

Report from Software Center –  
Project #18 Visualization of  
Continuous Integration  
Kristian Sandahl (LiU)  
Azeem Ahmad (LiU and Ericsson)

# Content

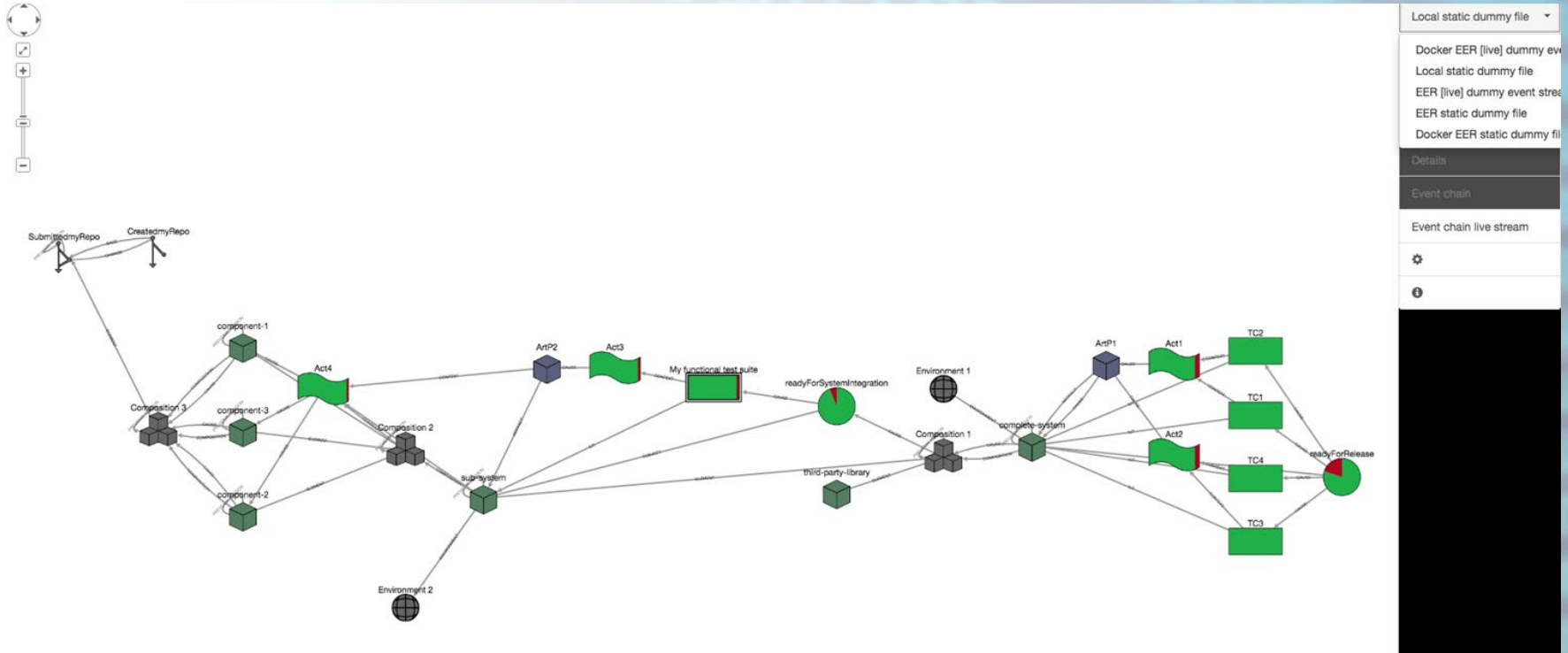
- Visualization projects

Vici

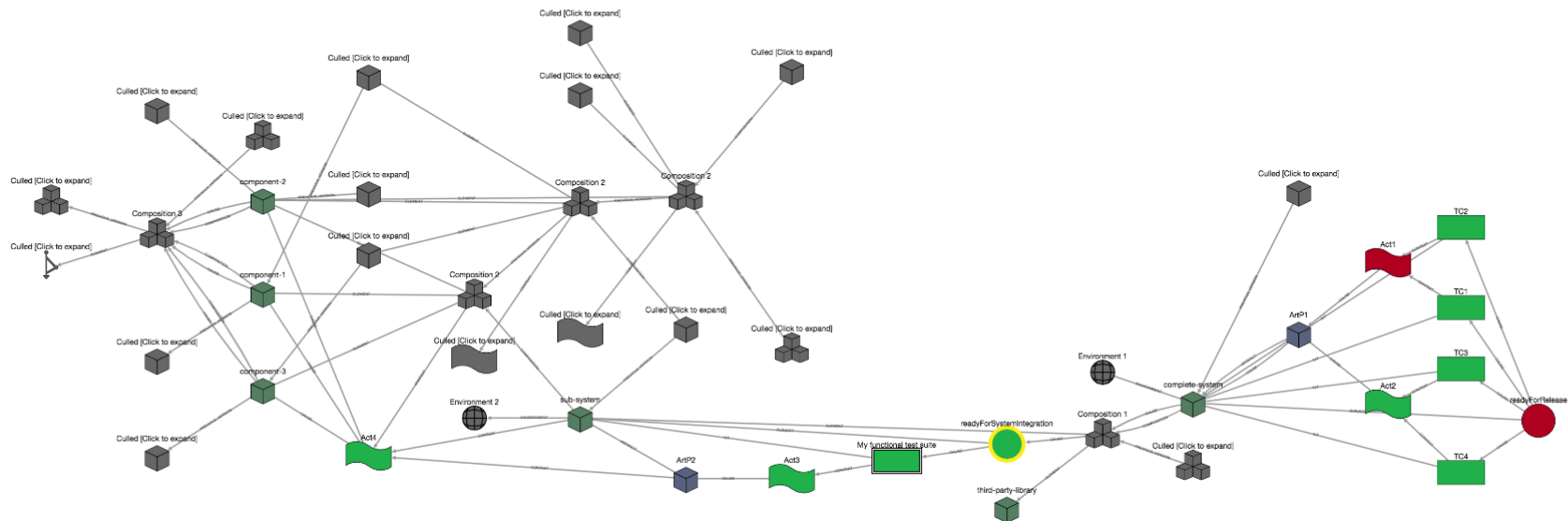
Eiffel Vis, Axis and University of Gothenburg

- Information needs
- Process mining

