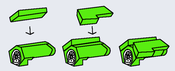
FAS Simulator – Executive Summary

Tero Keski-Valkama
Faculty of Computing and Electrical Engineering, Tampere University of Technology

Topic

- Once upon a time, there was a factory.

- The factory had a process.



- The process creates a log file.

Product moved to Workplace 1
Part A attached to Product
Product moved to Workplace 2
Part B attached to Product
Product moved to Workplace 3
Product tested

- Multiple products are going through the factory at the same time.
- The log becomes interlaced with process instance 1 and process instance 2.

Product moved to Workplace 1
Part A attached to Product
Product moved to Workplace 2
Product moved to Workplace 1
Part A attached to Product
Part B attached to Product
Product moved to Workplace 3
Product moved to Workplace 2
Part B attached to Product
Product tested
Product moved to Workplace 3
Product tested

- The factory owner adds an internet of things vibration sensors to the equipment, to get information when the equipment is started and stopped. Additionally, there is a periodic clock with tick events unrelated to the main assembly process.
- These events are mixed into the logs along with the process events. *)

Product moved to Workplace 1
Equipment 1 started
Tick
Equipment 1 stopped
Part A attached to Product
Tick
Product moved to Workplace 2
Product moved to Workplace 1
Tick
Equipment 1 started
Equipment 1 stopped
Tick
Part A attached to Product
Equipment 2 started
Tick
Equipment 2 stopped
Part B attached to Product
Product moved to Workplace 3
Tick
Product moved to Workplace 2
Equipment 2 started
Tick
Equipment 2 stopped
Part B attached to Product
Tick
Tick
Product tested
Product moved to Workplace 3
Product tested

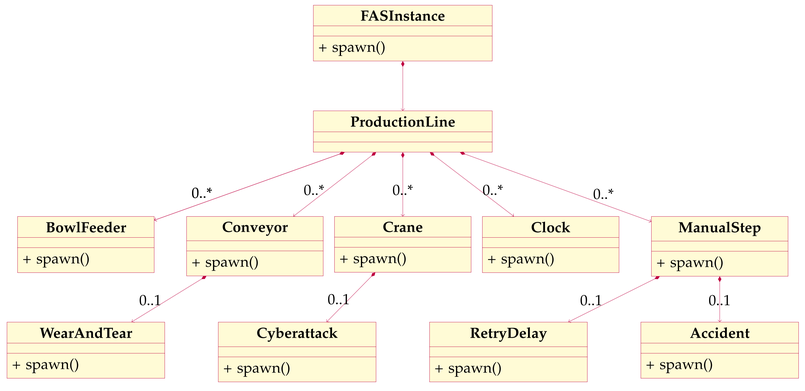
Idea

- If we had a learning system that can learn these kinds of processes from observing the logs, it could alert us if it sees something unexpected.
- How can we support the design of a learning system that is adept at learning process models from symbolic logs?

Discrete Event Simulator

- A Discrete Event Simulator (DES) is a software program which simulates a system in an event-oriented fashion.
- The system consists of component processes which wait and send events, and interact with each other.
- The simulation output is a sequence of events with timestamps.
- Different kinds of faults that are currently difficult to detect can be incorporated to the simulation.
- We can generate a lot of logs to get statistically significant measures of the performance of different methods.
- FAS Simulator simulates a plausibly realistic flexible assembly system with an assembly process and several different kinds of systemic faults that are not trivially captured by standard diagnostic codes.
- This simulator can be used as a benchmark for evaluating different learning systems.

Simulator

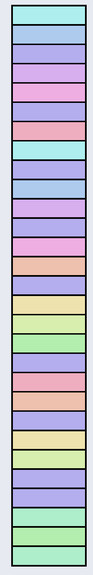


From Simulators to Learning Systems

All kinds of industrial and logistic processes create event logs. The events in these logs are created by different devices, and often we do not know the explicit process model of the processes that create these events.
Wouldn't it be great if we could create a machine learning system that can observe these kinds of logs, and deduce a model of the process?
There are some existing methods for process mining, but those are too limited. Current process mining methods can only extract an explicit process model from logs when the process instance is identified for each event. This identification does not exist in the general case.
The extracted process mined model is formal and definite but does not capture an intuitive understanding of the process. The formal process model only tells us which sequences are valid traces through the process model and which violate the model.
An explicit process model is not strictly required to detect process deviations.
Modern machine learning methods can capture an intuitive understanding and higher level features of such processes. For example, a system can observe that the new vibration sensor added is related to the process even if the vibration events are not matched one-to-one to a specific product. A learning system can also learn the normal number of timer ticks between two observed events.
Short process traces are a challenge for automatic learning methods. This kind of a simulation benchmark will work as a platform for developing interleave symmetry exploiting methods and potentially applying bayesian methods, one shot learning and transfer learning with immediate real world applications.

*) Computer's View to the Log

- As the computer does not know what events correspond to what products, the computer only sees the type of each event without knowing their meaning, like this:



- Each color signifies a different event type.
- It is in principle possible for a computer to discover features of such processes so that deviations can be automatically detected once the normal process has been learned.
- The main challenges are related to randomness inherent in the process and random interlacing of the events generated by parallel processes.