Log-based Event Gaps Analysis

S. Suriadi, C. Ouyang, W.M.P. van der Aalst, and A.H.M. ter Hofstede

Information Systems School
Queensland University of Technology
Brisbane, Australia

s.suriadi@qut.edu.au

June 13, 2013

About Myself

- PhD in Information Security (till late 2010) - QUT
  - designing privacy-enhancing cryptographic protocols
  - verifying privacy properties with CPN and state space analysis
- Research Assistant, then Postdoc (2007 - 2011) - QUT
  - Web services security
  - Mitigating denial of service attacks
  - Cryptographic client puzzles
- Postdoc (2011 - now) - QUT
  - BPM Group
  - Process mining - ARC Discovery Project - Risk Aware BPM grant (2011-2013)

My Research Interests

- Process mining
  - Event gap analysis (this talk)
    - Working paper
  - Visualization of log
    - Work with Massimiliano - journal submission soon?
  - A framework for 'Root cause analysis' with enriched log
    - Approach formalization - BPI Workshop 2012
  - Quality and reliability of process mining results
    - Inspired by application of process mining (health and finance) - CAiSE 2013
    - Data quality - DECRA proposal
- Process mining for information/network security
  - Similarities with traffic flow analysis
  - Opportunities?

Event Gap Analysis
Problem

- Background: originally interested in measuring resources' workload
  - BPI Workshop 2012 paper: assume start and complete timestamps
  - Realistic?
- Event log may contain limited information (e.g. only 'complete' or 'start' timestamp)
- May hinder certain type of analyses, e.g. performance analysis
  - how to obtain waiting time of cases/resources?

Question

How can we still obtain a good estimation of the performance of cases and resources from the log with limited event transaction lifecycle information?

Event Gap Analysis

Contribution

- An approach to support the analysis of case and resource performance based on the event logs with minimum (timestamp) information.
  - Exploits the use of time duration (or gap) between any two adjacent events referring to the same case/resource
  - May yield a better estimation of case and resource performance
  - Copes with event logs with limited information (e.g. only one event transaction lifecycle timestamp)
  - Implemented in ProM

Related Work

- Basic Performance Analysis plug-in
  - Estimate the working time of tasks
  - Can work with only one event transaction type.
  - Task-view only, not considering other dimensions
    - Our approach supports other views (case, resource) and exploits relationships between those views
- Resource availability analysis (Nakatumba's work)

Approach

- Use 'event gaps' as the basic building blocks for performance analysis
  - $\mathscr{E}$ is the set of events
    - Definition of *event* - see van der Aalst (2011).
  - An event log $\mathcal{L} \subseteq \mathscr{E}$ is a set of events.
  - Ordered by timestamps, i.e., $e_1 < e_2$ if and only if $time(e_1) < time(e_2)$ and $e_1 \leq e_2$ if and only if $time(e_1) \leq time(e_2)$.

Gap

Let $e_1, e_2 \in \mathscr{E}$ with $e_1 < e_2 : (e_1, e_2)$ is a gap.
$\mathscr{G} = \{(e_1, e_2) \in \mathscr{E} \times \mathscr{E} \mid e_1 < e_2\}$ is the set of all possible gaps.
$dur(e_1, e_2) = time(e_2) - time(e_1)$ is the duration of gap (e_1, e_2).

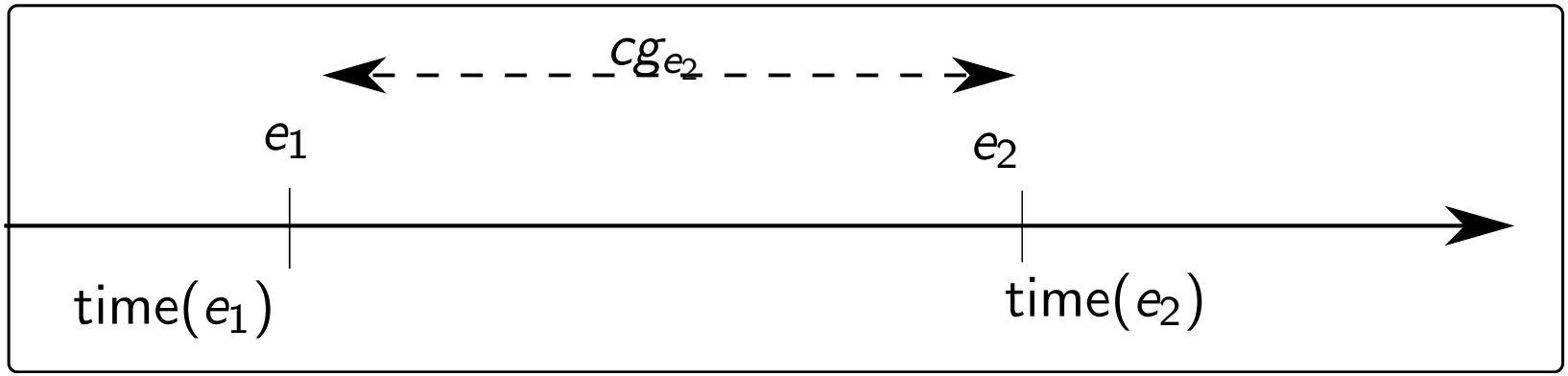
Approach

- Given a log, the existence of various types of gaps can be identified and calculated,
- Types of gaps:
  - Case gap
  - Resource gap
  - Minimum gap
  - Case-Resource gap
  - Resource-Case gap
- These gaps form the *basic* building blocks for subsequent (performance) analyses

Case Gap

Case Gap

$$CG(\mathcal{L}) = \{(e_1, e_2) \in \mathcal{G} \cap (\mathcal{L} \times \mathcal{L}) \mid case(e_1) = case(e_2) \wedge (\nexists_{e_3 \in \mathcal{L}} e_1 < e_3 < e_2 \wedge case(e_3) = case(e_2))\}.$$

