ones could be used to read a Pandas dataframe from the receipt.csv file:

```
import os
from pm4py.objects.log.adapters.pandas import
csv_import_adapter

df =
csv_import_adapter.import_dataframe_from_path(os.
 "input_data", "receipt.csv"))
```

Calculating statistics about variants

The following code returns the top **5** variants of the log (parameter **max_variants_to_return**).

```
from pm4py.statistics.traces.log import
case_statistics

var_with_count =
case_statistics.get_variant_statistics(log,
parameters={"max_variants_to_return": 5})
```

Doing **print(var_with_count)** on the receipt log produces the following result:

```
[{'variant': 'Confirmation of receipt,T02 Check
confirmation of receipt,T04 Determine
confirmation of receipt,T05 Print and send
confirmation of receipt,T06 Determine necessity
```

of stop advice,T10 Determine necessity to stop indication', 'count': 713}, {'variant': 'Confirmation of receipt,T06 Determine necessity of stop advice,T10 Determine necessity to stop indication,T02 Check confirmation of receipt,T04 Determine confirmation of receipt,T05 Print and send confirmation of receipt', 'count': 123}, {'variant': 'Confirmation of receipt', 'count': 116}, {'variant': 'Confirmation of receipt,T02 Check confirmation of receipt,T06 Determine necessity of stop advice,T10 Determine necessity to stop indication,T04 Determine confirmation of receipt,T05 Print and send confirmation of receipt', 'count': 115}, {'variant': 'Confirmation of receipt,T02 Check confirmation of receipt,T04 Determine confirmation of receipt,T06 Determine necessity of stop advice,T10 Determine necessity to stop indication,T05 Print and send confirmation of receipt', 'count': 75}]

Calculating statistics about cases

Logs

The following code retrieves the case statistics on the log object, getting the first **5** cases ordered by start timestamp:

```
from pm4py.statistics.traces.log import
case_statistics
```

```
cases =
case_statistics.get_cases_description(log,
parameters={"sort_by_index": 1, "max_ret_cases":
5})
```

The maximum number of cases is expressed by the **max_ret_cases** parameter.

The following values are admitted for the **sort_by_index** parameter:

- **0** -> sort by case ID
- **1** -> sort by start time (of cases)
- **2** -> sort by end time (of cases)
- **3** -> sort on difference

The parameter **sort_ascending** is a boolean value deciding whether the sort has to been done in an ascending or descending fashion.

Pandas dataframes

The following code retrieves the case statistics on the Pandas dataframe object, getting the first **5** cases ordered by start timestamp:

```
from pm4py.statistics.traces.pandas import
case_statistics

cases =
case_statistics.get_cases_description(df,
```

```
parameters={"sort_by_column": "startTime",
"max_ret_cases": 5})
```

The maximum number of cases is expressed by the **max_ret_cases** parameter.

The following values are admitted for the sort_by_column parameter:

- **startTime** -> orders the cases by their start time.
- **completeTime** -> orders the cases by their completion time.
- **caseDuration** -> orders the cases based on their case duration.

The parameter **sort_ascending** is a boolean value deciding whether the sort has to been done in an ascending or descending fashion.

In both cases, the output is the same:

```
{'case-891': {'startTime': 1286004039,
'endTime': 1289565644, 'caseDuration':
3561605.0}, 'case-3756': {'startTime':
1286260368, 'endTime': 1286284570,
'caseDuration': 24201.0}, 'case-3766':
{'startTime': 1286276518, 'endTime': 1286457415,
'caseDuration': 180897.0}, 'case-3818':
{'startTime': 1287386667, 'endTime': 1288000321,
'caseDuration': 613654.0}, 'case-416':
{'startTime': 1287572218, 'endTime': 1289221662,
'caseDuration': 1649444.0}}
```

Getting events

The following code retrieves the events belonging to a particular case (in the particular example, case-891).

```
from pm4py.statistics.traces.log import
case_statistics

events = case_statistics.get_events(log,
case_id='case-891')
```

Obtaining the following output:

```
[{'org:group': 'Group 1', 'concept:instance':
'task-4', 'org:resource': 'Resource26',
'concept:name': 'Confirmation of receipt',
'time:timestamp': datetime.datetime(2010, 10, 2,
9, 20, 39, 266000,
tzinfo=datetime.timezone(datetime.timedelta(0,
7200))), 'lifecycle:transition': 'complete'},
{'org:group': 'Group 4', 'concept:instance':
'task-5', 'org:resource': 'Resource26',
'concept:name': 'T02 Check confirmation of
receipt', 'time:timestamp':
datetime.datetime(2010, 10, 2, 9, 21, 26,
588000,
tzinfo=datetime.timezone(datetime.timedelta(0,
7200))), 'lifecycle:transition': 'complete'},
{'org:group': 'Group 1', 'concept:instance':
'task-7', 'org:resource': 'Resource26',
```

