Saturn Site Production Flow (SSPF): Accomplishments and Challenges

Earl Long
Saturn Corporation
Spring Hill, TN
email:LNUSSAT. ELONG1@GMEDS.COM
Phone: (931) 486-6077
Fax: (931) 486-6845

Amit Misra, and Janos Sztipanovits
Measurement and Computing Systems Laboratory
Vanderbilt University
Nashville, TN 37235
email: {misra,sztipaj}@vuse.vanderbilt.edu
Phone: (615) 322-2771
Fax: (615) 343-6702

Abstract

The Saturn Site Production Flow (SSPF) system is a client-server application that has been developed using Model-Integrated Computing (MIC) approach. It is 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 to continuously improve plant throughput. With one year of successful production experience there are many accomplishments to recognize and challenges that will need to be met. These accomplishments and challenges occur in both the arenas of the business application and systems engineering. As a whole, MIC via the MultiGraph Architecture (MGA) [1] has significantly contributed toward SSPF being a successful application and therefore significantly contributing to Saturn's manufacturing throughput. An overall responsiveness to business needs and shifting requirements was successfully realized during the SSPF launch as well as in day to day production use. A robust, well-integrated system has reliably provided the throughput information required for running the business via informed decision making. The year's experience has also presented challenges that are or will be addressed. Business challenges concern providing a fit into the overall suite of manufacturing systems in use by Saturn or that are in development. Most of the systems engineering challenges concern resolving technical issues that enable specific required features or provide for an improved level of responsiveness to the application domain. This paper will be a thorough review of these accomplishments and challenges within the context that many of the lessons learned and problems to be solved are common with other MIC practitioners.

1. Background

SSPF is a system that was designed around the throughput improvement process that has evolved in Saturn Manufacturing's General Assembly plant over the last several years. That process involved using available data from Cimplicity© (a plant monitoring and control system from General Electric Fanuc), logging that data into a database and performing trend analysis of the data[4]. Data collected and used is production counts, production downtime, starving downtime, blocking downtime, etc. In conjunction with downtime reporting an analysis of bottlenecks is performed on each shift concerning the reasons for lost vehicles. Appropriate teams utilizing trends and vehicle loss analysis as a guide implement action plans.

Beginning in late 1995 we began work on SSPF development (initial production release in August, 1996) applying the same techniques for measurement as used in Saturn's General Assembly to the rest of the Saturn site[2]. A principle measurement adopted for measuring throughput consistently across the site is *standalone capacity*. This measure is the daily unobstructed