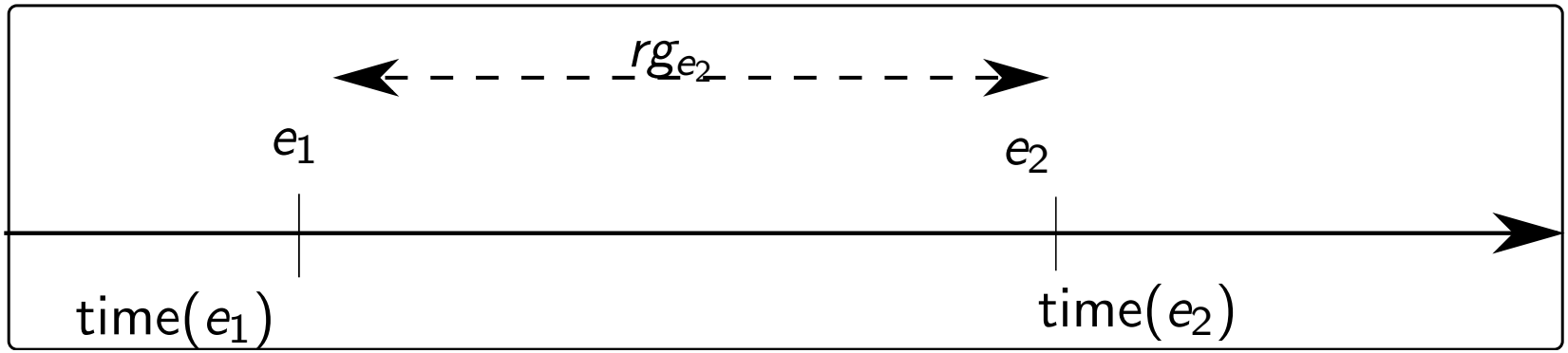


We use the notation cg_{e_j} to denote $(e_i, e_j) \in CG(\mathcal{L})$.

Resource Gap

Resource Gap

$RG(\mathcal{L}) = \{(e_1, e_2) \in \mathcal{G} \cap (\mathcal{L} \times \mathcal{L}) \mid res(e_1) = res(e_2) \wedge (\nexists_{e_3 \in \mathcal{L}} e_1 < e_3 < e_2 \wedge res(e_3) = res(e_2))\}$.

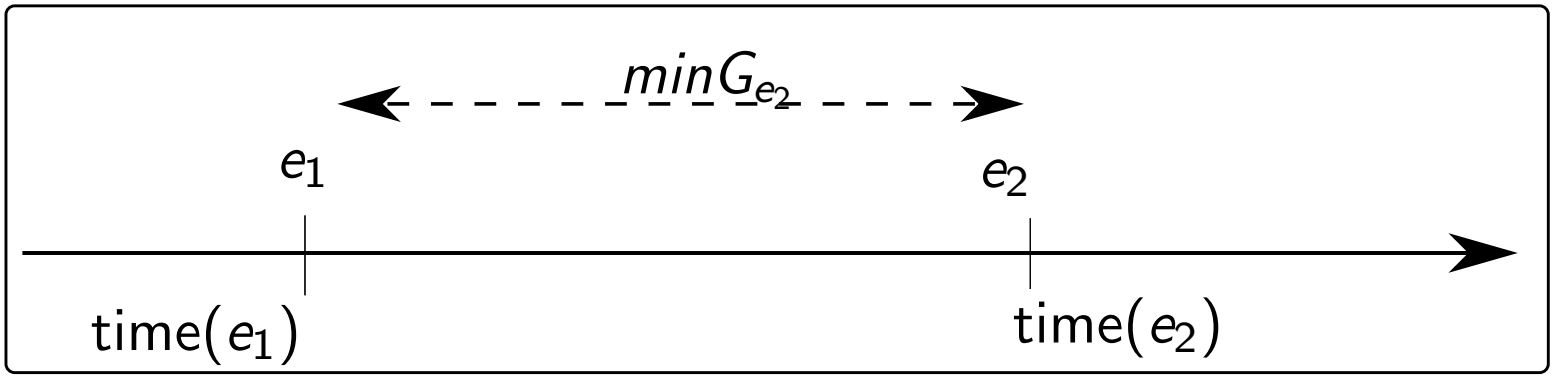


We use the notation rg_{e_j} to denote $(e_i, e_j) \in RG(\mathcal{L})$.

Minimum Gap

Minimum Gap

$MinG(\mathcal{L}) = \{(e_1, e_2) \in \mathcal{G} \cap (\mathcal{L} \times \mathcal{L}) \mid (res(e_1) = res(e_2) \vee case(e_1) = case(e_2)) \wedge (\nexists_{e_3 \in \mathcal{L}} e_1 < e_3 < e_2 \wedge (res(e_3) = res(e_2) \vee case(e_3) = case(e_2)))\}$.

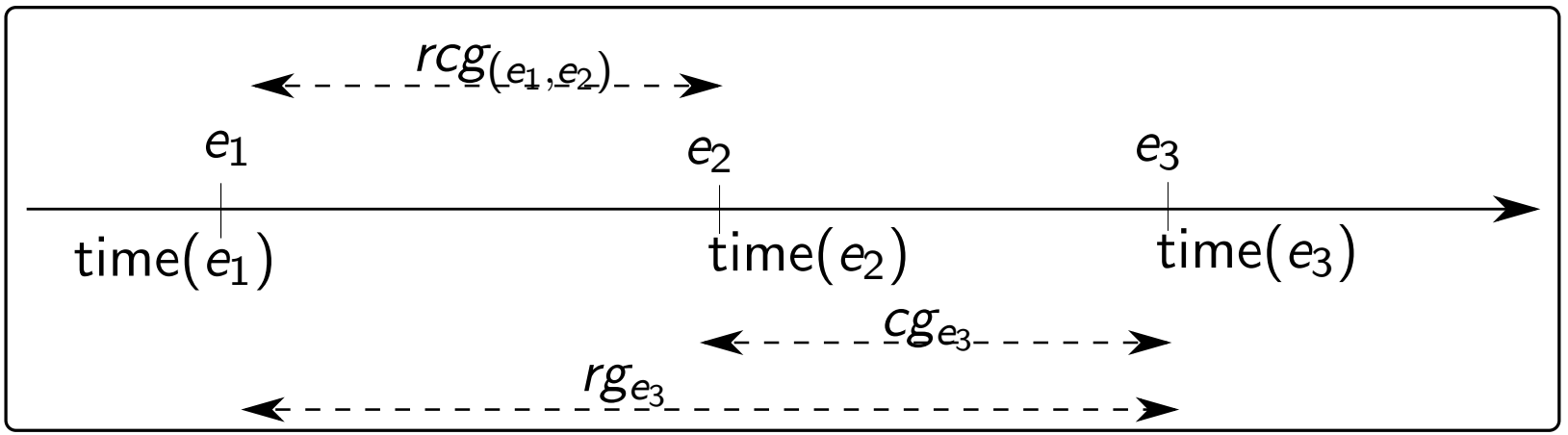


We use the notation mg_{e_j} to denote $(e_i, e_j) \in MinG(\mathcal{L})$. A *minimum gap* is the shorter of either a case gap or resource gap.

Resource-Case Gap

Resource-Case Gap

$RCG(\mathcal{L}) = \{(e_1, e_2) \in \mathcal{G} \cap (\mathcal{L} \times \mathcal{L}) \mid \exists_{e_3 \in \mathcal{L}} : e_2 < e_3 \wedge (e_1, e_3) \in RG(\mathcal{L}) \wedge (e_2, e_3) \in CG(\mathcal{L})\}$.