'concept:name': 'T03 Adjust confirmation of receipt', 'time:timestamp': datetime.datetime(2010, 10, 2, 9, 31, 12, 836000, tzinfo=datetime.timezone(datetime.timedelta(0, 7200))), 'lifecycle:transition': 'complete'}, {'org:group': 'Group 4', 'concept:instance': 'task-8', 'org:resource': 'Resource26', 'concept:name': 'T02 Check confirmation of receipt', 'time:timestamp': datetime.datetime(2010, 10, 2, 9, 31, 40, 160000, tzinfo=datetime.timezone(datetime.timedelta(0, 7200))), 'lifecycle:transition': 'complete'}, {'org:group': 'Group 1', 'concept:instance': 'task-9', 'org:resource': 'Resource26', 'concept:name': 'T03 Adjust confirmation of receipt', 'time:timestamp': datetime.datetime(2010, 10, 2, 9, 32, 1, 401000, tzinfo=datetime.timezone(datetime.timedelta(0, 7200))), 'lifecycle:transition': 'complete'}, {'org:group': 'Group 4', 'concept:instance': 'task-10', 'org:resource': 'Resource21', 'concept:name': 'T02 Check confirmation of receipt', 'time:timestamp': datetime.datetime(2010, 11, 9, 14, 13, 52, 563000, tzinfo=datetime.timezone(datetime.timedelta(0, 3600))), 'lifecycle:transition': 'complete'}, {'org:group': 'Group 3', 'concept:instance': 'task-1275', 'org:resource': 'Resource21', 'concept:name': 'T04 Determine confirmation of receipt', 'time:timestamp': datetime.datetime(2010, 11, 9, 14, 14, 10,

16000,
tzinfo=datetime.timezone(datetime.timedelta(0,
3600))), 'lifecycle:transition': 'complete'},
{'org:group': 'Group 2', 'concept:instance':
'task-1276', 'org:resource': 'Resource21',
'concept:name': 'T05 Print and send confirmation
of receipt', 'time:timestamp':
datetime.datetime(2010, 11, 9, 14, 14, 22,
628000,
tzinfo=datetime.timezone(datetime.timedelta(0,
3600))), 'lifecycle:transition': 'complete'},
{'org:group': 'Group 1', 'concept:instance':
'task-6', 'org:resource': 'Resource21',
'concept:name': 'T06 Determine necessity of stop
advice', 'time:timestamp':
datetime.datetime(2010, 11, 9, 14, 14, 37,
300000,
tzinfo=datetime.timezone(datetime.timedelta(0,
3600))), 'lifecycle:transition': 'complete'},
{'org:group': 'Group 13', 'concept:instance':
'task-1277', 'org:resource': 'Resource21',
'concept:name': 'T07-1 Draft intern advice
aspect 1', 'time:timestamp':
datetime.datetime(2010, 11, 9, 14, 17, 36,
97000,
tzinfo=datetime.timezone(datetime.timedelta(0,
3600))), 'lifecycle:transition': 'complete'},
{'org:group': 'Group 1', 'concept:instance':
'task-1278', 'org:resource': 'Resource26',
'concept:name': 'T06 Determine necessity of stop
advice', 'time:timestamp':
datetime.datetime(2010, 11, 10, 7, 52, 33,
800000,
tzinfo=datetime.timezone(datetime.timedelta(0,

3600))), 'lifecycle:transition': 'complete'},
{'org:group': 'Group 13', 'concept:instance':
'task-1335', 'org:resource': 'Resource26',
'concept:name': 'T07-1 Draft intern advice
aspect 1', 'time:timestamp':
datetime.datetime(2010, 11, 10, 7, 52, 51,
41000,
tzinfo=datetime.timezone(datetime.timedelta(0,
3600))), 'lifecycle:transition': 'complete'},
{'org:group': 'Group 1', 'concept:instance':
'task-1336', 'org:resource': 'Resource26',
'concept:name': 'T06 Determine necessity of stop
advice', 'time:timestamp':
datetime.datetime(2010, 11, 10, 7, 53, 8, 65000,
tzinfo=datetime.timezone(datetime.timedelta(0,
3600))), 'lifecycle:transition': 'complete'},
{'org:group': 'Group 1', 'concept:instance':
'task-1337', 'org:resource': 'admin1',
'concept:name': 'T10 Determine necessity to stop
indication', 'time:timestamp':
datetime.datetime(2010, 11, 10, 9, 38, 45,
959000,
tzinfo=datetime.timezone(datetime.timedelta(0,
3600))), 'lifecycle:transition': 'complete'},
{'org:group': 'Group 1', 'concept:instance':
'task-1338', 'org:resource': 'admin1',
'concept:name': 'T11 Create document X request
unlicensed', 'time:timestamp':
datetime.datetime(2010, 11, 10, 9, 38, 57,
202000,
tzinfo=datetime.timezone(datetime.timedelta(0,
3600))), 'lifecycle:transition': 'complete'},
{'org:group': 'Group 4', 'concept:instance':
'task-1339', 'org:resource': 'admin1',

'concept:name': 'T12 Check document X request unlicensed', 'time:timestamp': datetime.datetime(2010, 11, 10, 9, 39, 7, 674000, tzinfo=datetime.timezone(datetime.timedelta(0, 3600))), 'lifecycle:transition': 'complete'}, {'org:group': 'Group 3', 'concept:instance': 'task-1340', 'org:resource': 'admin1', 'concept:name': 'T14 Determine document X request unlicensed', 'time:timestamp': datetime.datetime(2010, 11, 10, 9, 39, 21, 494000, tzinfo=datetime.timezone(datetime.timedelta(0, 3600))), 'lifecycle:transition': 'complete'}, {'org:group': 'Group 2', 'concept:instance': 'task-1341', 'org:resource': 'Resource26', 'concept:name': 'T15 Print document X request unlicensed', 'time:timestamp': datetime.datetime(2010, 11, 12, 13, 40, 44, 291000, tzinfo=datetime.timezone(datetime.timedelta(0, 3600))), 'lifecycle:transition': 'complete'}]

.