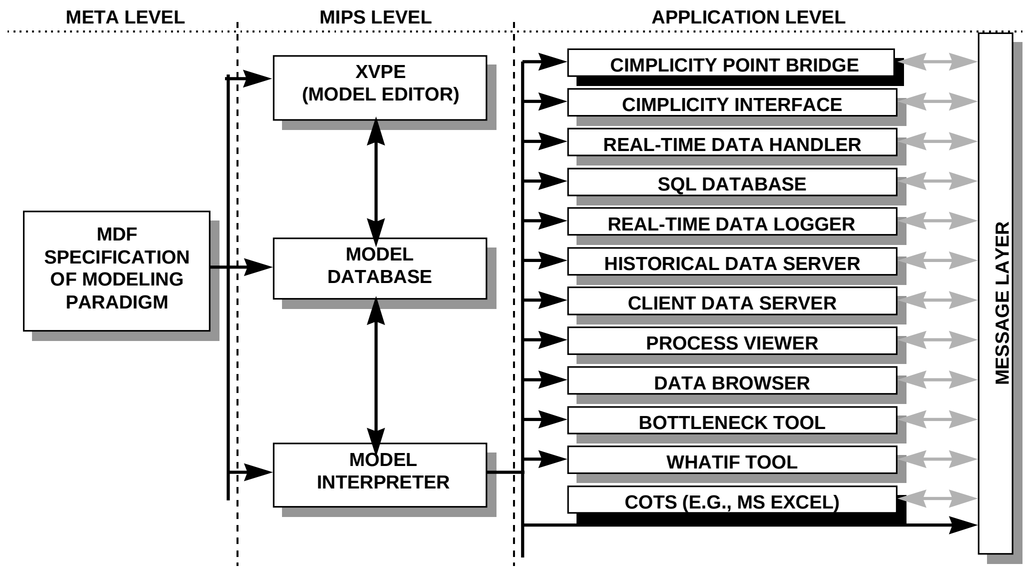

Figure 1: SSPF Architecture

production rate for processes. The calculation is based on actual production counts divided by unobstructed scheduled runtime. For example a sample process is scheduled to run 9.067 hours in a shift, it produces 400 parts, is starved 1 hour and blocked .067 hours. The unobstructed production time is 8 hours therefore the unobstructed production rate is 50 jobs per hour. Multiplying this rate by the straight time schedule of 18.1333 hours yields a standalone capacity of 907 jobs per day (the process' standalone capacity). There are some key points concerning this approach:

- Process downtime is not directly included in the calculation. For many processes downtime data was / is not available because PLC logic for calculating downtime requires much more effort than measuring production counts and starving / blocking times. Therefore any process that has these measurements are readily included in the calculation.
- The use of standalone capacity normalizes for the different relief plans, 18.1333 hours per day on mass relief processes and 20 hours for tag relief processes.
- Bottlenecks are identified by that process with the minimum standalone capacity.
- Measurements are based on straight time performance only. Although data concerning overtime performance is logged it is not used in the calculation.
- All reporting is done in vehicle equivalence. For example if there are 70 automatic transmission clutch hubs produced, SSPF reports 100 vehicles produced (assuming a 70% automatic transmission penetration). In the case where there are 10 stampings required for each vehicle then the production of 1000 stampings will be reported as 100 vehicles produced.

SSPF is based on an evolving technology called Model-Integrated Computing (MIC)[3][5][6][7]. Configuring SSPF to a plant involves using a graphical tool to describe processes / banks and the relationships to other processes / banks. Properties are described in the model such as point names for starving, blocking, production counts, and any other optional data that would be useful (facility for collecting downtime or CMORE data if available). This model building tool is targeted for manufacturing engineer's usage (although all models to date have been built by Vanderbilt graduate students). There is an automatic process that converts these models into configuration files used by the runtime system (see **Error! Reference source not found.**).

With the configuration files in place, SSPF *automatically* does the following:

- Establishes communication with the Cimplicity project containing the required data (from the properties set in the model).