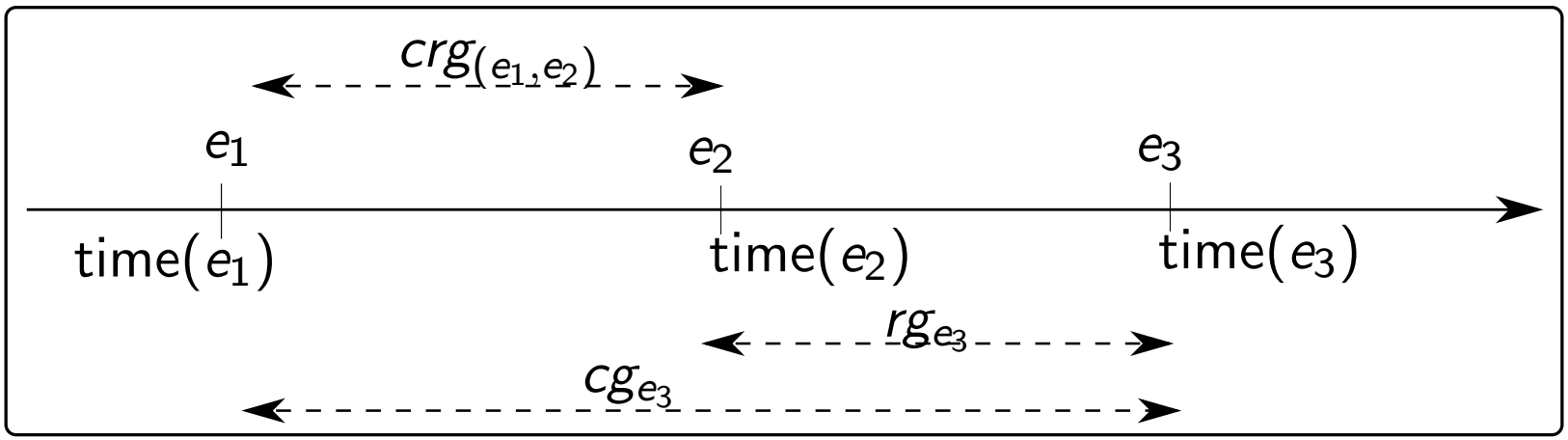


We use the notation $rcg_{(e_i,e_j)}$ to denote $(e_i, e_j) \in RCG(\mathcal{L})$. Exists when resource gap is longer than the corresponding case gap.

Case-Resource Gap

Case-Resource Gap

$CRG(\mathcal{L}) = \{(e_1, e_2) \in \mathcal{G} \cap (\mathcal{L} \times \mathcal{L}) \mid \exists_{e_3 \in \mathcal{L}} : e_2 < e_3 \wedge (e_1, e_3) \in CG(\mathcal{L}) \wedge (e_2, e_3) \in RG(\mathcal{L})\}$.



We use the notation $crg_{(e_i,e_j)}$ to denote $(e_i, e_j) \in CRG(\mathcal{L})$.Exists when case gap is longer than the corresponding resource gap.

Interpretation of Gaps

Meaning of Gaps

- Assume:
  - Any two events considered for gaps calculation correspond to work items not executed in parallel.
  - Only consider 'complete' event type.
- Case gap (cg_{e_j}): expected working time of `activity`(e_j) in a case?
- Resource gap (rg_{e_j}): expected working time of `activity`(e_j) by a resource?
- Minimum gap (mg_{e_j}): the actual working time of `activity`(e_j)?
  - both resource and case are available
- Case-resource gap ($crg_{(e_i,e_j)}$): case waiting time for `activity`(e_j)
- Resource-case gap ($rcg_{(e_i,e_j)}$): resource waiting time for `activity`(e_j)

Group By

- Event gaps represent basic performance data unit –> 'isolated', fine-grained
- 'Group by' allows an aggregation of gaps based on certain characteristics, e.g.
  - resource,
  - activity,
  - team,
  - null, etc.
- By grouping calculated event gaps using different characteristics, we can derive more meaningful results.

Group By

caseID	activity	timestamp	resource	caseResourceGap	minGap	resourceGap	resourceCaseGap	caseGap
341	Repair (Complex)	1970-01-09 16:52:00	SolverC1		720000			720000
387	Repair (Complex)	1970-01-09 18:01:00	SolverC1	0	300000	4140000	3840000	300000
369	Repair (Complex)	1970-01-09 22:24:00	SolverC1	0	2040000	15780000	13740000	2040000
412	Repair (Complex)	1970-01-10 00:44:00	SolverC1		1500000			1500000
379	Repair (Complex)	1970-01-10 01:35:00	SolverC1	0	180000	3060000	2880000	180000
...	...	.....	....	.....	....	.....	....	...
354	Test Repair	1970-01-09 16:22:00	Tester1		600000			600000
351	Analyze Defect	1970-01-09 16:39:00	Tester1	0	420000	1020000	600000	420000
341	Test Repair	1970-01-09 16:58:00	Tester1	0	360000	1140000	780000	360000
387	Analyze Defect	1970-01-09 17:32:00	Tester1	0	360000	2040000	1680000	360000
337	Test Repair	1970-01-09 18:06:00	Tester1	0	480000	2040000	1560000	480000
..	.....	.....	...	.....	....	.....	.....	.....

- For each gap type, and for each 'group by' on a specific characteristic, we can do three types of analysis:
  - Metrics
  - Decision Tree
  - Gaps Evolution over Time

Group By

caseID	activity	timestamp	resource	caseResourceGap	minGap	resourceGap	resourceCaseGap	caseGap
341	Repair (Complex)	1970-01-09 16:52:00	SolverC1		720000			720000
387	Repair (Complex)	1970-01-09 18:01:00	SolverC1	0	300000	4140000	3840000	300000
369	Repair (Complex)	1970-01-09 22:24:00	SolverC1	0	2040000	15780000	13740000	2040000
412	Repair (Complex)	1970-01-10 00:44:00	SolverC1		1500000			1500000
379	Repair (Complex)	1970-01-10 01:35:00	SolverC1	0	180000	3060000	2880000	180000
...	...	.....	....	.....	....	.....	....	...

354	Test Repair	1970-01-09 16:22:00	Tester1		600000			600000
351	Analyze Defect	1970-01-09 16:39:00	Tester1	0	420000	1020000	600000	420000
341	Test Repair	1970-01-09 16:58:00	Tester1	0	360000	1140000	780000	360000
387	Analyze Defect	1970-01-09 17:32:00	Tester1	0	360000	2040000	1680000	360000
337	Test Repair	1970-01-09 18:06:00	Tester1	0	480000	2040000	1560000	480000
..	.....	.....	...	.....	....	.....	.....	.....

