SATURN SITE PRODUCTION FLOW REQUIREMENTS, CONSTRAINTS, ISSUES

Earl Long
Saturn Corporation
Springhill, TN

Amit Misra
Vanderbilt University
Nashville, TN

ABSTRACT

The use of Information Systems (IS) in manufacturing plants for monitoring, control, diagnostics, etc. has increased considerably in the recent past. The primary drivers are efficiency and quality increase through automation, facilitation of better business processes and improved decision making. Many problems and issues relating to the design, development, integration, evolution and maintenance of ISs in large-scale and complex plants have become apparent. Model Integrated Computing (MIC) [1] offers a feasible approach towards providing cost-effective development, integration, evolution and maintenance of ISs through the extensive use of models during the life cycle. This paper describes an application of MIC in providing a problem-solving environment and decision support tool in the context of discrete manufacturing operations at Saturn. The Saturn Site Production Flow (SSPF) system is a client-server application, designed to provide an integrated problem-solving environment. It presents consistent and pertinent information, provides analysis and decision support services that are needed for informed decision making by the team members and leaders within Saturn.

INTRODUCTION

The use of Information Systems in manufacturing plants for monitoring, control, diagnostics, etc. has increased considerably in the recent past. As more and more manufacturing plants seek to automate operations and facilitate better business processes and decision making through the use of ISs, many problems and issues relating to software design, development, integration, evolution and maintenance have become apparent. The problems arise out of the scale and complexity of plants, the diversity of IS applications employed, and the tight interdependence between the two. Typically, there are a myriad of IS applications developed and deployed for different purposes including control, monitoring, diagnostics, simulation, analysis, and business decision making. Model Integrated Computing (MIC) [1] offers a feasible approach towards providing cost-effective development, integration, evolution and maintenance of CIS through the use of design-time models of the system to provide a common framework for different applications.

In this paper, we describe the requirements, constraints and issues concerning the application of MIC towards providing a problem-solving environment and decision support tools in the context of discrete manufacturing operations at Saturn Corp. The Saturn Site Production Flow (SSPF) system is a client server application designed to meet an initiative within Saturn Manufacturing to increase the number of cars built utilizing existing facilities and processes. The primary focus of tools and services provided by SSPF is the flow of material through the production facility. SSPF is intended to provide an integrated problem-solving environment which presents consistent and pertinent information, and analysis and decision support services that are needed for informed decision making by the team members and leaders within Saturn.

BACKGROUND

Information systems for complex environments have to be designed as evolutionary systems. Cost and operational concerns mandate the gradual introduction of new system components that can be integrated with existing ones. In this section, we summarize the context for SSPF and the underlying technology which is used for building the system.

Current Information Systems

There is a data rich environment at Saturn based on traditional process monitoring and control (PM&C). There are an estimated 80,000 points of live data that are monitored. The data being measured consists of production count, downtimes, bank counts, and other production related information. The majority of these data points are provided by the Programmable Logic Controllers (PLCs) of machines. This data is collected, logged and presented to users on status screens that are configured using a data acquisition and display package called Cimplicity [2]. This architecture has been in place since start of production in 1990. However, in the absence of any process and organizational models to guide the data collection, logging and presentation, the enormous volume of data presents considerable difficulties in using the system for monitoring site-wide status and for performing simulations and other decision making analyses.

During 1993, a new concept of “production flow” was designed and implemented at Saturn. It involved selecting key measurements instead of production data for all processes and buffers. These key indicators were identified through experience. The main new indicators that received emphasis were buffers between processes. Over the last 3 years, these indicators have proven to be extremely useful for the operations leadership. This proved a principle that excess data must be eliminated to provide only key indicators.

Beginning in late 1994, a manual evaluation process was initiated that involved calculation and posting of *standalone capacity*. This was intended as a key measurement of identifying bottlenecks for investment opportunities. The calculation of standalone capacity involves determining what the production rate of a process would be if there were no obstructions. Obstructions are:

- starving for material from upstream processes, or
- being blocked by processes downstream.