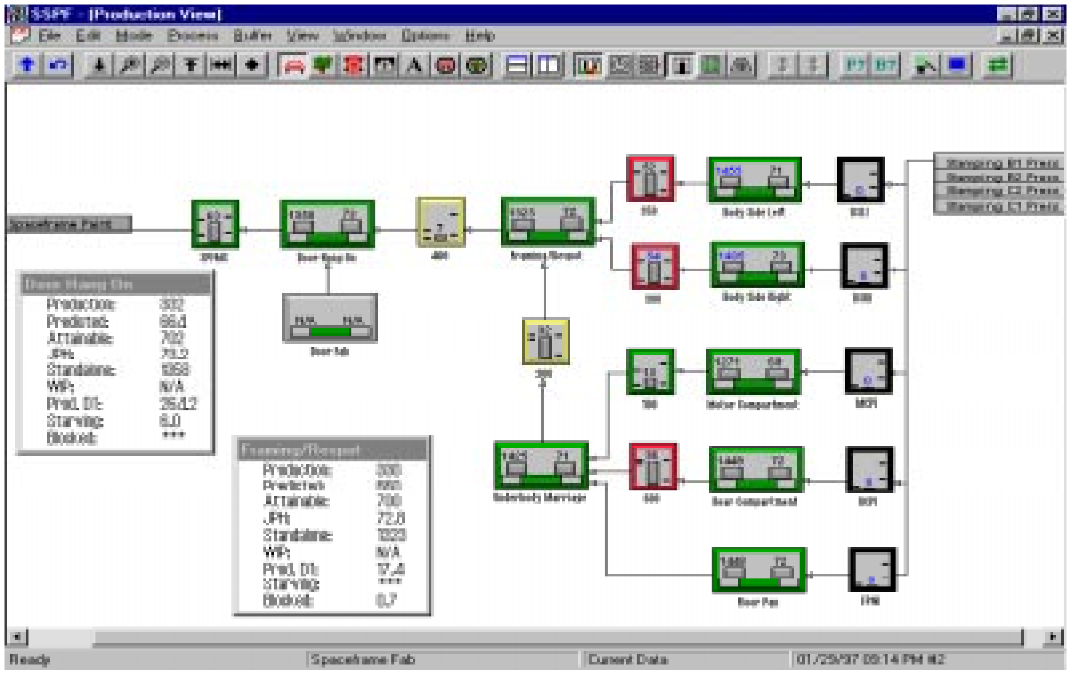

Figure 2: SSPF Viewer

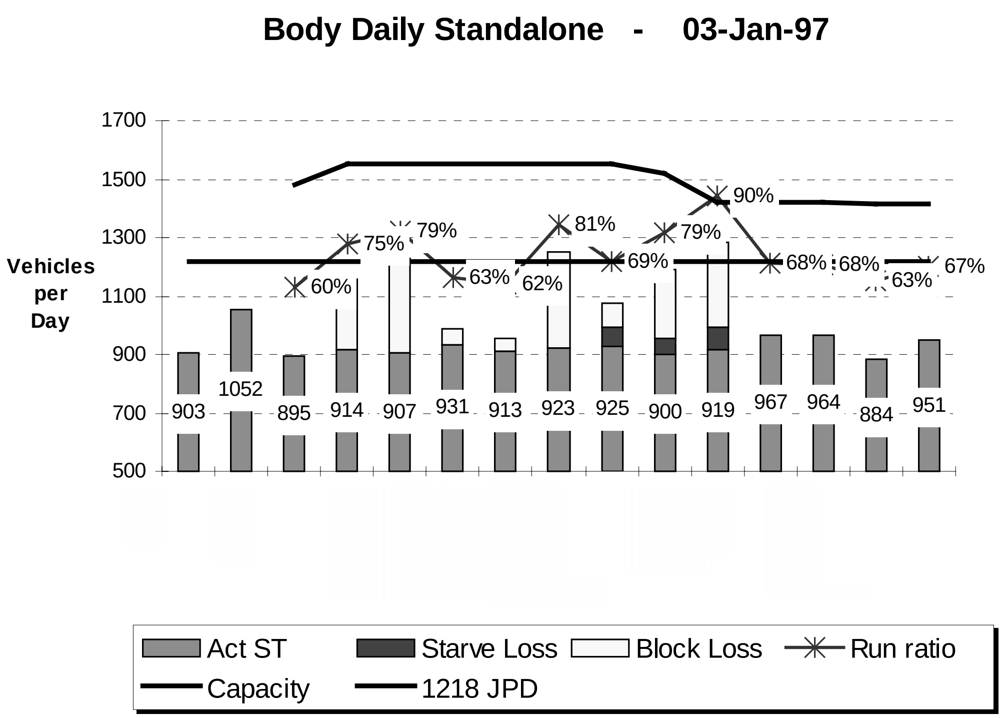

Figure 3: Body Daily Standalone Chart

- Logs into a database all changes in all points and calculated values for all processes in the model.
- Logs into a database all point and calculated values for all processes in the model every half hour and at the beginning and end of breaks.
- Logs summary of all process values at the end of shift and end of week.
- Configures a graphical viewer (see **Error! Reference source not found.**) that is available to all team members showing real time and historical values.
- Configures a data browser that allows the user to select specific values (real time and historical) for specific processes and link those data elements into an Excel spreadsheet.
- All tables in the database are available for analysis using Access and Excel.