- For each gap type, and for each 'group by' on a specific characteristic, we can do three types of analysis:
  - Metrics
  - Decision Tree
  - Gaps Evolution over Time

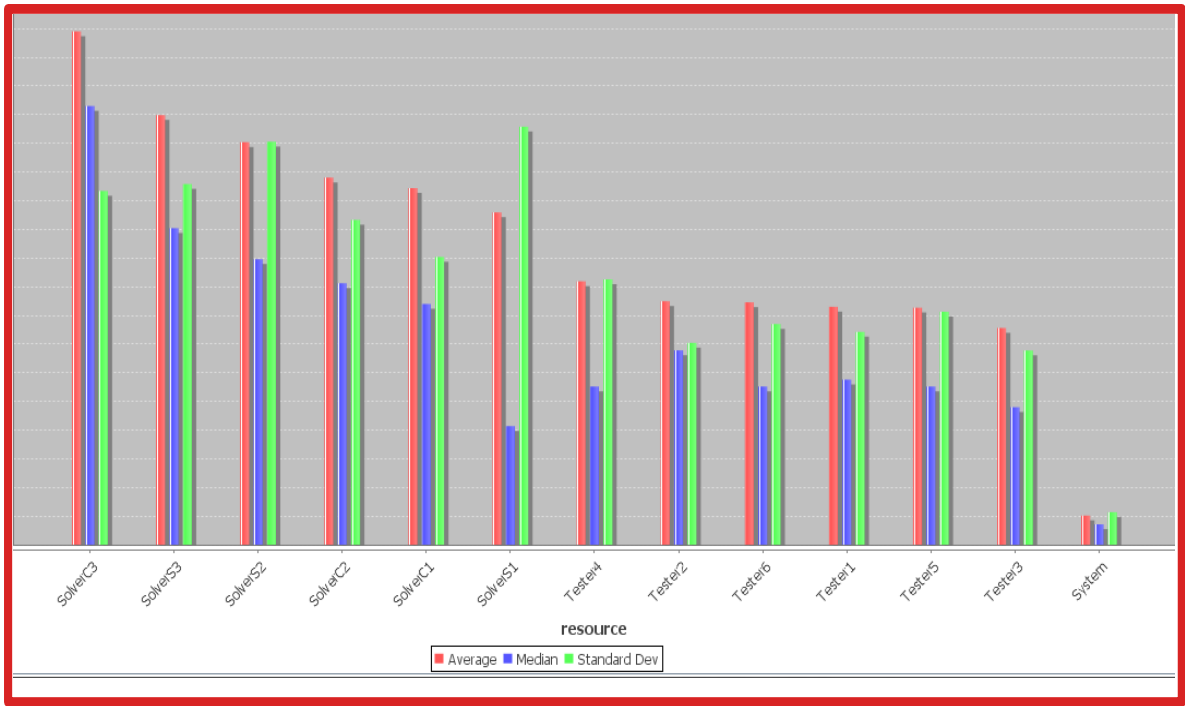
Group By

caseID	activity	timestamp	resource	caseResourceGap	minGap	resourceGap	resourceCaseGap	caseGap
341	Repair (Complex)	1970-01-09 16:52:00	SolverC1		720000			720000
387	Repair (Complex)	1970-01-09 18:01:00	SolverC1	0	300000	4140000	3840000	300000
369	Repair (Complex)	1970-01-09 22:24:00	SolverC1	0	2040000	15780000	13740000	2040000
412	Repair (Complex)	1970-01-10 00:44:00	SolverC1		1500000			1500000
379	Repair (Complex)	1970-01-10 01:35:00	SolverC1	0	180000	3060000	2880000	180000
...	...	.....	....	.....	....	.....	....	...
354	Test Repair	1970-01-09 16:22:00	Tester1		600000			600000
351	Analyze Defect	1970-01-09 16:39:00	Tester1	0	420000	1020000	600000	420000
341	Test Repair	1970-01-09 16:58:00	Tester1	0	360000	1140000	780000	360000
387	Analyze Defect	1970-01-09 17:32:00	Tester1	0	360000	2040000	1680000	360000
337	Test Repair	1970-01-09 18:06:00	Tester1	0	480000	2040000	1560000	480000
..	.....	.....	...	.....	....	.....	.....	.....

- For each gap type, and for each 'group by' on a specific characteristic, we can do three types of analysis:
  - Metrics
  - Decision Tree
  - Gaps Evolution over Time

Gap Analysis
Metrics

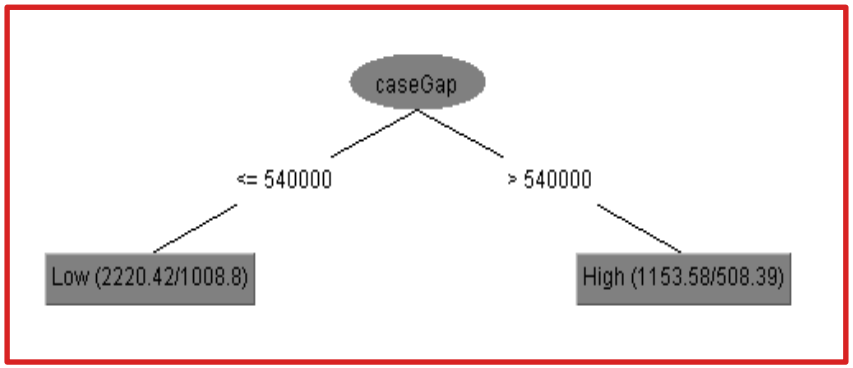
Metrics supported: mean, median, and standard deviation



