

PROCESS MONITOR AND SIMULATION IN CIRCUIT BOARD ASSEMBLY USING WEB TECHNOLOGY

Hua Xiao Du, School of Industrial and Systems Engr. Siemens

Andrew Dugenske, Manufacturing Research Center, Georgia Institute of Technology

Chen Zhou, School of Industrial and Systems Engr, Georgia Institute of Technology

Introduction

Software integration in the factory is a big challenge. Amalgamating applications produced by dissimilar vendors, that make use of varying protocols, programming languages, data structures, operating systems and interface definitions can cause even the most experienced programmer to question whether they have chosen the correct career path.

Evidence of these challenges is the high cost of installing new software into the factory. Depending upon the specifics, the installation cost of a new application can easily range between 4x and 20x of the purchase price. The main reason for these high costs is the ad hoc nature of connecting the new software to systems present in the factory. Special wrappers and translators must be developed so that new applications can work within the confines of the legacy systems. Flexibility also suffers since the specialized code requires time to write, debug, and test. Business opportunities can disappear while this code is being developed.

Recognizing these challenges, the National Electronics Manufacturing Initiative (NEMI) has undertaken a project entitled, "The Plug and Play Factory." The purpose of this project is to design, develop, and recommend a software framework for use in the electronics assembly industry that greatly reduces the cost and time required for new software and equipment to reach the break-even point. Over a two year period, more than 16 companies and organizations participated in the NEMI project that tested a variety of framework types. The

frameworks were tested in the Center for Board Assembly Laboratory (CBAR) at the Georgia Institute of Technology. The CBAR lab contains a complete electronics manufacturing line that greatly assisted in testing the frameworks. Sometimes, however, it was helpful to make use of a virtual machine to rapidly test certain concepts. This paper describes issues, approaches and the concept of virtual machine, a simulated environment for testing monitoring systems.

The Monitoring Framework

The monitoring framework is based on HTTP/XML protocol. The user interface is browser based. This combination provides very high level of interoperability across different computers, operating systems, programming languages, databases and applications. The power of HTTP protocol lies in its simplicity and wide support. The power of XML lies in its extensibility. However, it is up to the users to define data dictionary and message structure. HTTP/XML do not provides links between the states of processing machines or host computer. .

In this project, we defined alternative frameworks in terms of module definitions, interfaces between modules and message transfer. We then test various framework in our laboratory that consists of a full scale, conveyerized SMT line.

Process Simulation with Virtual Machine

A virtual machine is a computer simulation model that mimics the behavior of a processing machine. In order to highest level of interoperability and accessibility, we implemented the virtual machine as a server side application. We focus on the simulation of a real machine in terms of state change and information exchange, not the physical or chemical aspects of the process such as forces, heat exchange and curing processes. A user can access a virtual machine via the internet using a browser. A collection of virtual machines can be used to simulate a manufacturing system. The concept is different from traditional discrete event simulation in that each machine can accept, process and respond messages in real time. Therefore, the dynamic configuration and real time control are inherent in the system design. It

also provides human-machine interface that allows a client to configure the machine behavior such as responses to specific messages. It can also display process information such as current state, percentage of time the equipment is in each of the state. It can also display productivity measures such as utilization, throughput and cycle time.

The virtual machine itself and its interactive partners can be replaced by any real equipment or host. As long as they support the generic HTTP/XML protocol described in the following. As a result, the virtual machine can also be used for prototype controllers.

In this project, the behavior of the virtual machine is based on a state model. In this section, we introduce the state model and the technologies used in the virtual machine. We then describe software logic and virtual machine interfaces.

Equipment State Model

In the designing of the state model, we attempted to capture all the generic characteristics of computer controlled equipment. We referenced machines in the surface mount technology (SMT) industry and received numerous feedback from the practitioners in the industry. We also try to strike a balance between simplicity and functionality. The result is depicted as a State machine in Harel definition, shown in Figure 1.

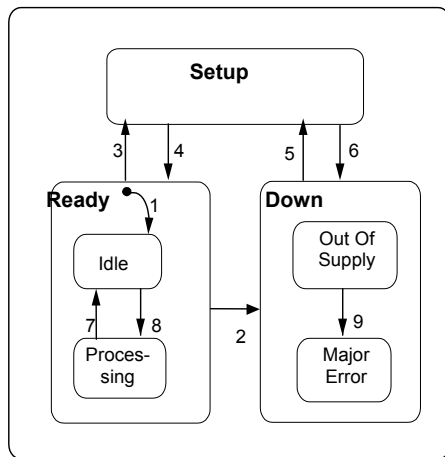


Figure 1. Equipment State model.

Software Architecture

In order to provide high level of flexibility in dynamic functions, we implemented

the virtual machine as a servlet running within a Java Web Server (JWS). It communicates with supervisory controllers and other machines over the Internet with web interface HTTP/XML protocol.

A servlet is a small Java application that runs on the server side to extend and enhance web servers. Servlets provide a component-based, platform-independent method for building web-based applications. Today, it is a popular choice for building interactive web applications.

JWS supports the generation of dynamic web sites that are extensible, easy to administer, and secure. It is the platform for servlet development: an easy way to make a dynamic web site.^[*]

The Hypertext Transfer Protocol (*HTTP*) is an application-level protocol for distributed, collaborative, hypermedia information systems. It is a request/response protocol, and has been in use by the World-Wide Web global information initiative since 1990. The Extensible Markup Language (*XML*) is a generic format for markup documents with tags. The structure and tag definitions can be defined by user groups in a document type definition file (DTD). It has been supported by popular browsers.

With servlets, JWS, and HTTP/XML protocol, the virtual machine can be easily interfaced with platform-independent clients. The only requirement on the client is a popular browser.