The type of information available to the end user is real time and historical. There is a viewer that is available to all Saturn team members that displays current real time data as well as user selected historical data. Key information concerning standalone capacity, starving downtime, blocking downtime, throughput rates, etc. are displayed graphically, with pop up windows, and text views. Processes are organized hierarchically allowing the user to select more detail or aggregate to a higher level through standard point and click maneuvers

Through the use of COTS tools such as Access and Excel, databases in SSPF are summarized in reports that are reported to management and made available to all team members (via Saturn's intranet) on a daily, weekly, monthly, etc. basis (e.g., see **Error! Reference source not found.**). These charts are used as part of a business process to perform continuous improvement on bottleneck processes to achieve overall vehicle throughput.

2. Accomplishments

The production development and implementation of SSPF has led to numerous accomplishments related to the business process for which the system was developed and technical areas of systems engineering:

- Throughput Improvement Contribution
- Business Need Responsiveness
- Organization Acceptance
- System Robustness
- Scaleability
- Inexpensive configuration

Throughput Improvement Contribution

Shortly after the initial production installation, a throughput task force was sanctioned to implement improvements in throughput. Although heavily focused on business processes a key tool was the use of SSPF for data collection, analysis, and reporting. This was a process for the organization to learn and required that SSPF be quite adaptive to task force requirements. This iterative process led to a significant improvement in plant throughput within a period of 6 months meeting specific goals driven by the marketplace.

The drivers for throughput improvement range from technical improvements to leadership methods, but part of the drivers are based on usage of SSPF. With pertinent, accurate data available this allows the team to focus on problem fixes. Often business problems of this nature consume enormous resources and effort in measuring rather than fixing. Very early in the effort, all questions concerning validity of information were answered and the team moved on to resolution of issues. As the task force progressed in resolving issues the rapid response of SSPF analysis to these new business understandings allowed for an iterative process of learn, measure, fix then learn, measure, fix. The low effort in supporting requirements assured that the majority of resources were focused on problem resolution. The further benefit of having low resource requirements provided the further benefit of avoiding the delays inherent in resource procurement thus allowing the task force to move directly to problem resolution.

Business Need Responsiveness

Since the initial release of SSPF there has been continuing changes in business priorities. As issues in one area were resolved then other issues have become more prevalent.. As new issues are identified SSPF has been quickly and responsively modified. Modifications to business processes and needs for information vary. As business processes change the utility provided in SSPF provides for changes in models and model paradigms to support these business process changes as well. Following the successful completion of the throughput task force mission the Powertrain organization has shifted toward using SSPF as a key analytical tool. This has required development and testing of models with frequent modifications as changes have been made. Another example is the collection of CMORE (a custom throughput simulation program) data from the body shop as a decision tool for the body shop expansion project. Additionally passing data from SSPF to other applications such as crew leader reporting are being implemented. These examples demonstrate the versatility of MIC in meeting the dynamics of a manufacturing business environment.

Work is now beginning to apply SSPF into another General Motors plant. Although this work is just beginning the reusability has provided rapid deployment. SSPF was installed in ½ day with a basic set of models with a rudimentary level of reporting provided the next day. There was no modification to the system except for models specific to the new plant. This deployment is in the pilot stage where issues concerning multiple plant support, modeling paradigm validity, and business practice consistencies are being addressed. Pilot results indicate that with little additional effort deployment and support for multiple plants will be readily manageable.

Organization Acceptance

Critical acceptance in key parts of business has been achieved. No system is successful unless it is used to meet the intended as well as expanding purpose. SSPF has been accepted as a key tool in various parts of the business. Summary reports are presented at the morning miniMAC (the sitewide manufacturing operations meeting) each day. Each of the business teams utilizes various summary data as part of it's business decision process. Other applications are using data from SSPF as part of other applications. Training of real time users is in progress.