However, these results are updated and posted manually, once a month. In addition, the postings are not accessible to all Saturn team members and leaders, whose input is part of the business decisions at Saturn. One of the primary goals of SSPF is that it will make this, and all other real-time and historical data from across the site, available to all team members and leaders. In addition, standalone capacities, production rates, WIPs, blocking, starving, etc. will be calculated and updated in real-time. By presenting this data live, improved decision making and responsiveness is expected.

Model Integrated Computing

One of the primary difficulties in building ISs for complex environments is the tight conceptual relationship between the information system and the "physical environment." This tight conceptual relationship makes the design, implementation, integration and maintenance of ISs expensive. A particularly hard problem in manufacturing environments is that of maintaining consistency between the software and the plant. Manufacturing processes are dynamic, change frequently and include complex, interacting processes. An IS which is rigid, hard to change and does not support evolution through end-user programmability will quickly lose relevance in manufacturing environments.

Model-Integrated Computing addresses the problems by providing rich, domain specific modeling environments combined with model analysis and application synthesis tools. The key capability is end-user programmability. While the result of traditional software technology is a closed application, MIC products provide an end-user programming environment around the applications. The Model-Integrated Program Synthesis (MIPS) environments use concepts, relations, model composition principles and constraints that are known for the end-users, and automatically synthesize/re-synthesize application from the models created and maintained by end-users. The Multigraph Architecture (MGA) is a framework for creating MIC products [1] and is used in building the SSPF application.

MODELING CONCEPTS USED BY SSPF

The SSPF application offers a structured view of the data representing the state of the manufacturing processes. This structured view and the related visualization services create a tight conceptual relationship between the plant and the SSPF software. In this section, we summarize the key modeling concepts that are used for defining the SSPF application and that are also provided for the users of the system.

The Saturn manufacturing plant can be viewed as consisting of *processes* and *buffers*. Processes represent the operations required for making a car. An operation can involve casting, machining, welding, etc., of car parts, or it could involve assemblies from, e.g., Transmission Assembly all the way to final car assembly. Associated with each process are certain measurements that relate to the productivity of the process. Examples of such measurements are : *cycle-time*, *production count* (how many parts were machined, assembled, etc.), *Work In Process* (WIP) (how many parts are currently being worked on), *production downtime* (equipment breakdown, etc.).

Buffers (or *banks*) lie between processes and hold parts and/or sub-assemblies, that are produced by an upstream process before they are consumed by a downstream process. In different sections of the plant, buffers take on different forms -- they may be Kanbans, conveyors, etc. However, the information that is pertinent for production is common to all buffers -- *bank count* (number of parts/sub-assemblies in the buffer) and the *minimum and maximum capacities* of the buffer.

The *inter-connectivity* of processes and buffers captures the sequence of operations required to produce a car, and the interdependence of processes on each other and on the buffer capacities. A process may have to remain idle due to the fact that an upstream process is not producing enough parts (this condition is called *starving*). On the other hand, a process may be forced to stop producing if a downstream process is not consuming enough parts (this condition is called *blocked*). The starving and blocked conditions may arise due to mismatches between cycle-times of processes, their WIPs, the capacities of buffers between them, and many other reasons.

The concept of *production flow* is concerned with the flow of material (raw materials, parts, sub-assemblies, etc.) through the processes and buffers, and encompasses all the production related entities of processes and buffers (e.g. production count, WIP, bank count, starving, blocking).

The MIPS environment of SSPF allows the end-user to model the production flow at Saturn using a graphical model builder. The processes are modeled hierarchically, allowing abstraction of relevant production flow information at higher levels in the hierarchy. Figure 1 shows the model for Vehicle Initial Build, a section of the Saturn plant. The process Vehicle Initial Build has five sub-processes : Cockpit 100, Cockpit Test, Hardware 200W, Hardware 200C, Hardware 200E and Hardware 310 (not shown completely in the figure). The icons for the sub-processes can be seen in the figure. The interface points on the icons represent conveyor systems that deliver parts/sub-assemblies from buffers. The buffers between processes can also be seen (2200A, 2200B, 300, etc.). The production flow is represented by the connectivity between buffers and processes, as shown. There are many more types of models, aspects, properties and attributes which together comprise the SSPF modeling paradigm. However, they are not described here for the sake of brevity.

