**Proceedings of DETC'01**
**ASME 2001 Design Engineering Technical Conferences and Computers and Information in Engineering Conference**
**Pittsburgh, Pennsylvania, September 9-12, 2001**

# DETC2001/CIE-21281

# AFFORDABLE SPACE SYSTEMS MANUFACTURING: INTELLIGENT SYNTHESIS TECHNOLOGY, PROCESS PLANNING, AND PRODUCTION SCHEDULING

Jeffrey W. Herrmann
Institute for Systems Research
University of Maryland
College Park, Maryland 20742 USA
301-405-5433. jwh2@eng.umd.edu

Mark Fleischer
Institute for Systems Research
University of Maryland
College Park, Maryland 20742 USA
301-405-6875. mfleisch@isr.umd.edu

Edward Lin
Institute for Systems Research
University of Maryland
College Park, Maryland 20742
USA
301-405-6571
lin@isr.umd.edu

Vidit Mathur
Institute for Systems Research
University of Maryland
College Park, Maryland 20742
USA
301-405-6572
viditm@isr.umd.edu

Jim Glasser
Advanced Technology Center
Lockheed Martin Space Systems
Sunnyvale, California 94088
USA
408-742-5694
jim.glasser@lmco.com

## ABSTRACT

Low volume production associated with space systems manufacturing is inherently expensive, time-consuming, and risk-laden. At the root of this problem is an inability to adequately predict, monitor, and control the product development and sustainment process. This paper describes the ASSIST system, an intelligent knowledge management system designed to address inefficient information management processes and improve space system affordability. ASSIST is designed for collaborative engineering, manufacturing, and testing within a company, between companies, and between local and remote individuals. ASSIST uses web-based standards (including HTTP and XML) as the common message approach connecting its components. Automated process planning and production scheduling is a key component of ASSIST. This paper describes an integrated process planning and production scheduling problem and discusses the solution approach.

**Keywords:** Internet based design and manufacturing.

## 1. INTRODUCTION

Low volume production associated with space systems manufacturing is inherently expensive, time consuming and risk laden. Typical characteristics of low volume manufacturing, such as lack of standard design and common parts, frequent changes to design, costly test environments, and loosely coupled product teams, including suppliers, are obstacles to achieving space systems affordability. At the root of this problem is an inability to adequately predict, monitor, and control the product development and sustainment process. This translates to an inability to have effective Integrated Product and Process Development which, as identified by the Department of Defense's Lean Aerospace Initiative (LAI), is essential for manufacturing affordability.

A team led by Lockheed Martin Space Systems Co. Missiles and Space Operations (LMSSC-MSO) is providing the solution: the Affordable Space Systems Intelligent Synthesis Technology (ASSIST) for Manufacturing program. ASSIST is an intelligent knowledge management system that dramatically reduces cost and cycle time in low volume space system manufacturing. This team also includes NexPrise, Inc., Orbital Network Engineering, Inc., Nabh Information Systems Inc., and the University of Maryland. The ASSIST Program is a 43-month program sponsored by the Air Force Research Laboratory, Materials and Manufacturing Division, as a Technology Investment Agreement under the Manufacturing Technology for Affordable Space Systems (MASS) initiative.

The goals of the ASSIST program are a reduction of 50% in subsystem design man hours, 15% in procurement cost, 50% in launch site support hours, and 50% in particular pre-launch testing regimens. The ASSIST System will be demonstrated in a series of validation pilots involving LMSS-Missiles & Space

Operations (MSO) satellite propulsion subsystems. These reductions will be made possible by the five components of ASSIST: electronic collaboration, electronic procurement, virtual product modeling & prototyping, automated process planning and production scheduling, and automated test operations management. All components will be available to the user through a single ASSIST graphical user interface (GUI).

The principal objective of ASSIST is process and enterprise affordability improvements within the space manufacturing enterprise that will reduce cost and cycle time, while demonstrating improvements in quality for current or planned Air Force spacecraft subsystems. ASSIST addresses such problems as the lack of interoperability between current information management tools used in manufacturing, which yield poor access to data and inadequate team integration. Inadequate team integration leads to isolated, redundant, costly and time consuming efforts such as design changes after tooling has been produced, redundant design efforts for similar elements, and manual testing, with little traceability to requirements or design changes.