System Robustness

SSPF over the first year of production use has demonstrated to be extremely robust. When compared to other Manufacturing Execution Systems that were developed and released at Saturn during the same time period SSPF was found to be one of the most reliable applications in operation. The reason for the robustness is the same as for relative inexpensive configuration (the point below) – use of models to generate the system. One can think of adding more models as adding functionality in a traditional solution. Of course, in a traditional solution, when you add functionality to software, you introduce bugs and degrade robustness (e.g., using Cimplicity, this would have required building new screens, connections, etc., each with its own set of bugs). If you use MIC instead, adding new models does not introduce any new bugs in the software (though it may bring out hitherto unknown bugs).

Scaleability

Since the SSPF application space was known to be one that would be dynamic and changing, in the beginning SSPF was *designed to be highly scaleable* allowing for various services to be implemented on different pieces of hardware. Another scaleable feature is the concept of having multiple real time data handlers (RTDH). Currently all services are running on a single server (except for client viewers) and there is a single RTDH but this is true only because there is no need to fully utilize SSPF's scaleability.

Inexpensive configuration

It has been demonstrated that it is a relatively inexpensive configuration process through use of models. Through the use of models the actual "programming" of the plant detail has been of relatively low effort and the modeling process has been highly responsive to business needs. One of the best examples was the creation of a set of models that was used to collect CMORE data for the new body shop design. This was accomplished in a period of a few hours. A traditional systems design and implementation effort would have required several weeks of effort as well as procurement of additional hardware with a significant lead time in implementation. Using SSPF model paradigm lead time was a matter of days.

The result is that as changes have been required in the business of building cars there has been orders of magnitude less effort. Due to this very low effort that has been a tendency to provide more utility than to focus on reducing the cost of providing information. This is as it should be; the goal is to improve production throughput and not necessarily to provide the *same information* at lower cost. The implications on costs though are extremely significant. We are finding that as time goes by the system life cycle costs are running much less than if the same utility were provided using traditional techniques. Modeling changes are frequent and significant but the system costs are typical of a system that is in the maintenance phase of its system life cycle.

3. Challenges

These accomplishments are significant, real, and very beneficial but there are challenges that must be addressed. Some of these challenges are business process in nature and others are systems engineering issues. These challenges represent tasks that will be either addressed as part of future work on SSPF, as part of work by others in the MGA Alliance, or have no current plans for being addressed:

- Decision support development
- MGA cost impact analysis
- Relaxing process hierarchy restrictions
- Manual transposition of models to new paradigm
- Freezing of models while constructing new paradigm
- Articulating the niche for SSPF
- Trending throughput information
- Development of internal expertise
- Domain extension for MIC

Decision support development

Current development work is focused on providing domain specific analytical tools. A key conceptual feature of SSPF is decision support, specifically deterministic simulation and bottleneck analysis. These are features that truly distinguish SSPF from traditional PM&C as well as other simulation tools. Being domain specific and based on the manufacturing process specific models, potential for insightful analysis appears to be highly promising.

Deterministic simulation is an analytical tool that allows the user to specify the results of a planned fix. Parameters such as mean time between failure (MTBF), mean time to repair (MTTR), bank sizes, etc. can be modified. These parameter changes are applied to actual historical data to determine what the change in net throughput is and where future resulting bottlenecks would be. A key feature of SSPF as an analytical tool is to provide a reliable low cost prediction of changes in throughput for proposed changes in the manufacturing process.

MGA cost impact analysis

Cost savings through the application of MIC have not been demonstrated. A key justification for using MIC is that there is a significant reduction in cost for delivery of such a system. A cost analysis is required to support that MIC is the low cost solution but no such hard numbers currently exist.

An approach that will be pursued is to perform a cost analysis on the development of SSPF and through experience and engineering estimates determine what it would have cost Saturn to develop something like this using Cimplicity, C++, SQL/Server, BackOffice, Excel, etc. Comparisons should be made for :

- Original scope of SSPF
- Extended scope and functionality.
- Versions
- Changes in models (equivalent to added functionality in traditional software)
- Maintenance, etc.
- Other factors……

The SSPF cost for SATURN cannot be separated from life-cycle considerations. The total SSPF cost incurred to date includes the full development cost (which was of course leveraged by the MGA tools). The cost saving could be made obvious by looking at the cost of changing, extending the installed and operational system through the modeling/system generation mechanism.

