Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 119 (2017) 121–130

Procedia Computer Science

www.elsevier.com/locate/procedia

6th International Young Scientists Conference in HPC and Simulation, YSC 2017,
1-3 November 2017, Kotka, Finland

A Simulator for Event-oriented Data in Flexible Assembly System Fault Prediction

Tero Keski-Valkama

Neter, Kemiankatu 9 D 44, 33720 Tampere, Finland

Abstract

We present an open source discrete event simulator, FAS Simulator, for generating interleaved process traces of Flexible Assembly Systems. The simulation includes plausible faults creating anomalies in the output. The generated outputs can be used as a benchmark for evaluating novel anomaly detection methods.

Peer-review under responsibility of the scientific committee of the 6th International Young Scientist conference in HPC and Simulation

Keywords: Discrete Event Simulation ; Flexible Manufacturing System ; Flexible Assembly System ; Anomaly Detection

1. Introduction

This paper describes a method for simulating event-oriented data sources of a flexible assembly system (FAS) including faults. Such a simulator is required for creating representative corpuses of data for teaching automated learning algorithms for predictive maintenance, condition monitoring and systemic fault detection.

Flexible assembly system was described by Donath and Graves [8] as a collection of flexible assembly cells comprising one or several work stations connected by automatic material-handling devices. Flexible manufacturing system (FMS) refers to similar systems which are not limited to assembly. In practise, a flexible assembly system contains parts and materials

* Corresponding author. Tel.: +358-40-706-9762.
E-mail address: tero.keski-valkama@neter.fi

1877-0509 © 2018 The Authors. Published by Elsevier B.V.
Peer-review under responsibility of the scientific committee of the 6th International Young Scientist conference in HPC and Simulation
10.1016/j.procs.2017.11.168

stored in an intermediate storage, a conveyor or crane system to move the parts, materials, intermediate assemblies and finished products between the stations, and the stations with work machines and necessary tooling to assemble, process or inspect intermediate assemblies and products. The stations might consist of for example manual assembly steps, and robotic assembly cells. Often the assembly sequence has multiple ways of being executed for a finished product leading to some redundancy and more efficient scheduling in terms of utilization.

As opposed to more specialized industrial production systems, flexible assembly systems are designed for smaller batches and greater flexibility so that the set of end products can vary more widely. Flexible assembly systems can be reconfigured more conveniently following the evolution of new versions of the end products. Flexible assembly systems also allow for a wider range of customization between the different end product instances of the same product. The nature of fast evolution of configurations and workflows, and varying operating conditions make planning maintenance more challenging.

System downtime is a significant cost for flexible assembly systems. System downtime is reduced by preventive maintenance typically scheduled periodically. Recently Internet of Things and opening of the networked industrial system APIs have created new possibilities for predictive maintenance and systemic fault detection. Different sites with similar flexible automation systems have widely different environmental and workload conditions which has an impact on the wear and tear of the flexible automation system components. There is a clear need to adapt maintenance based on actual conditions in the operation rather than by simply scheduling maintenance periodically[15].

Going from preventive maintenance towards predictive maintenance optimizes and targets the maintenance related costs towards the activities that have best impact on improving the availability and operation of deployed flexible assembly systems.

Predictive maintenance in flexible assembly systems benefits from automatic indicators for immediate and potential future faults. It is generally not possible to enumerate all the possible fault conditions and their sensory indicators of a flexible assembly system in an exhaustive fashion, as the rare errors are neglected and faults can present in an unexpected manner[4]. There is a need for autonomously learning systems which are able to deduce the correct operation of the flexible assembly system and report potential deviances. One way to monitor a process is to observe the symbolic logs generated as the process is being executed.

Current predictive maintenance and condition monitoring systems concentrate on measuring device health for example by means of measuring temperature[24], vibration[28], lube oil particle analysis[16] and electrical current signals [29]. Research on anomaly detection for non-interleaved discrete sequences is summarized in a survey by Varun Chandola et al.[6] Interleaved discrete sequences, or uncorrelated process traces have been mainly researched in relation to specification mining for multi-threaded or asynchronous software processes. A solution for mining message sequence graphs by Sandeep Kumar et al.[20] works for message-oriented systems, and this solution requires sets of separate and complete traces corresponding to executions of an application[18].

In process mining there are several approaches in automatically extracting process models such as Alpha algorithm and its variants, genetic process mining, heuristic process mining, multi phase mining, and region-based process mining. In general these algorithms work with log messages labelled with the process instance.

Some success have been reported by Niels Landwehr using Mixture Hidden Markov Models [22] in labelling interleaved activities from event logs. However, these methods require manual labelling of the training data and are therefore unsuitable for an unsupervised setting. A method used in parallel software workflow mining described here [18] requires the execution logs to be separated into a large number of complete runs, and would not work with a flexible assembly system in a continuous operation.