Automated process planning and production scheduling is a key component of ASSIST. ASSIST includes a process planning and production scheduling application (called IPPS) that is flexible and general enough to handle complex manufacturing systems such as those used to manufacture space systems. Space system manufacturing consists of many complex, labor-intensive tasks. Assembly, integration, and testing require skilled personnel, expensive equipment, and specialized facilities. Although the production quantity is low, the production complexity is high. Dramatically reducing the manufacturing cost and time requires utilizing the available resources effectively. In such complex manufacturing systems, a large number of equipment and human resources must work together harmoniously to meet production goals efficiently. Given a set of tasks to complete within a given time horizon, production planning and scheduling activities must assign individual resources, or combinations of resources, to specific tasks and decide when the tasks should begin. Resources include trained personnel, machines, infrastructure, and components from inventory. In some cases, the particular combination of resources applied to the task affects the amount of time that the task requires. This type of resource allocation and scheduling problem is encountered in many manufacturing systems where personnel have a variety of skills and machines can be assigned to several different types of tasks.

In this setting process planning denotes using information about a specific product design and resource availability to determine which tasks should be done and how they should be done. Production scheduling refers to assigning specific resources to each task and determining a task start time. Thus, some portion of process planning must be done dynamically, since changes in resource availability (which can occur unexpectedly) may require finding a different resource combination to perform the task.

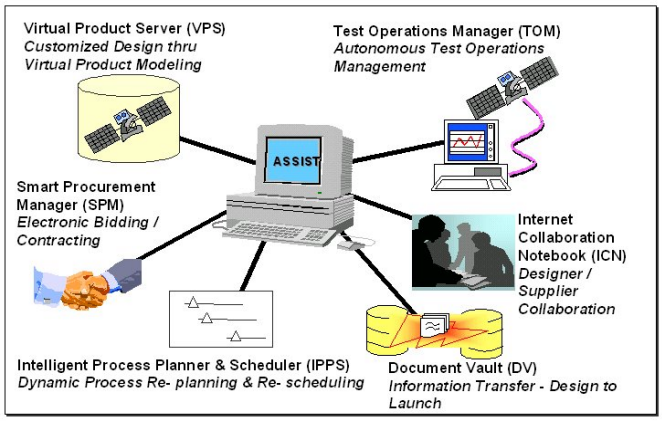


Figure 1. The applications that ASSIST integrates.

The remainder of this paper proceeds as follows. Section 2 describes the intelligent synthesis technology that is the architecture of ASSIST. Section 3 discusses the process planning and production schedule component (IPPS). Section 4 concludes the paper.

## 2. INTELLIGENT SYNTHESIS TECHNOLOGY

The ASSIST approach leverages collaborative and information management technologies that have been successfully demonstrated in stand-alone environments, and integrates them in an innovative and extendible infrastructure (see Fig. 1). The resulting environment, which provides a seamless electronic communication management system, overcomes enterprise-wide manufacturing cost and time consumption due to limited availability and inefficient use of information. ASSIST facilitates practices which improve team dynamics through all phases of the space systems acquisition cycle. This is accomplished by managing technical, cost and schedule information so that all project participants are supported in, *and guided to*, more efficient practices.

The technical approach being used to develop the ASSIST system consists of:

- defining the objective of ASSIST
- defining requirements for the system based on the business case for the LMSS-MSO standard Propulsion Subsystem (PSS) (which is the first testbed for the technology)
- gathering user-defined requirements
- analyzing current “as-is” processes
- identifying and evaluationg commercial off-the-shelf (COTS) products and other leveraged technologies that will satisfy the requirements
- adhering to software industry standards
- defining the ASSIST architecture
- developing prototype implementations of the architecture
- testing prototype modules
- providing modules to users
- refining the ASSIST system

♦ validating the technologies through validation pilots

During the requirements gathering phase, the "as-is" design, procurement, manufacturing/assembly, testing and launch operations processes and associated product and data models of the PSS were examined. Interactions and coordination with suppliers were also included in the process analysis, since ASSIST will impact processes from all phases of the satellite PSS development, i.e., from design, to procurement, to manufacturing/assembly, to testing and launch operations.

In addition to user requirements, issues related to scalability, portability, and utilization of standards are addressed in the approach. The ASSIST development team is also following the standard LMSSC/MSO software development process. As part of that process, they have adopted a spiral software development approach.

The development approach is based on a spiral development model, where the spiral process cycles through design, development, delivery, and evaluation phases several times. Spiral development supports rapid prototyping in environments such as ASSIST, where developers are tasked with evolving an architecture as the technology changes. LMSSC-MSO is building an integrated system of disparate parts, which include legacy, commercial, and internal systems.