Relaxing process hierarchy restrictions

Process hierarchy has a level of *inflexibility* that restricts application. In the spirit of inheritance there are rules in the modeling paradigm that restrict the expressiveness of the models that are built. Specifically that very top level has been found to be less than process focused (processes are actually organizations rather than processes). This top level cannot be changed without a significant design effort. There needs to be methods developed that will allow the hierarchy to be changed without creating major disruptions in related features of the system. The design of SSPF 2.0 is such that one can do away with the whole hierarchy if needed and have just *sets* of processes, buffers, etc. (though the current implementation of run-time components assumes a hierarchy, changing it to non-hierarchical form would require only minor changes in the runtime components and models). Further study on the impact of making changes to the hierarchy will need to be studied and understood before changes are made.

Manual transposition of models to new paradigm

During the first 3 months of SSPF production use, it became obvious that significant changes to the modeling paradigm were required. Through use there is learning that not only influence (requiring change) the models but at a meta level there is a learning that requires change to the paradigm. Three months of effort was required to *manually reconstruct and test the models from the old paradigm to the new paradigm*. This is a significant shortcoming to the MGA since a truly dynamic system will see occasional and regular changes in the modeling paradigm. We expect a drastic change in this with the new generation DARPA tools. This experience just outlines the need for a model elaborate set of meta-level tools in MGA.

Freezing of models while constructing new paradigm

Logistically it became a requirement to *freeze models* while translation was in progress. Because there was a need to manually translate models from the old paradigm to the new paradigm, a freeze on all models was required for 3 months. This effectively meant that we could not implement what the organization was learning and severely hampered the utility of SSPF during that time period (a key utility of SSPF is to respond to current business needs). It would be expected that any MIC application that is part of a dynamic business process would encounter the same problem. Since the application of MIC is toward these types of problems, this implementation issue becomes a significant issue requiring resolution.

Articulating the niche for SSPF

As a manufacturing execution system, SSPF represent a new, emerging category of manufacturing information systems. One of the more difficult business design issues has been exactly how SSPF fits in the overall suite of manufacturing execution systems at Saturn. There is a PM&C system that gathers data and reports status of processes on the manufacturing floor. There are custom database applications for logging and reporting of downtime and production data. There is a data warehouse application under development that integrates all manufacturing data. There are simulation systems in place to provide decision support. Although there is more work required in articulating a vision of where SSPF fits in this mix, it can be stated that SSPF has features of all these other applications but focused on the business issue of improving throughput. SSPF performs to some degree PM&C, database logging and reporting, data warehousing, and decision support in an integrated contiguous application.

PM&C systems, simulators and data warehousing systems are usually not considered manufacturing execution systems. MES is a glue between the PM&C functions and the classical back-office systems. SSPF is a clear-cut case of manufacturing execution systems, and as such, it has links, interfaces and at least some function overlaps with the other systems. Being a quite new system category, the industry is just absorbing the business role and boundaries for MES. The unique characteristics of these systems are clear though: PM&C systems do not have a plant-wide perspective, they are usually localized to specific manufacturing processes, MESs do have a plant-wide perspective and use a lot of information about the overall plant. Back-office systems are typically off-line and are not linked to live process data while an MES is a real-time system.

Overall there is a significant challenge in articulating how SSPF fits in EIS. The SSPF application domain concerns throughput, a narrowed focus of the overall manufacturing domain. How does SSPF differ? First it uses a subset of overall data in the domain. Therefore this provides a focus on this information to achieve a very specific goal. SSPF coexists with other systems in the domain, which are under an evolution using varying philosophies and technologies. Cimplicity© is focused on status while CMORE is intended to perform a detailed analysis. Data warehouse provides for data availability via efficient utilities. The challenge is to fit SSPF into this collection of solutions. Also there is the question of fit with respect to rest of General Motors. Use of common systems for all of General Motors is a key initiative that includes common business processes. In performing this analysis a key question is what are the alternatives to SSPF and how does SSPF stack up to those alternatives?