Gap type: resouce gap; Group by: resource

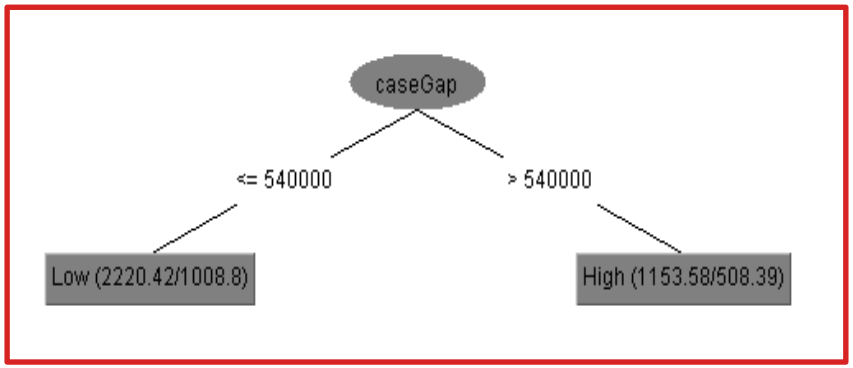
Gap Analysis
Decision Tree

caseID	activity	timestamp	resource	caseResourceGap	minGap	resourceGap	resourceCaseGap	caseGap
341	Repair (Complex)	1970-01-09 16:52:00	SolverC1		720000			720000
387	Repair (Complex)	1970-01-09 18:01:00	SolverC1	0	300000	4140000	3840000	300000
369	Repair (Complex)	1970-01-09 22:24:00	SolverC1	0	2040000	15780000	13740000	2040000
412	Repair (Complex)	1970-01-10 00:44:00	SolverC1		1500000			1500000
379	Repair (Complex)	1970-01-10 01:35:00	SolverC1	0	180000	3060000	2880000	180000
...	...	.....	....	.....	....	.....	....	...
354	Test Repair	1970-01-09 16:22:00	Tester1		600000			600000
351	Analyze Defect	1970-01-09 16:39:00	Tester1	0	420000	1020000	600000	420000
341	Test Repair	1970-01-09 16:58:00	Tester1	0	360000	1140000	780000	360000
387	Analyze Defect	1970-01-09 17:32:00	Tester1	0	360000	2040000	1680000	360000
337	Test Repair	1970-01-09 18:06:00	Tester1	0	480000	2040000	1560000	480000
..	.....	.....	...	.....	....	.....	.....	.....



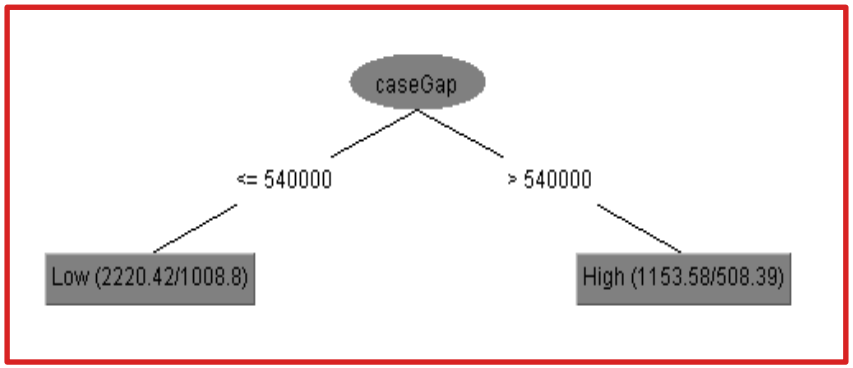
Gap Analysis
Decision Tree

caseID	activity	timestamp	resource	caseResourceGap	minGap	resourceGap	resourceCaseGap	caseGap
341	Repair (Complex)	1970-01-09 16:52:00	SolverC1		720000			720000
387	Repair (Complex)	1970-01-09 18:01:00	SolverC1	0	300000	4140000	3840000	300000
369	Repair (Complex)	1970-01-09 22:24:00	SolverC1	0	2040000	15780000	13740000	2040000
412	Repair (Complex)	1970-01-10 00:44:00	SolverC1		1500000			1500000
379	Repair (Complex)	1970-01-10 01:35:00	SolverC1	0	180000	3060000	2880000	180000
...	...	.....	....	.....	....	.....	....	...
354	Test Repair	1970-01-09 16:22:00	Tester1		600000			600000
351	Analyze Defect	1970-01-09 16:39:00	Tester1	0	420000	1020000	600000	420000
341	Test Repair	1970-01-09 16:58:00	Tester1	0	360000	1140000	780000	360000
387	Analyze Defect	1970-01-09 17:32:00	Tester1	0	360000	2040000	1680000	360000
337	Test Repair	1970-01-09 18:06:00	Tester1	0	480000	2040000	1560000	480000
..	.....	.....	...	.....	....	.....	.....	.....



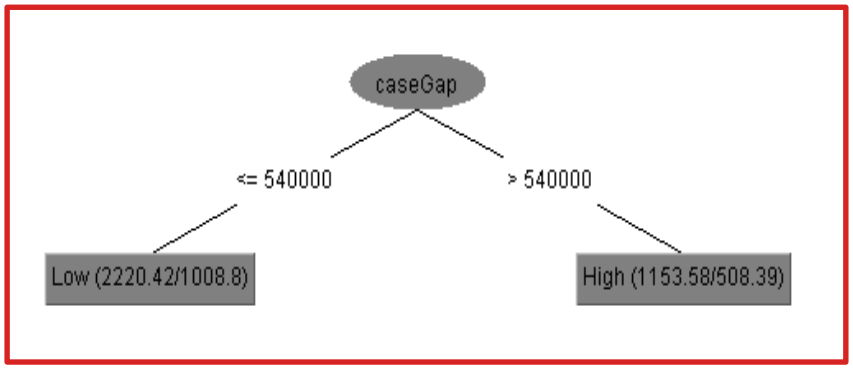
Gap Analysis
Decision Tree

caseID	activity	timestamp	resource	caseResourceGap	minGap	resourceGap	resourceCaseGap	caseGap
341	Repair (Complex)	1970-01-09 16:52:00	SolverC1		720000			720000
387	Repair (Complex)	1970-01-09 18:01:00	SolverC1	0	300000	4140000	3840000	300000
369	Repair (Complex)	1970-01-09 22:24:00	SolverC1	0	2040000	15780000	13740000	2040000
412	Repair (Complex)	1970-01-10 00:44:00	SolverC1		1500000			1500000
379	Repair (Complex)	1970-01-10 01:35:00	SolverC1				)000	180000
...	...	.....	....	.	Response Variables		....	...
354	Test Repair	1970-01-09 16:22:00	Tester1		600000			600000
351	Analyze Defect	1970-01-09 16:39:00	Tester1	0	420000	1020000	600000	420000
341	Test Repair	1970-01-09 16:58:00	Tester1	0	360000	1140000	780000	360000
387	Analyze Defect	1970-01-09 17:32:00	Tester1	0	360000	2040000	1680000	360000
337	Test Repair	1970-01-09 18:06:00	Tester1	0	480000	2040000	1560000	480000
..	.....	.....	...	.....	....	.....	.....	.....