A series of ASSIST validation pilots will be the formal assessments of the impact of the ASSIST technology on the PSS processes and will provide feedback on the system, as applied to the production environment by the actual end-users. This feedback will also be used in the refinement of the ASSIST system as the program progresses through development spirals, and as input for the market and business plans for the commercialization feasibility of the technology.

There are three views that are used to develop and explain the ASSIST architecture: the operational view, the system view, and the technical view. This breakdown into three views is borrowed from the US Defense Department's C4ISR requirements for describing architectures. The operational view is an operational picture of the use, and provides background on *why* a particular architecture decision was made. The system view identifies *what* components are in the architecture, and their interrelationships. The technical view relates *how* the architecture will be built, identifying the technical standards and methods used in the architecture.

## 2.1 OPERATIONAL VIEW

The operational view is an operational picture of the intended use for an architecture and provides background on *why* a particular architecture decision was made. Operationally, the ASSIST system is intended to be used for collaborative engineering and manufacturing/testing within a company, between companies, and between local and remote individuals. In Fig. 2, the internal users have been identified, as well as traveling users, and external users (suppliers). These users must be able to obtain and update information while external to their company system and must be able to communicate that information with external companies and individuals. To operate on this information, the users require a variety of legacy, commercial, and internal tools. Thus the primary requirement for ASSIST is to provide an integrated view and use of the different ASSIST information and tools (whether commercial, legacy, or internal).

Table 1 lists the operational architectural requirements that have been distilled for application to the ASSIST architecture. These include availability requirements, integration requirements, and security requirements.

## 2.2 SYSTEM VIEW

The system view identifies *what* functional components are in the architecture, and their interrelationships. ASSIST has a component-based, client-server architecture (see Fig. 3), and there are two categories of components: infrastructure modules and application modules. Infrastructure modules are used as a framework for the application modules, connecting people, tools, and information into a process. Application modules are focused on supporting a particular task, such as automated testing of satellite propulsion or maintaining an engineering notebook. IPPS is one such module. The other modules include the workflow agent (WORK), the notification agent

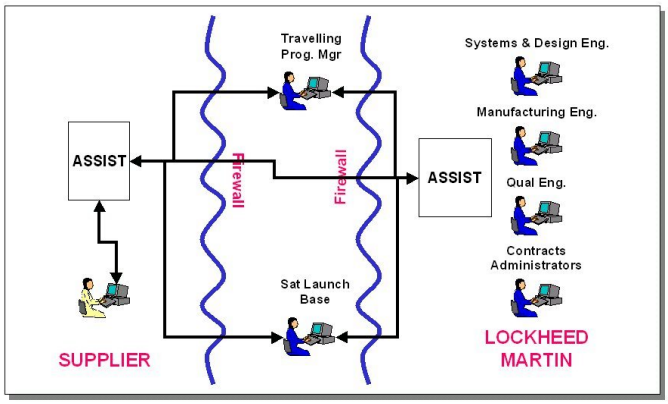


Figure 2. The ASSIST operational view.

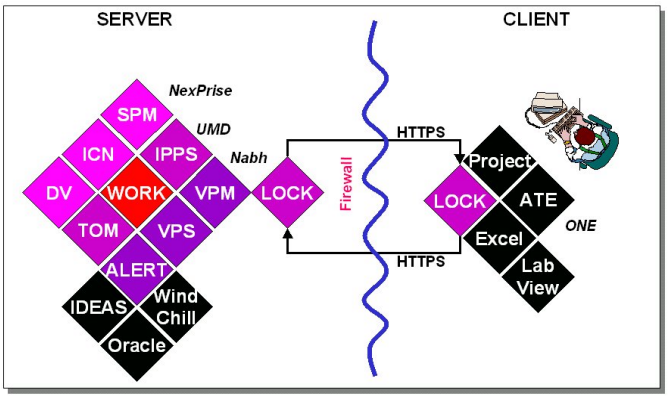


Figure 3. The ASSIST system view.