Trending throughput information

A significant feature that has had some progress but still requires work is the ability to provide trend information. Provision of meaningful tools and support has been exacerbated in the effective utilization of COTS and related systems. There has been progress by providing OLE interfaces and summarization tables but the goal of being able to readily provide trend information “on demand” has not been achieved. It appears that extending the modeling paradigm to encompass summary reporting would make sense. Further development in this area is in progress.

Development of internal expertise

A frequent issue that occurs with any contracted solution is the *lack of internal expertise* concerning the system provided. This is also true with SSPF. Saturn's intent is to be fully capable of supporting SSPF but as a practical method of resource management a high dependency on using Vanderbilt continues. There is more than a risk of having critical applications being externally supported; there is the risk of not learning as an organization. Without that knowledge being internalized other applications will not have that competitive benefits of applying model integrated computing.

Domain extension for MIC

Another class of challenges is the application of MIC to other application spaces within Saturn. These are being considered with only initial evaluations commencing.

- As in any organization *quality* is a key success factor therefore a major business issue requiring system development. Development activities that are currently underway are focused on more systematically dealing with (internal to) manufacturing data. Future initiatives are expected to be focused on integrating internal quality measures with quality measures as seen by customers. At this point plans call for traditional development techniques to be utilized. There is a potential fit of MIC to this problem in having models describe the relationships between all the measures, provide comprehensive reporting of this information (from the ultimate customer's perspective), and maintain relationships among the various measures.
- *Data warehousing* is a strategic application under development within Saturn. The premise of data warehousing is that there are related information flows from various sources and that there are efficiencies in applying broadbased utilities for managing such information. There is a potential fit for MIC in the application of models that map users of information in the enterprise to the various sources of information.
- *Data mining* is a new conceptual initiative within Saturn that has the potential of being a MIC application. It would appear that MIC models could be utilized to express the type of relationships that should be mined.

ACKNOWLEGMENT

This work has been supported in part by the SATURN Corporation and DARPA/ITO EDCS program, Contract#F30602-96-2-0227

BIBLIOGRAPHY

[1] Sztipanovits, J., Karsai, G., Biegl, C., Bapty, T., Ledeczi, A., Misra, A.: "Multigraph: An Architecture for Model-Integrated Computing", *Proc. Of the IEEE ICECCS'95*, pp. 361-368, Florida, 1995.

[2] Long, E. and Misra, A., "Saturn Site Production Flow: Requirements, Constraints, Issues," 1996 Workshop on Model-Integrated Computing, Old Hickory, TN, June 3-4, 1996.

[3] Misra, A. et al., "A Model-Integrated Information System for Increasing Throughput in Discrete Manufacturing," International Conference and Workshop on Engineering of Computer Based Systems, Monterey, CA, March 24-28, 1997.

[4] Long, E. and Misra, A., "A Model-Based Engineering Process for Increasing Productivity in Discrete Manufacturing," International Conference and Workshop on Engineering of Computer Based Systems, Monterey, CA, March 24-28, 1997.

[5] Misra, A., Karsai, G. and Sztipanovits, J., "Model-Integrated Development of Complex Applications," Proceedings of the Fifth International Symposium on Assessment of Software Tools, Pittsburgh, PA, June 2-5, 1997.

[6] Karsai, G., Misra, A., Sztipanovits, J., Ledeczi, A., and Moore, M., "Model-Integrated System Development: Models, Architecture and Process," Proceedings of the 21st Annual International Computer Software and Application Conference (COMPSAC 97), Bethesda, Maryland, August 13-15, 1997.

[7] Misra, A., Long, E., Sztipanovits, J.: "Evolutionary Design for Manufacturing Execution Systems", *Proc. Of the Manufacturing World Congress 1997*, November 21-24, Auckland, New Zealand, 1997.