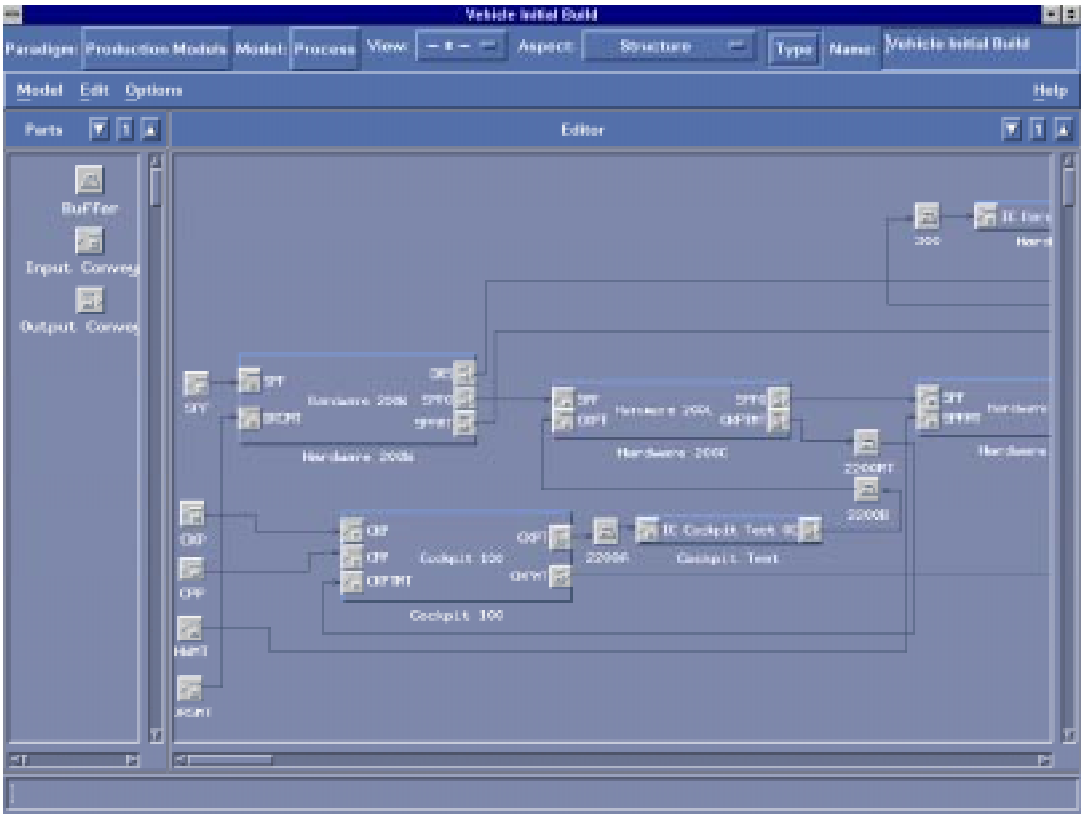

Figure 1: Graphical Model for Vehicle Initial Build

As mentioned before, the SSPF/MIPS models represent the information about the plant which is used to auto-generate the SSPF application including the operator interface. Figure 2 shows the operator interface for Vehicle Initial Build, as synthesized form the models.