Table 1. The ASSIST operational architectural requirements.

| **Availability Requirements** |
|---|
| Provide a Web front end to the ASSIST system |
| Need to interact with supplier and other systems across firewalls |
| Need for the ASSIST system/user interfaces to interact with multiple servers |
| Need to interact with legacy systems |
| Business logic independence from "plumbing" details and user interface |
| Scalability for large amounts of users, data, and projects |
| High availability and easy fail recovery |
| **Integration Requirements** |
| Eliminate or reduce vendor-lock in for COTS tools |
| Need to interact with common COTS tools (Excel, IDEAS, Microsoft Project, etc.) |
| Need to interact with various databases (Oracle, Sybase, Informix, etc.) |
| Need to support transactions that span across various systems (i.e., provide workflow capabilities) |
| Need for tools that help set up process / data flow |
| Leverage investments in legacy code |
| **Security Requirements** |
| Ability to locate components and execute them securely (i.e., a Directory Service) |
| Authentication of users and servers (e.g. PKI, CA, two-way authentication) |
| Provide secure communications (e.g. HTTPS, IIOP-SSL) |
| Collaboration services (Workflow, Notification) |

(ALERT), the virtual product server (VPS), security (LOCK), the test operations manager (TOM), the document vault (DV), the Internet collaboration notebook (ICN), and the smart procurement manager (SPM).

ASSIST's client-server approach has several system implications. The first is that an emphasis on a web-based interface to system components without internal heavy applications that require long downloads or unusual plugins. At the same time, the system must support real engineering and manufacturing operations such as propulsion testing as well as supporting client applications such as LabView, and Microsoft Excel and Project.

The system view figure (Fig. 3) shows a cluster of client tools talking with the server-side modules using the ASSIST framework. Note that there is a need to cross the firewalls between companies and a need to support client interfaces to server-based tools. Both of these imply that the ASSIST system should adhere to industry standards and use commercial off-the-shelf (COTS) software wherever possible.

### 2.2 TECHNICAL VIEW

The technical view relates *how* the architecture will be built, identifying the technical standards and approaches used in the architecture, as well as the reasons for adopting the standards. This section describes the capabilities desired, then relates qualities for industry standards, and finally discusses a tiered architecture that enables these capabilities and qualities.

The ASSIST system is to incorporate previously developed collaboration and special COTS technologies. Some of these technologies exist as a part of their own framework (e.g. the ASSIST DV module is the incorporation of the ipTeam iVault COTS product which operates within the ipTeam framework). In addition, there are several legacy software and hardware tools that users would access during its operation. Users, however, do not want a fragmented view of the ASSIST system nor should they need to interact with multiple frameworks and user interfaces. Thus a "component-based approach" that satisfies both these potentially conflicting requirements is being used. This approach has four key features, which the following paragraphs describe.

**Encapsulation of individual products as "Components."** Components are objects that communicate via a standardized remote communication protocol. The ASSIST modules are a type of component. Given the mix of computational platforms identified in the operational view, ASSIST uses web-based standards (HTTP – hypertext transport protocol) and XML (eXtensible Markup Language) as the common message approach connecting our components. Once the application modules have been encapsulated as components, the approach then consists of building customized graphical user interfaces that can access multiple ASSIST components and yet present an integrated, user-friendly view of the underlying information base.

**Separation of Business Logic and "ilities."** ASSIST provides a clean separation between the actual functionality and "ility" issues such as scalability and security. This will allow deployment to the same business object in different execution context without any re-implementation of the logic.

**Single-Sign-On.** We plan a centralized user authentication mechanism that is used by all ASSIST infrastructure modules. This will eliminate the need to log on separately for each of the ASSIST components.

**Dual User Interface.** ASSIST users fall into two categories. The first and smaller category of users includes process owners who configure the information flow within the ASSIST environment. Typically these users need a "heavy-weight" user interface and would typically interact with the system from one desktop. Thus ASSIST will use Java applications (as opposed to applets or HTML interfaces) that provide such user interfaces. The other category of everyday ASSIST users include managers, engineers, administrators that need lightweight user interface and access from wherever they are located. ASSIST will include a set of web-based user interfaces for this class of users.

## 3. PROCESS PLANNING AND SCHEDULING

As described in Section 1, planning and coordinating production in complex manufacturing systems such as those used to manufacture space systems becomes a resource

 

allocation and scheduling problem. Modeling such a problem can require a large number of discrete variables (for task assignments), continuous variables (for task start times), and nonlinear constraints. Many special cases of the problem form interesting machine scheduling problems, and many workers have studied these and proposed solution techniques. For an overview of scheduling results and research, see Pinedo [1]. This research has studied parallel machine problems, flow shop problems, and project scheduling problems. The special structure of these problems reduces the problem formulation and the solution space.

As manufacturing systems become more complex, however, the limitations of these special cases cause problems. Thus, it is necessary to consider a general formulation that includes *resource combinations* and the characteristic that a task's duration depends upon the particular combination performing the task. The generalized problem is more flexible and can address a broader range of manufacturing systems. IPPS formulates a very general model and employs an approach that can harmoniously incorporate the discrete and continuous variables that are necessary.

