Model Based Analysis of Industrial Equipment

Authors: André GELLRICH, Volodymyr VASYUTYNSKYY, Klaus KABITZSCH
Affiliation: Dresden University of Technology, Germany
Presenter: André Gellrich
E-Mail: andre.gellrich@tu-dresden.de

1. Motivation

Analysis of internal logistic processes in semiconductor equipment is necessary to evaluate optimization potential. Valuable sources of information about these processes are communication messages sent from the tool (equipment) to its host system (e.g. manufacturing execution system) where they can be logged and easily accessed. However, there's still a lack of software tools for analyzing these logs and extracting fault and performance information.

2. Description of the demonstrator

According to the widely applied SECS/GEM standard, tools communicate internal events (e.g. initialization, results of wafer processing steps and faults) to the host system. Our demonstrator uses these event messages to construct an abstract model of the internal equipment behaviour. Based on the model, significant statistics are calculated and visualized to support optimization, testing and fault diagnosis. The software is highly extensible and currently provides modules to assist the user with the following steps:

1. Formulation of transformation rules using regular expressions
   As there are no standards that define exact formatting, filtering or querying specific event data in SECS/GEM logs, our demonstrator provides a module capable of transforming arbitrary text files to XML documents using simple regular expressions.

2. Parsing and filtering of relevant data
   Having the logs transformed to XML, the demonstrator extracts the data needed to build a state model according to the user needs. The powerful query language XPath can now be used to dismiss uninteresting events and address specific data items e.g. within report-events. The SECS-logs we analysed usually contained a large amount of these reports, each giving detailed insight into the wafer processing steps involved whenever such a process was finished.

3. Building of state-transition models and calculation of characteristic values
   First, states corresponding to particular stations inside the equipment are extracted. The movements of wafers between stations are mapped to transitions. Along with the definition of static elements like stations, the tracing of single wafers or paths is possible. For each model element, characteristic values like *'processing time'* for states and *'overall processing time'* of single wafers in the equipment as well as in singular stations are relevant.

   Figure 1 shows stations of a *Clean Track Lithius Pro* equipment model, wafer movements as transitions between them and the minimum trace times of transitions of a specific path from one station to another. In the upper left corner, the trace time for some transitions is visualized in a bar chart. Using our approach, this model was extracted automatically from SECS/GEM-logs.

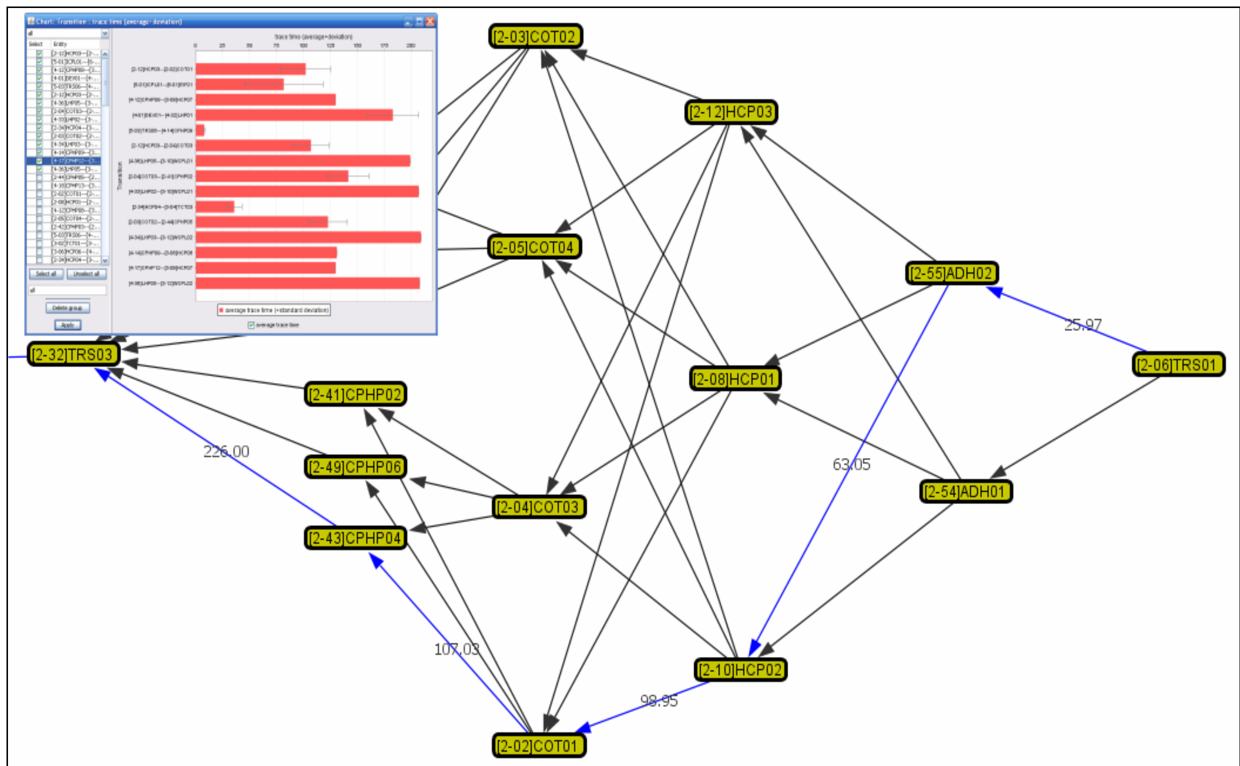

Fig. 1:State-transition model and a histogram of characteristic values

4. Analysis of characteristic values
   When the relevant characteristic values have been extracted from the logs, they can be analyzed using different statistical approaches. For example, wafers with abnormally long processing time may be detected as possible faults. Afterwards, the causes of these faults may be detected by inspecting the prehistory of the wafers and stations. The characteristic values also allow benchmarking of different equipments.

The generic state-transition model can be applied on different equipment types. This as well as definition of a set of model characteristics allows a quick adjusting of the log analysis to the specific purposes. The automatically built state-transition model (Fig. 1) allows a rapid identification of internal transport processes in equipment, enabling faster fault detection and classification (FDC). The approach can also be applied to other applications like analysis of internal logistics within a factory or conveyor handling system, analysis of processes in distributed systems and so on. The current work on the demonstrator focuses on improved visualization, usability and integration of a default set of diagnosis rules for some classes of applications.