# Vici aka Eiffel Store



# Vici, detailed layer



Local static dump

Events collected  
minutes ago

Update events

Aggregation

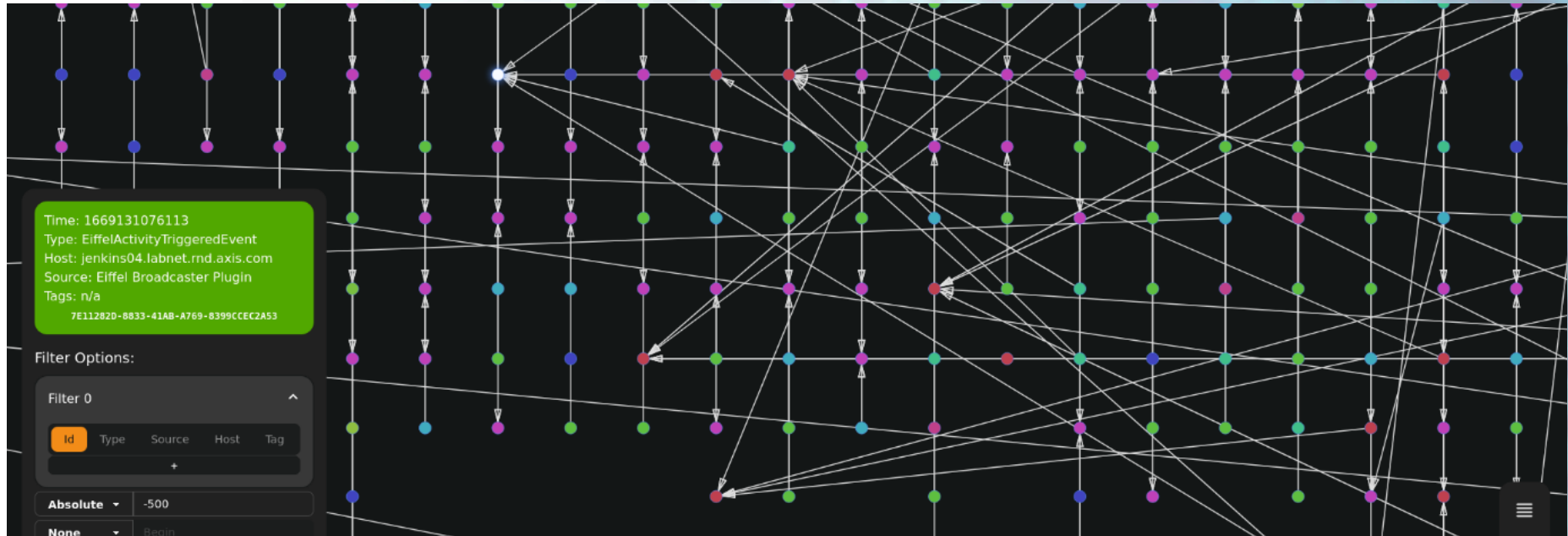
Details  
*readyForSystem*

Event chain  
*69f50654-b0fd-4*

Event chain live



# Eiffel Vis (Axis and University of Gothenburg)





# Information needs

- Two investigations
- Interview based: 27 needs
- Observation based: 39 needs (un-published)