### 3.1 IPPS OVERVIEW

This section gives an overview of IPPS by describing the typical actions that a user would complete to plan and coordinate production. IPPS has seven modules, through which the user can perform various IPPS functions (see Fig. 4). Each module is designed to organize a set of functions. Through the functions of IPPS, the user can perform the following types of actions:

- View and edit information needed for scheduling
- Construct a scheduling problem
- Solve the scheduling problem
- View and edit a schedule
- Publish a schedule

To begin, the user updates information about the orders, materials, and resources in the factory. When the user adds a new order, IPPS locates critical design information about the specific system that has been ordered. This specific system will be a customized version of a certain model. A process template describes the generic set of tasks needed to produce one unit of this model and rules used to update the process template to create a process plan for the specific system ordered. The process plan specifies the set of required tasks, the task precedence constraints, all feasible resource combinations for each task, and the task duration for each feasible resource combination. A resource combination is one possible method for performing the task. It identifies the specific types of employees and equipment needed for that method and how many that method requires. Different resource combinations for the same task specify different types of employees and equipment. Note that the factory may have many employees or

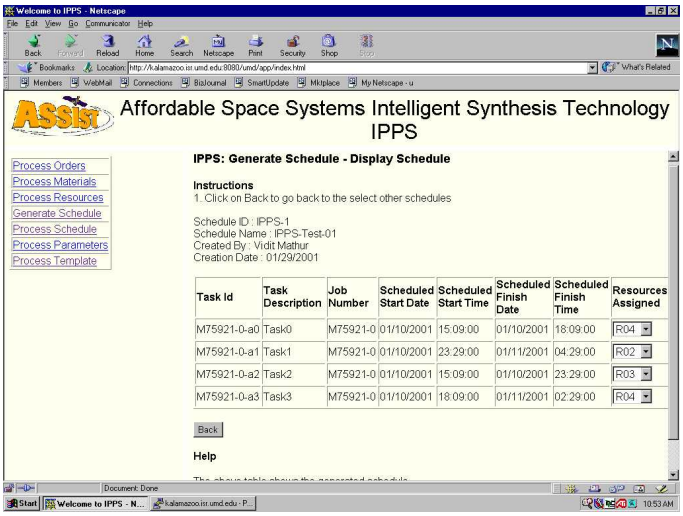

Figure 4. The IPPS user interface.

tools of the same type. Thus, IPPS has to determine which method (resource combination) should be used to perform a task (a process planning activity) and which individual employees and tools will perform the task (a scheduling activity). Thus IPPS integrates process planning and production scheduling.

IPPS first uses the critical design information and the process template to create the process plan. This process plan is still incomplete since some tasks may have multiple feasible resource combinations. The schedule optimization engine will complete the process plan based on resource availability at the same time it assigns specific resources (employees and equipment) to the task.

After the needed information has been updated, the user may release one or more new orders that need to be scheduled. If no additional orders are released, the new schedule will include the active and planned tasks in the current published schedule. If orders are selected for release, the user will be asked to confirm the release, and the tasks associated with these orders will be included in the new schedule.

Then, the user may add preferences about specific tasks by specifying any of the following information about the task: earliest start date, earliest start time, earliest completion date, earliest completion time, latest start date, latest start time, latest completion date, and latest completion time. The user may force one or more specific resources to perform the task or prohibit such an assignment.

When ready, the user asks IPPS to optimize the schedule. Then, IPPS reads all of the necessary information, generates an instance of the scheduling problem, and calls the optimization algorithm to find a good schedule. (Section 3.2 gives the problem formulation, and the solution approach is described in Section 3.4.) When the optimization algorithm is finished, the user can view and edit the new schedule and may publish the new schedule. The problem size will vary based on the number of system in process and the number of tasks required for each

system. For the satellite propulsion subsystem factory, a typical scheduling problem will have fewer than 10 systems and approximately 35 tasks per system.

### 3.2 PROBLEM FORMULATION AND RELATED WORK

This section describes problem PSP, the integrated process planning and production scheduling problem that IPPS generates. There exist a finite set $T$ of tasks, $T = \{1, \ldots, n\}$, and a finite set $R$ of resources, $R = \{1, \ldots, m\}$. The problem is to assign one or more resources to each task and to determine each task's start time. Simple task precedence relationships may exist. We can represent these as a directed, acyclic graph $G = (T, E)$, where the edge $(j, k)$ exists in $E$ if and only if task $j$ must be completed before task $k$ can begin.