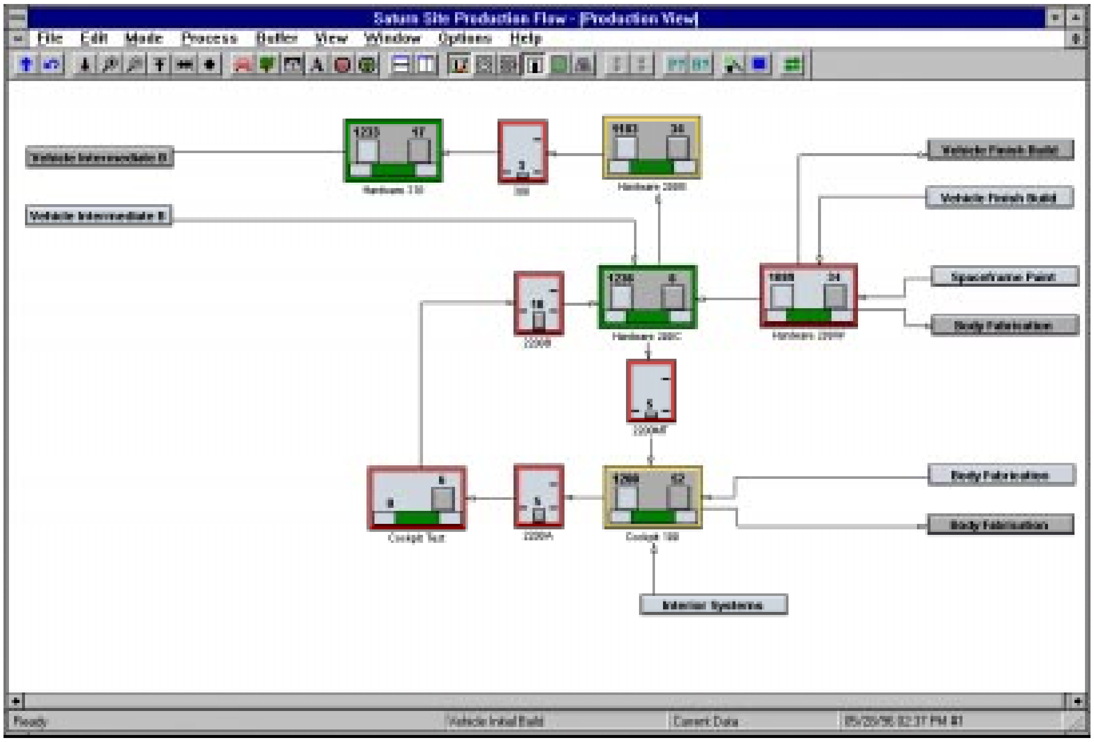

Figure 2: Operator Interface for Vehicle Initial Build

SSPF FUNCTIONALITIES

The SSPF is designed to be an application that evolves easily. The list of functionalities describe below represent the startup capabilities of the system, which will be extended in the future.

Data Acquisition

SSPF functions involve collection, presentation, storage, retrieval, and analysis of data. Data collection as mentioned previously is focused on data streams that currently exist in the Saturn environment. The primary source of data is from PLCs that is collected using Cimplicity, as mentioned earlier.

Data Storage and Retrieval

Time is an essential dimension in understanding the dynamics of production flow. There are some dynamics that are within the bounds of the current shift. Others involve multiple shifts, weeks, or months of history. There are “seasons” within the dynamics of production flow; statistically represented as a function of auto-correlation. To understand the dynamics of production flow, storage of data in a structured format is required. The stored data should contain detailed histories for every process and buffer in the plant. In addition, summarized information for a shift, day, week and month should be maintained.

Graphical User Interface

The fundamental business process that will be applied by SSPF is a concept called visual controls. Visual controls are intended to make obvious the variations in the process being performed by an individual. An example is taping an area on the plant floor for appropriate placement of material totes. As applied to SSPF, visual controls will be focused on making bottlenecks apparent and obvious.

In the current implementation, the SSPF GUI includes a navigation tool and a visualization tool. Drill down in the GUI is required to allow a user to answer questions concerning detail (what is causing the results being observed?). In normal use, a glance at a screen will either confirm that all is OK or that there is an exception (note the application of visual controls). In the condition of there being an aberration in the production flow, detail is required. SSPF provides standard techniques of “drill down”, drop down boxes, and alternate views to provide the user the level of explanation required. Figure 3 shows the SSPF GUI with hierarchical navigation, drop down boxes and detailed textual report. On the left, the process hierarchy is shown, which allows the user to go to any section of the plant and examine it. On the right the GUI layout for Vehicle Initial Build, as synthesized from the modes, and the textual report of production related data is shown. A drop down box for Hardware 200C (a sub-process of Vehicle Initial Build) is also shown, with summarized information about Hardware 200C.