Gap Analysis
Decision Tree

caseID	activity	timestamp	resource	caseResourceGap	minGap	resourceGap	resourceCaseGap	caseGap
341	Repair (Complex)	1970-01-09 16:52:00	SolverC1		720000			720000
387	Repair (Complex)	1970-01-09 18:01:00	SolverC1	0	300000	4140000	3840000	300000
369	Repair (Complex)	1970-01-09 22:24:00	SolverC1	0	2040000	15780000	13740000	2040000
412	Repair (Complex)	1970-01-10 00:44:00	SolverC1		1500000			1500000
379	Repair (Complex)	1970-01-10 01:35:00	SolverC1				)000	180000
...	...	.....	....	.	Response Variables		....	...
354	Test Repair	1970-01-09 16:22:00	Tester1		600000			600000
351	Analyze Defect	1970-01-09 16:39:00	Tester1	0	420000	1020000	600000	420000
341	Test Repair	1970-01-09 16:58:00	Tester1	0	360000	1140000	780000	360000
387	Analyze Defect	1970-01-09 17:32:00	Tester1	0	360000	2040000	1680000	360000
337	Test Repair	1970-01-09 18:06:00	Tester1	0	480000	2040000	1560000	480000
..	.....	.....	...	.....	....	.....	.....	.....



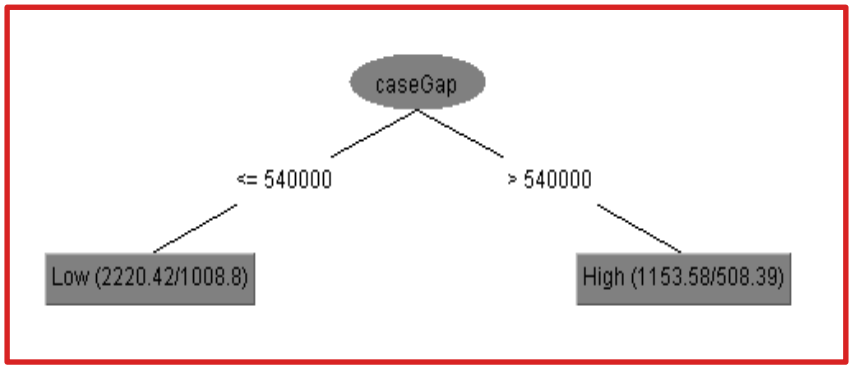
Gap Analysis
Decision Tree

caseID	activity	timestamp	resource
341	Repair (Complex)	1970-01-09 16:52:00	SolverC1
387	Repair (Complex)	1970-01-09 18:01:00	SolverC1
369	Repair (Complex)	1970-01-09 22:24:00	SolverC1
412	Repair (Complex)	1970-01-10 00:44:00	SolverC1
379	Re		olverC1
...	...		...
354	Test Repair	1970-01-09 16:22:00	Tester1
351	Analyze Defect	1970-01-09 16:39:00	Tester1
341	Test Repair	1970-01-09 16:58:00	Tester1
387	Analyze Defect	1970-01-09 17:32:00	Tester1
337	Test Repair	1970-01-09 18:06:00	Tester1
..	.....	.....	...

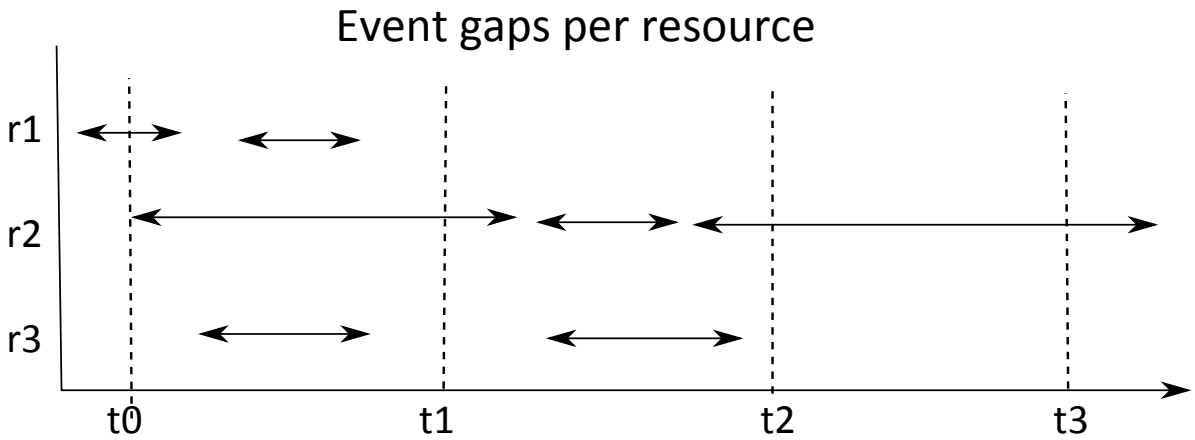
Predictor Variables

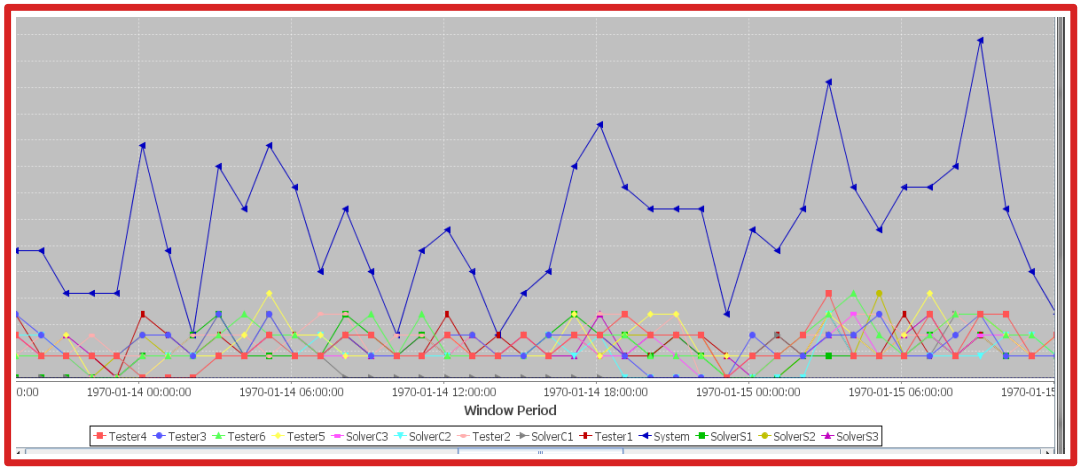
caseResourceGap	minGap	resourceGap	resourceCaseGap	caseGap
	720000			720000
0	300000	4140000	3840000	300000
0	2040000	15780000	13740000	2040000
	1500000			1500000
			000	180000
.			....	...
	600000			600000
0	420000	1020000	600000	420000
0	360000	1140000	780000	360000
0	360000	2040000	1680000	360000
0	480000	2040000	1560000	480000
.....	....	.....	.....	.....

Response Variables



Gap Analysis
Evolution of Gaps





Resource gap (or any other type of gap), group by resource.