Tasks are non-preemptive; once some resources begin performing a task, that task must continue uninterrupted until it is completed. A resource can perform at most one task at a time. All tasks and resources are available at time 0.

There are two sets of decision variables: the task-resource assignments and the task start times. Let $S_j$ be the start time of task $j$ in $T$. Let $A_{ij} = 1$ if resource $i$ in $R$ is assigned to task $j$ in $T$. $A_{ij} = 0$ otherwise. Note that, in a feasible schedule, one or more resources are assigned to each task. The duration of a task $j$ in $T$ is a function of the resources assigned to it. Each task has a duration function $D_j(A_{1j}, \ldots, A_{mj})$ that is the task's duration for a given set of resource assignments. $D_j(A_{1j}, \ldots, A_{mj}) = \infty$ if the resource combination is infeasible for task $j$.

The objective of PSP is to minimize the total flowtime. Given a feasible schedule (a set of assignments and a set of start times), let $F_j$ be the finish time of task $j$:

$$F_j = S_j + D_j(A_{1j}, \ldots, A_{mj})$$

Let $X_{ij}(t) = 1$ if resource $i$ in $R$ is performing task $j$ at time $t \geq 0$ and 0 otherwise.

$$X_{ij}(t) = 1 \text{ if and only if } S_j \leq t \leq F_j \text{ and } A_{ij} = 1.$$

The objective function is the total flowtime: $F_1 + \ldots + F_n$.

A feasible schedule must satisfy the following constraints:

$D_j(A_{1j}, \ldots, A_{mj}) < \infty$ for all $j$ in $T$.

$F_j \leq S_k$ for all edges $(j, k)$ in $E$.

$X_{i1}(t) + \ldots + X_{in}(t) \leq 1$ for all $i$ in $R$ and $t \geq 0$.

$A_{ij} \in \{0, 1\}$ for all $i$ in $R$ and $j$ in $T$.

$S_j \geq 0$ for all $j$ in $T$.

Research into project management has considered controlling processing times by allocating resources [2, 3]. Other workers have studied resource-constrained project scheduling problems with controllable processing times [4-10]. Machine scheduling problems with controllable processing times have also received much attention [11-18]. Daniels and Mazzola [19], Daniels *et al.* [20], and Olafsson and Shi [21] have studied parallel machine and flow shop problems where there exists a set of renewable resources. Assigning one or more of these (identical) resources to a job affects the processing time. The problem is to simultaneously allocate these resources and sequence the jobs.

PSP, with multiple unrelated resources and a more general task precedence structure, includes these problems as special cases.

Chauvet, Levner, and Proth [22] study a general scheduling problem where there exist alternative jobs that must be selected. Applications include environments where these exist alternative process plans [23]. PSP could be transformed into such a formulation, where each feasible resource combination becomes an alternative job.

### 3.3 EXAMPLE

The following example illustrates some of the characteristics of PSP. There are four tasks and three resources. Two precedence relationships exist: Task 1 must precede Task 2. Task 1 must precede Task 3.

The following resource assignments are feasible: Task 1 requires either Resource 1 or Resource 2. In either case, the task duration is 5 hours. Task 2 requires either Resource 1 or the combination of Resource 1 and Resource 3. Its duration is 4 hours if Resource 1 performs the task and 2 hours if Resource 3 helps. Task 3 requires either the combination of Resource 1 and Resource 3 or the combination of Resource 2 and Resource 3. Its duration is 4 hours if Resource 1 and Resource 3 perform the task together and 2 hours if Resource 2 and Resource 3 perform the task together. Task 4 requires Resource 3 alone. Its duration is 7 hours.

Table 2 summarizes the task duration times for all possible resource combinations. Table 3 lists one combination of start times and resource assignments that form a feasible schedule. In this schedule, Resource 1 performs Task 1 and then Task 2. Resource 3 performs Task 4 and then, with Resource 2, performs Task 3. Since $F_1 = 5$, $F_2 = 9$, $F_3 = 9$, and $F_4 = 7$, the total flowtime is 30.