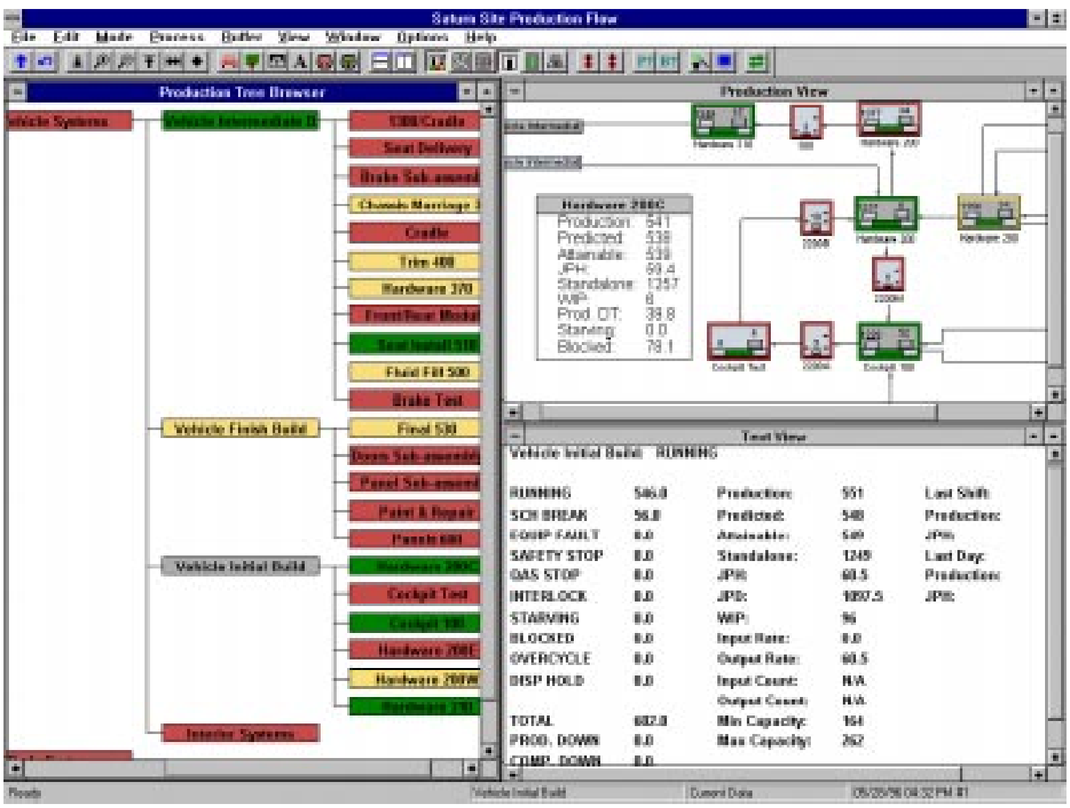

Figure 3: SSPF GUI

These GUI techniques provide for an ability to handle both depth and breadth of data views. In certain instances, it is most appropriate to adopt a detailed (depth) view while at times it is better to implement a sitewide (breadth) view. There are physical restrictions of the CRT, limiting the amount of data that can be presented. There is also a restriction introduced by the perception of the user. Both restrict what can be observed and understood by the user at any point in time.

Bottleneck Analysis

A key concept that is central to production flow is that of a *bottleneck*. A bottleneck is that process that is limiting the overall production flow. Bottlenecks have characteristics that are apparent and measurable. A bottleneck has more downtime than other processes in the production stream, little if any starving and blocking, and less or equal production counts compared to other processes in the overall stream. Also, bottlenecks have the characteristic of "hiding" potential bottlenecks. Bottleneck identification has been compared to lowering the level of water in a stream, removing rocks and then finding new rocks as the level is lowered. The overall concept of production flow is built on focusing attention on bottlenecks and potential bottlenecks.

Bottlenecks may at first appear to be obvious, which is true if the process is simple enough and can be completely observed from a single point. In the case of a large discrete manufacturing plant such as Saturn, this is not possible. Saturn processes are located in four separate buildings covering a total of 4.3 million square feet. Physical observation by a person is restricted to a small part of the overall process. SSPF is intended to provide a total virtual view of the production flow across the plant and thus, aid in clear identification of bottlenecks.