Putting them all together

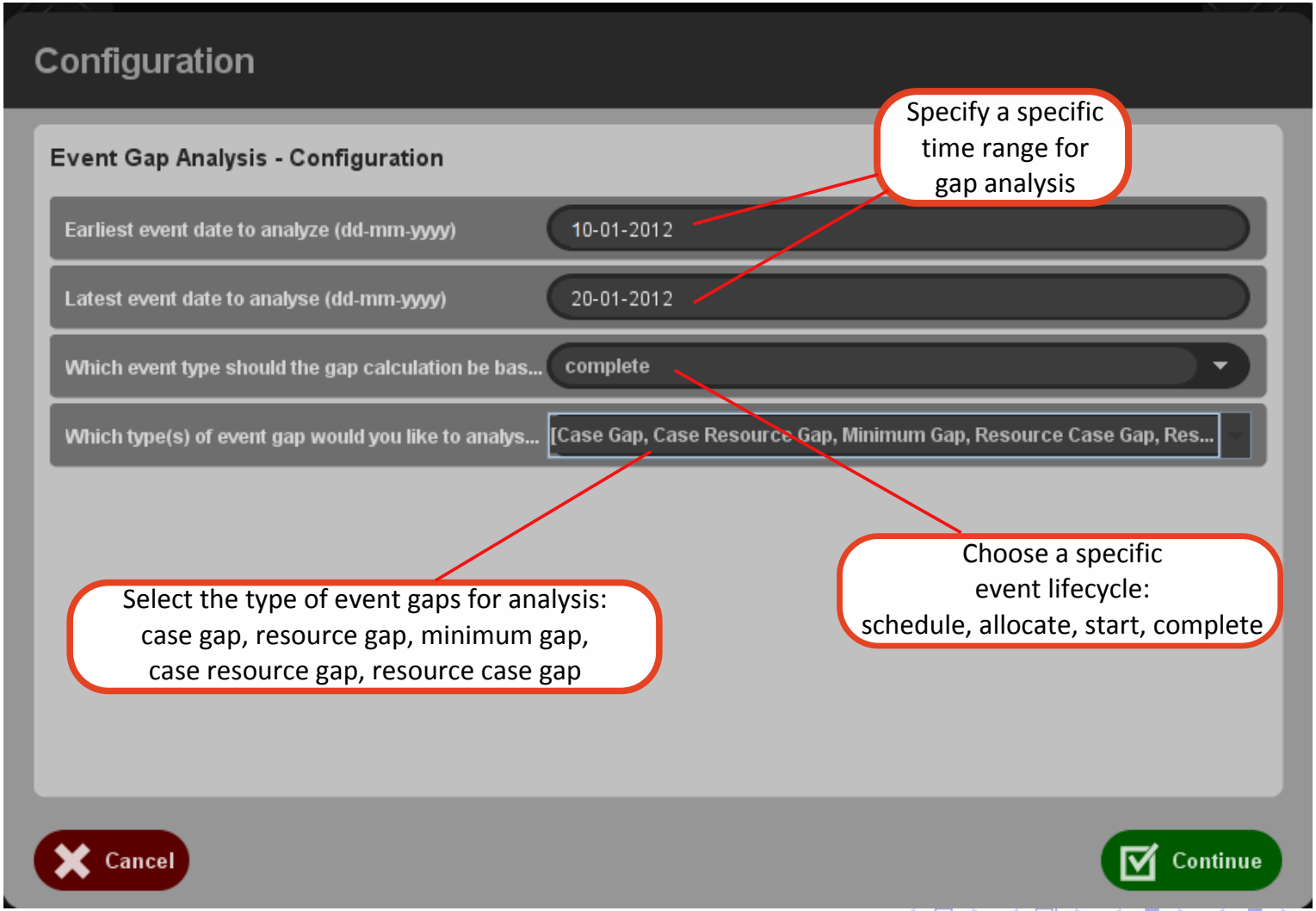
Gap Type	Group By	Analysis	Meaning
Case	Resource	Metrics	?
Resource	Activity		?
Minimum	Case	Decision Tree	?
Case-Resource	Team		?
Resource-Case	Null	Evolution	?

- For each analysis type, we can use 25 different 'grouping':
  - 5 gap types $\times$ 5 'group by'
- Interpretation for each grouping?
- All useful? Probably not...