Table 2. All possible resource combinations.

| Task $j$ | Assignment $(A_{1j}, \ldots, A_{mj})$ | Duration (hours) $D_j(A_{1j}, \ldots, A_{mj})$ |
|---|---|---|
| 1 | (1,0,0) | 5 |
| | (0,1,0) | 5 |
| 2 | (1,0,0) | 4 |
| | (1,0,1) | 2 |
| 3 | (1,0,1) | 4 |
| | (0,1,1) | 2 |
| 4 | (0,0,1) | 7 |

Table 3. A feasible set of resource assignments and start times.

| Task $j$ | Assignment $(A_{1j}, \ldots, A_{mj})$ | Start time $S_j$ |
|---|---|---|
| 1 | (1,0,0) | 0 |
| 2 | (1,0,0) | 5 |
| 3 | (0,1,1) | 7 |
| 4 | (0,0,1) | 0 |

### 3.4 SOLUTION APPROACH

IPPS uses a three-step approach to solving the integrated process planning and production scheduling problem described above as PSP. First, based on information from the current schedule, the set of additional orders, material availability, and resource availability, IPPS generates a set of tasks that need to be completed. These include active tasks that have already been started and planned tasks that have not yet started.

The user is able to review these tasks and define additional preferences. These may force specific resources to perform the task, prohibit specific resources from performing the task, or constrain the task's start time or end time. This allows the PSP to capture external issues that are beyond the ordinary. PSP is formulated as described in Section 3.2.

Second, IPPS uses cybernetic optimization by simulated annealing (COSA) to find a superior solution. COSA is a parallel variant of the simulated annealing (SA) algorithm [24, 25]. The COSA framework utilizes feedback control mechanisms that enhance the convergence behavior of SA. SA is a metaheuristic and has the advantage of being broadly applicable. In addition, it is relatively easy to model problems for solutions by SA (and COSA). All that is required is a definition of an objective function and some neighborhood defined for each solution. Constraints can be easily incorporated into most problems by using penalty functions that augment the objective function. SA however, suffers from generally slow convergence. The COSA approach attempts to mitigate this slow convergence by using what is referred to as *probabilistic feedback control* and parallel processing. This gives COSA improved performance relative to SA and makes it a candidate for solving PSP.

By determining values for the resource assignment and start time variables, COSA selects the best process plan for each job and schedules each required task. To guide its search, COSA measures the total task flowtime of a schedule and adds penalties when constraints are violated. Thus, COSA attempts to find feasible schedules that minimize the time that jobs are in the shop (which minimizes in-process inventory). When COSA completes, IPPS uses a heuristic to repair any remaining infeasibilities.

Third, IPPS allows the user to modify the constructed schedule. The user can change resource assignments and task start times. After reviewing and updating the schedule the user can make it the official schedule by publishing it.

## 4. SUMMARY AND CONCLUSIONS

The ASSIST system is an intelligent knowledge management system designed to address inefficient information management processes associated with low-volume production. The goals of the program are a reduction in subsystem design effort, procurement cost, launch site support hours, and particular pre-launch testing. These objectives are met by the integration of successfully demonstrated individual technologies into a seamless electronic communication system, the ASSIST System. The system will be validated in a series of pilots and the results migrated to the industry through dissemination and the exploration of the commercialization. The ASSIST team is confident that these goals can be met, as well as achieving defense manufacturing affordability.

ASSIST integrates process planning and production scheduling by linking manufacturing operations with design decision support. The IPPS module performs process planning and production scheduling. The user can view and edit information needed for scheduling, construct a scheduling problem, solve the scheduling problem, view and edit a schedule, and publish a schedule.

The development of IPPS has led to a new combinatorial optimization technique for a very general class of difficult resource allocation and scheduling problems. The new technique, based on COSA, has great promise and will be tested extensively against other algorithms.

## ACKNOWLEDGMENTS

The development of IPPS was sponsored by Lockheed Martin Space Systems as part of the Affordable Space Systems Intelligent Synthesis Technology (ASSIST) for Manufacturing program. The University of Maryland performed this work under subcontract SY01H8901R. The University of Maryland team is grateful to all of the assistance provided by the Lockheed Martin ASSIST team.

The ASSIST Program is a 43-month program sponsored by the Air Force Research Laboratory, Materials and Manufacturing Division, as a Technology Investment Agreement (F33615-99-3-5902) under the Manufacturing Technology for Affordable Space Systems (MASS) initiative.

## REFERENCES

[1] Pinedo, Michael, *Scheduling: Theory, Algorithms, and System*s, Prentice-Hall Inc., Englewood Cliffs, New Jersey, 1995.

[2] Moder, J.J., C.C. Phillips, and E.W. Davis, *Project Management with CPM and Precedence Diagrammin*g, Van Nostrand Reinhold, New York, 1970.

[3] Wiest, J., and F. Levy, *A Management Guide to PERT/CPM With GERT/PDM/DCPM and Other Networks*, Prentice-Hall, Englewood Cliffs, New Jersey, 1977.