<https://doi.org/10.1049/sfw2.12030>

5 – very important  
1 – good to know  
5 – very frequently  
1 – depends  
5 – very hard  
1 – easy

Information Need	Importance	Frequency	Effort	Time (min)
How much confidence do we have in the release to deploy to the customers?	4.8	4.9	5.0	15-20
Is the given feature ready to release to customers?	4.5	4.6	5.0	15-20
Is the bug fix ready to release to customers?	4.5	4.6	5.0	15-20



# Different categories

## Interview:

- Testing
- Code & commit
- Confidence level
- Bug
- Artefacts

## Observation:

- + Configuration & CI jobs
- + Testing infrastructure
- + Trends & statistics
- + Release & deployment
- Bug
- Confidence level

# Information needs, where to go from there

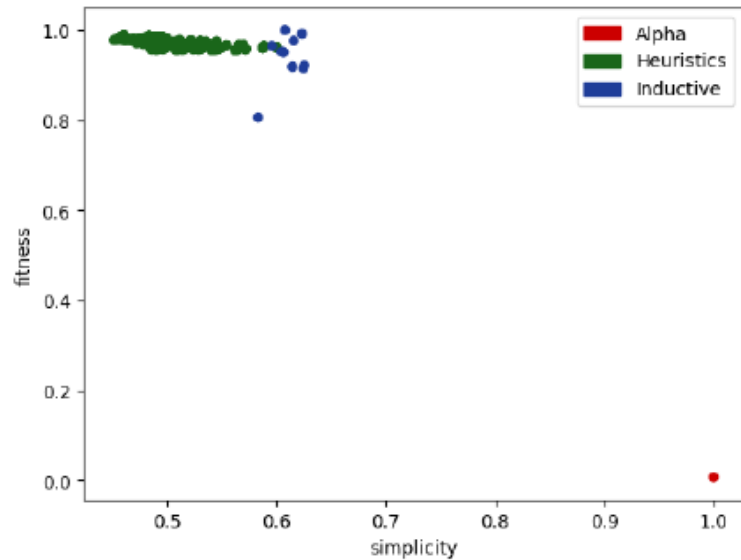
- Speculative:
- Use the needs to form queries for the database
- Consider de-coupling query engine – data storage



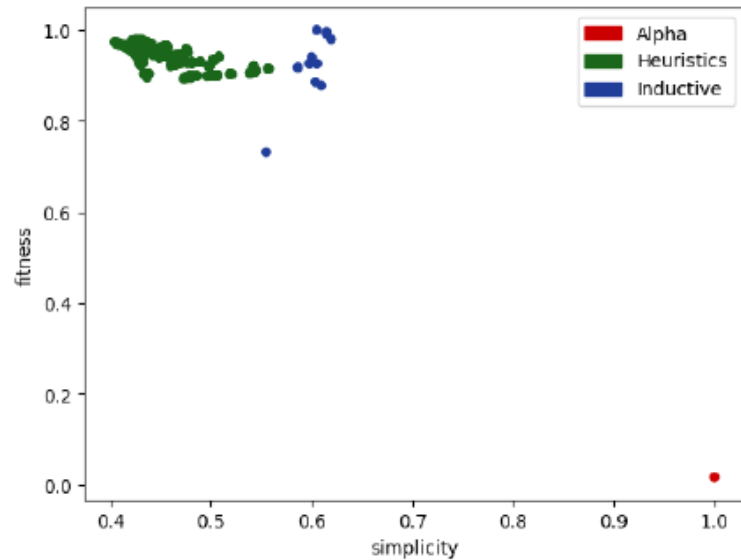
# Process Mining

- Master's thesis by Axel Ekblom and Jacob Karlén at SECTRA
  - Alpha miner
  - Inductive miner
  - Heuristics miner
- Hard to find interpretable models
- Pre-process filtering did not have significant effects
- Prototype work-bench was a success
- Max data set 240 000 events

# Results



(a) Dataset 1 (MG)



(b) Dataset 2 (CT)

# Process mining, where to go from here

- Create a prediction model for the coming events
- This can allow batching for resource-efficient CI/CD
- Possible techniques:

Graph neural networks <https://distill.pub/2021/gnn-intro/>

Small language models <https://paperswithcode.com/paper/it-s-not-just-size-that-matters-small>

...

Hope to see you at software center  
reporting workshop tomorrow!