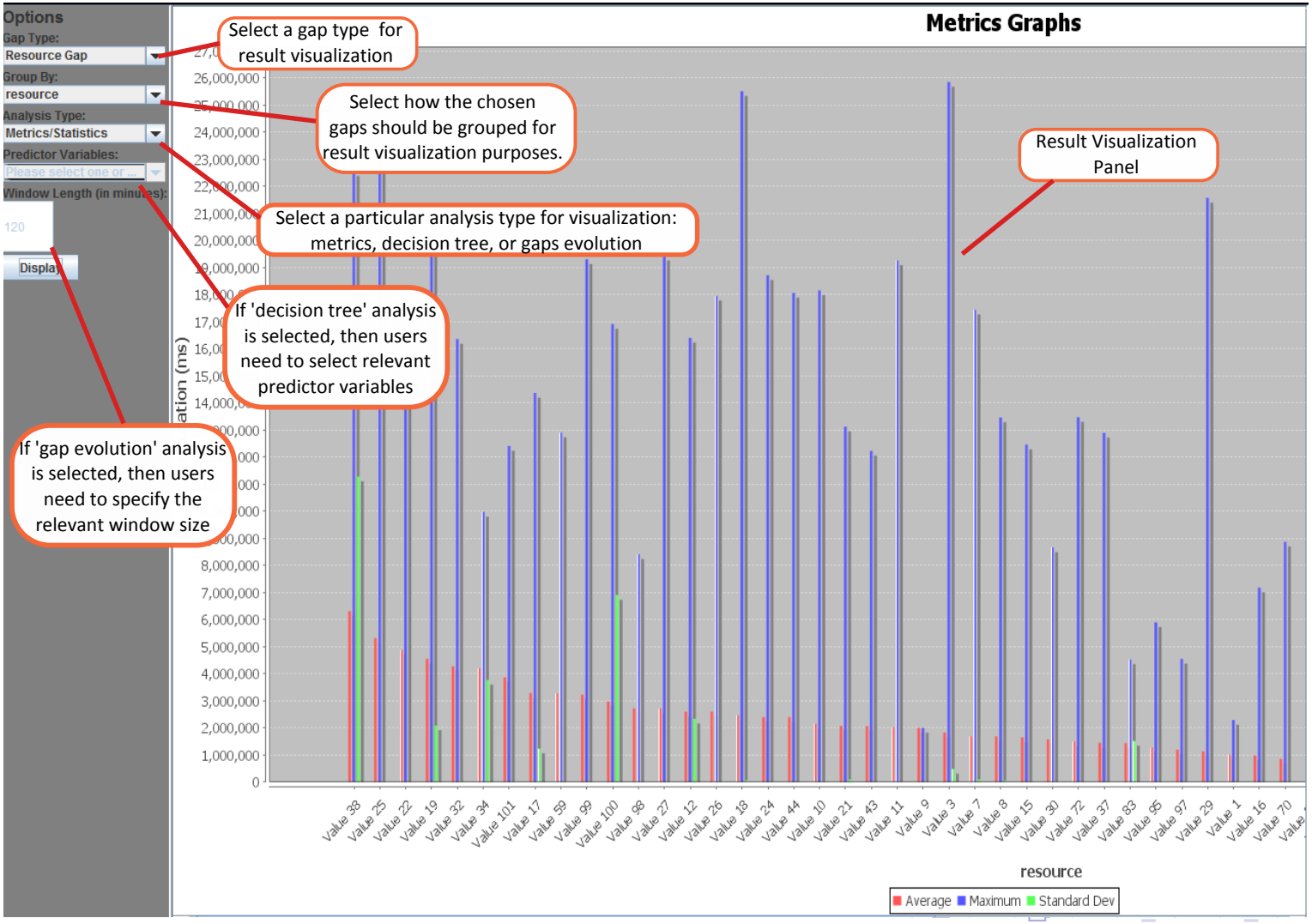
ProM Plug-in

- Has been implemented as a ProM plug-in “Event Gap Analysis”
- Input: XES/MXML log
- Configuration:
  - Period of time to use
  - Types of event gap
  - Event transaction lifecycle to consider

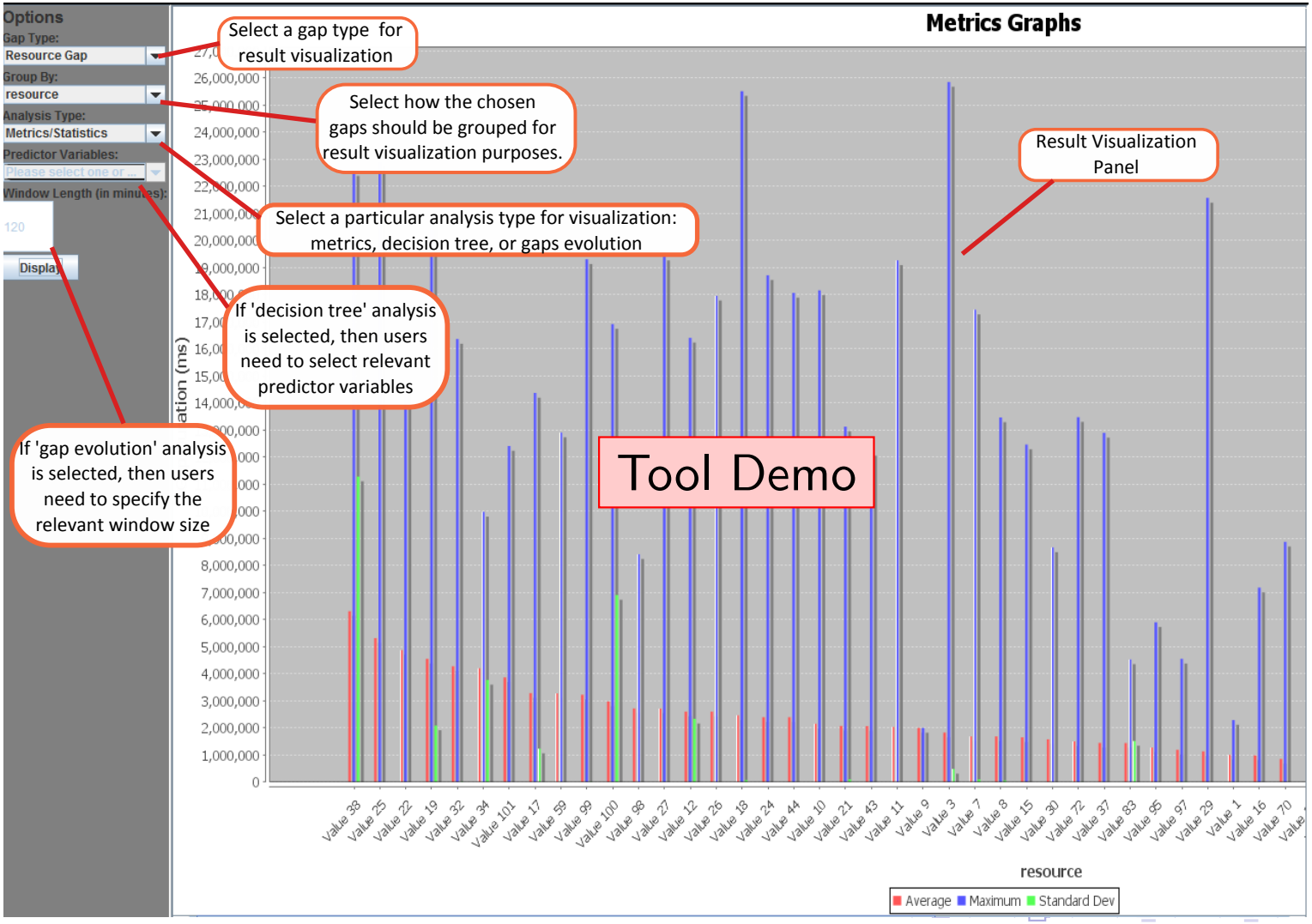
ProM Plug-in
Configuration Panel



ProM Plug-in
Visualization Panel



ProM Plug-in
Visualization Panel



Validation - Application to Logs

- Still need to be properly conducted
- Preliminary attempts with Suncorp data
- Results
  - Evolution graph shows peaks consistent with business hours
  - Distinct differences between resources
  - Decision tree: unclear
    - Make sense to correlate one gap (e.g. case gap) with another gap (e.g. resource gap)?
- Challenges:
  - Unrealistic short gap durations (in seconds/miliseconds)
  - May be due to log quality (imprecise recording of events)

Summary

- Event gaps analysis may provide useful insights about process behaviours, especially performance
- Conceptually straight-forward, potentially useful
- Has been implemented as a ProM plug-in
- Next steps:
  - Proper validation using a variety of logs
  - Suncorp, hospitals, BPI 2013 challenge
  - Exploit certain configurations of event gaps analysis (i.e. basic gaps + group by + analysis type)
    - to address typical business analysis questions,
    - leading to tool improvement to suit domain (business) users.
  - Publication strategy: conference then journal
- Collaboration sought

Log-based Event Gaps Analysis

S. Suriadi, C. Ouyang, W.M.P. van der Aalst, and A.H.M. ter Hofstede

Information Systems School
Queensland University of Technology
Brisbane, Australia

s.suriadi@qut.edu.au

June 13, 2013