[4] Lawrence, S.R., and T.E. Morton, "Resource-constrained multi-project scheduling with tardy costs: comparing myopic,

bottleneck, and resource pricing heuristics," *European Journal of Operational Researc*h, Volume 62, pages 168-187, 1993.

[5] Slowinski, R., "Two approaches to problems of resource allocation among project activities: a comparative study," *Journal of Operational Research Societ*y, Volume 31, pages 711-723, 1980.

[6] Slowinski, R., "Multiobjective network scheduling with efficient use of renewable and nonrenewable resources," *European Journal of Operational Researc*h, Volume 7, pages 265-273, 1981.

[7] Slowinski, R., and J. Weglarz, "Solving the general project scheduling problem with multiple constrained resources by mathematical programming," *Lect. Notes Cont. and Infor. Syst.*, Volume 7, pages 278-289, 1978.

[8] Talbot, F.B., "Resource-constrained project scheduling with time-resource tradeoffs: the nonpreemptive case," *Management Scienc*e, Volume 28, pages 1197-1210, 1982.

[9] Weglarz, J., "Control in resource allocation systems," *Found. Cont. Eng*.,Volume 5, pages 159-180, 1980.

[10] Weglarz, J., J. Blazewicz, J. Cellary, and R. Slowinski, "An automatic revised simplex method for constrained resource network scheduling," *ACM Trans. Math. Soft*., Volume 3, pages 295-300, 1977.

[11] Adiri, I., and Z. Yehudai, "Scheduling on machines with variable service rates," *Computers and Operations Researc*h, Volume 14, pages 289-297, 1987.

[12] Daniels, R.L., "A multi-objective approach to resource allocation in single machine scheduling," *European Journal of Operational Researc*h,Volume 48, pages 226-241, 1990.

[13] Daniels, R.L., and R.K. Sarin, "Single machine scheduling with controllable processing times and number of jobs tardy," *Operations Researc*h,Volume 37, pages 981-984, 1989.

[14] Karabati, S., and P. Kouvelis, "Flow-line scheduling problems with controllable processing times," *IIE Transaction*s, Volume 29, pages 1-15, 1997.

[15] Trick, M.A., "Scheduling multiple variable-speed machines," *Operations Researc*h, Volume 42, pages 234-248, 1994.

[16] Van Wassenhove, L.N., and K.R. Baker, "A bicriterion approach to time cost trade-offs in sequencing," *European Journal of Operational Researc*h,Volume 11, pages 48-54, 1982.

[17] Vickson, R.G., "Two single-machine sequencing problems involving controllable job processing times," *AIIE Transaction*s, Volume 12, pages 258-262, 1980.

[18] Vickson, R.G., "Choosing the job sequence and processing times to minimize processing plus flow cost on a single machine," *Operations Researc*h, Volume 28, pages 1155-1167, 1980.

[19] Daniels, R.L., Hoopes, B.J., and J.B. Mazzola, "Scheduling parallel manufacturing cells with resource flexibility," *Management Scienc*e, Volume 42, pages 1260-1276, 1996.

[20] Daniels, R.L., and J.B. Mazzola, "Flow shop scheduling with resource flexibility," *Operations Researc*h, Volume 42, pages 504-522, 1994.

[21] Olafsson, S., and L. Shi, "A method for scheduling in parallel manufacturing systems with flexible resources," *IIE Transaction*s, Volume 32, pages 135-146, 2000.

[22] Chauvet, F., E. Levner, and J.-M. Proth, "On PERT Networks with Alternatives," Research Report 3583, INRIA, Le Chesnay Cedex, France, 1998.

[23] Kusiak, A., and G. Finke, "Selection of process plans in automated manufacturing systems," *IEEE Transactions on Robotics and Automatio*n, Volume 4, pages 397-402, 1988.

[24] Fleischer, M. (1996), "Cybernetic Optimization By Simulated Annealing: Accelerating Convergence By Parallel Processing and Probabilistic Feedback Control". *Journal of Heuristic*s, Vol. 1, No. 2, Spring 1996, pp.225–246.

[25] Fleischer, M. (1998), "Generalized Cybernetic Optimization: Solving Continuous Variable Problems" in *Meta-heuristics: Advances and Trends in Local Search Paradigms for Optimizatio*n, S. Voß , S. Martello, I. Osman, and C. Roucairol, editors. Kluwer Academic Publishers, Norwell, MA., Chapter 28, pp.403–418.