The virtual machine engine itself is a servlet, automatically launched by JWS. Its main thread is always standing by for external requests, while a sub-thread will be created in the background to simulate equipment operations and execute client instructions. The logical flowcharts for the virtual machine operation are shown in Figure 2.

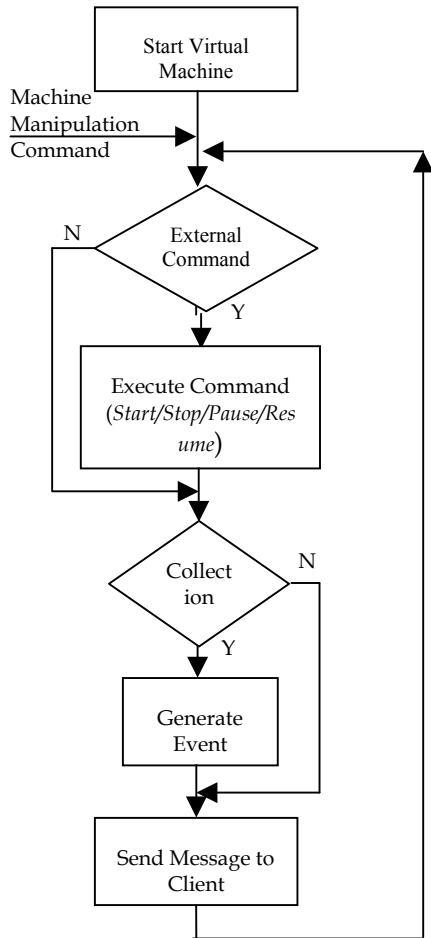


Figure 2. Logical flow of the virtual machine

User Interface

We designed a virtual machine control console for a user to monitor and manipulate the machine from a browser. The console offers control, Performance and Communication panels a user can use to configure, manipulate and monitor a machine. In the control panel, there are sub panels for Production Management, Process Program Management and Material Management. The Production Management (PM) sub-panel is shown in Figure 3. In the PM, a user can start, stop the simulation and can pause and resume the machine operation. It also highlights the current machine state.

In the Performance Panel, there are sub-panels for Equipment Time, Productivity Analysis and Quality Analysis. The Equipment Time interface is shown in Figures 4.

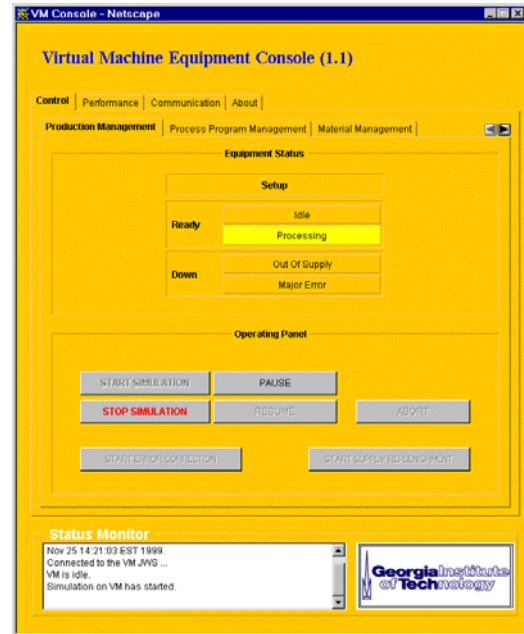
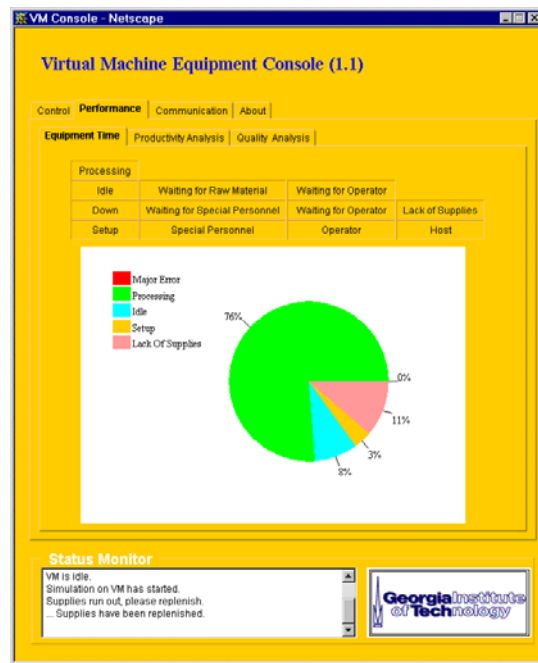


Figure 3. Production Control sub-panel in Control



Panel.

Figure 4. The equipment times from Performance console.

Acknowledgements

This project is in part supported by National Electronic Manufacturing Initiative (NEMI). We also benefited from many participate members in the NEMI Plug and Play project.

Biographical Sketch

Hua Xiao Du just joined Siemens Energy and Automation group. She was Ph.D. student in the School of Industrial and Systems Engineering at Georgia Institute of Technology. She received her bachelor degree in Tsinghua University China in Automation in 1995.
Short Dugenske Bio

Andrew Dugenske is Research Manager and Senior Research Engineer at the Georgia Institute of Technology's Manufacturing Research Center in Atlanta where he has been leading industry sponsored projects in electronics assembly for the past nine years. He received his B.S. from the University of Illinois at Urbana-Champaign and M.S. from Georgia Tech.

Chen Zhou is an Associate Professor in the School of Industrial and Systems Engineering at Georgia Institute of Technology. His interests are in robotic applications, process monitor and control and dimensional measurements. He received his MS in Mechanical Engineering and Ph.D. in Industrial Engineering in the Pennsylvania State University.