By using SSPF, bottlenecks will be identified and can then be resolved by team members. The first step toward resolving any issue consists of clear unambiguous data. Through clear presentation of data, team members will be able to understand the total production environment. Dependent on the role of the team member or leader, SSPF will also be used to support the problem solving process. SSPF will provide an integrated environment for various tools, which will be able to access current or historical data and perform analyses using a common framework. It will be used to test alternative solutions, examine various scenarios, simulate a proposed solution and perform any other required analysis before a making an informed decision.

Many tools and methodologies can be used for bottleneck identification and resolution, including analysis, simulation, data mining, etc. Currently, there are some tools and systems deployed at Saturn for performing analyses, simulations, etc. However, these systems are isolated from each other and many times the representations of the plant used by these systems differ significantly from each other. This makes it difficult to use the tools together, and to combine the results to facilitate better decision making. Using MIC technology, SSPF will provide an integrated problem-solving environment, which will allow Saturn personnel to use various tools together and make more informed business decisions.

Problem Solving Activities

Extending the capability of moving in time (between current and historical data), the SSPF user will also be operating in the future time domain. This is the context of data analysis where the user is applying analytical techniques to predict, understand, and evaluate possible future events. Simulation, predictive techniques, and decision support will be provided in a common framework using the same models that are used for real-time monitoring, data storage and retrieval. Using traditional techniques for analysis is facilitated through application of MIC. Tools can be readily integrated into the problem space, seamlessly mixed as appropriate.

The SSPF user is focused on making decisions concerning operations in order to improve throughput. By understanding the condition of the plant and in turn using one's experience, a prediction is made concerning the likely future. It is straightforward to understand that depleted banks upstream in the process will lead to a drop in the production count in the near future. Predictive tools involve applying experiential rules to the data expressing conditions to identify the most likely future condition of the plant. By understanding the conditions of the plant flow in the future, decisions may be tested and in turn predict the likely results.

Decision support tools help to design and verify interventions in plant operation. A decision support activity may include: data processing and visualization to recognize problems, bottleneck analysis for understanding the cause of the problem, and prediction to assess the impact of a planned intervention to resolve the problem. In this sense, decision support tools must help to define and execute integrated analyses. The components of these analyses may be elementary data processing, visualization, planning, prediction, etc., activities.

REQUIREMENTS

In this section we give a brief overview of some requirements, issues and concerns for SSPF.

User Interface

Presentation of real time data is to be provided through a Graphical User Interface (GUI) that is intended to readily show the important information concerning production flow. There is a design challenge in applying human engineering principles to the design of the screen presentation. Some concepts that are applied are simplification, drill down, and flexibility. Simplification involves selecting only the key information that a user needs to understand the current dynamics of the system. A rule of thumb is "understanding at a glance." An experienced user should be able to understand the current production flow in an extremely short period of time (less than 5 seconds).

Data Storage

SSPF is required to retrieve this stored data and provide presentation with the same tools as used in the real time presentation. This provides to the user a reduced learning curve and a logical view that is not restricted by time. It is expected that the user will readily move from the current time domain to the historical domain.

Integration

There are constraints for SSPF that are driven by the computing environment that exists in Saturn manufacturing. A strategic direction of client-server computing has been implemented. The operating system at the server level is Window NT and at the client level is Windows 95. These selections are intended to provide optimum hardware independence, cost, and user friendliness. It is also required that SSPF be a consistent and seamless application from the view of the user with respect to other Saturn applications. These applications have the look and feel of Windows based applications. Also a seamless interaction with the existing PM&C, Cimplicity is required. Within Saturn manufacturing, SQL Server is a standard for all databases. SSPF is required to follow that standard and meet all functional requirements. This requirement is driven by the need to provide a common environment for support personnel to handle.

There are expectations of the technology used in SSPF that must be met. Hardware independence is crucial in the design and implementation of SSPF. Experience and predictions are that the hardware used for deployment will not be the same throughout the life of the application. This is essentially the emphasis on the restriction to Windows operating systems. This equally applies to the other parts of SSPF.

Creating a “user friendly” application (application of human engineering principles) is also required. The users of the system will have a variety of skills and responsibilities. Some users will have advanced computer skills, others will be novices. By application of Windows based techniques, simplification of learning is achieved. It is expected that users will transport learning from other Windows applications the mechanics of using SSPF. Ultimately, usage must be obvious.

Performance

SSPF is expected to support a multitude of users with low load on networks, servers, or clients. The number of users is high (several hundred) because any person in the workforce at Saturn would be expected to be an active user. There must be no adverse economic penalties for deploying a successful application. There has been some experience with needing to ration availability of successful applications. SSPF is expected to avoid this condition.

Robustness

A significant design issue that needs to be addressed by the design of SSPF is the diversity of data sources. It is an absolute requirement that robustness and data integrity are achieved in spite of the lack of robustness and data integrity in the data stream. It is required that the SSPF data is significantly better than the data source being used. This is not unprecedented, simply this is filtering but on a scale of dynamic response to nonspecific conditions. There will be little human intervention, thus no one to check the integrity of the data. Because the data presented are *conclusions*, any error may or may not be obvious. The presence of random errors in the data stream is unavoidable. The natural conclusion is that unknown errors introduce a finite confidence in the data. This finite confidence needs to be apparent to the appropriate users.

Since there will be errors, methods that handle and manage errors are essential. There is a continuum of techniques for SSPF to handle errors ranging from automatic error detection and recovery to generating alarms requiring system operator response. The most preferable and robust method is to favor automatic error recovery; manual recovery should be reserved for those conditions that are judged to be beyond current technological solutions. The success of these error handling services will have an impact on operating costs. The degree of manual intervention will determine to a great degree the ongoing support cost for the system.

In either case of manual or automatic recovery, automatic error detection is required. Error can be introduced through several methods including loss of communication, failure or interruption of processes or services, PLC data collection failure, and undocumented logic changes in data collection processes. The expectation is that SSPF be constructed to detect when the data stream being received "does not make sense."

Ideally, the models will support automatic reconciliation of data that is inconsistent. Soft rules are routinely used in the current PM&C environment to conclude what data is correct when there are inconsistencies. These soft rules should be part of the plant models. The data stream that is identified as being incorrect should be overridden and the alternative data stream should be used. Creating an alarm to fix the primary data stream would be the appropriate course of action if an automatic recovery routine cannot be developed.

Alarms for operator intervention are a routine expectation of all applications used. There are multiple non-related applications that will be supported by system operators and other support staff. Alarms must be seamlessly integrated with other alarms so that appropriate operator intervention is achieved. Escalation and follow up by alarm management is also necessary. SSPF cannot assume that because an alarm was issued, appropriate action has been taken. Just as in normal non-computer activities appropriate escalation actions need to be taken to get the help that is required.

CONCLUSION

In this paper we presented a model-integrated solution for monitoring and analyzing production flow at a discrete manufacturing plant. MIC technology provides a conceptually strong solution framework for the SSPF application. Considering the diversity of requirements and data sources, complexity of logic would be high with a traditional code solution. The solution environment is quite chaotic. Traditional code solutions would require a significant effort in placing order to these data sources. This structure can readily be expected in a virtual view of the manufacturing plant floor through models. Diverse services are required for the application such as PM&C, process simulation, statistical analysis packages, and other data manipulation tools. Integration of these tools into a common problem-solving environment is readily achieved through the use of model integrated computing. MIC raises a level of expectation for achieving advanced and complex computation at very low cost. It is expected that through the use of the MIC, advanced analysis will be readily supported. The most fundamental business problem involved in the SSPF application concerns identification of bottlenecks and the ultimate prediction of the results of elimination of the bottleneck, specifically total throughput from an action or set of actions. The expectations is that through application of models, the creation of algorithms that emulate soft rules and the seamless integration of third party analytical tools, decision making can be supported in a cost effective manner.

REFERENCES

[1] Sztipanovits, J., Karsai, G., Biegl, C., Bapty, T., Ledeczi, A., Misra, A.: "MULTIGRAPH: An Architecture for Model-Integrated Computing," *Proc. of the 1st IEEE International Conference on Engineering of Complex Computer Systems*, pp. 261-369, Ft. Lauderdale, Fl., November 6-10, 1995

[2] "CIMPLICITY MMI and MES/SCADA Products: User Manual," GE Fanuc Automation, February 1996.