A Planning, Scheduling and Control Architecture for Advanced Life Support Systems

V. Jorge Leon
Texas A&M University
College Station, TX 77843-3367
leon@entc.tamu.edu

David Kortenkamp and **Debra Schreckenghost**
Metrica Inc., Texas Robotics and Automation Center
NASA Johnson Space Center – ER2
Houston, TX 77058

Abstract

This paper describes an integrated planning, scheduling and control architecture for robotics and advanced life support systems. The distinctive characteristics of controlled ecologies and the requirements for planning, scheduling and control architectures are presented. Next, the main components of the proposed architecture are described, and the interaction among the user, the intelligent planner, the generic scheduler, and the crop planner and scheduler is illustrated with a hypothetical scenario. Some successful implementations of components of the architecture and current efforts are also mentioned.

Introduction

We are developing a completely integrated planning, scheduling and control architecture for robotic and life support systems. The operation of a Controlled Ecological Life Support System (CELSS), either aboard a space station or ship or on the surface of the Moon or Mars, will require an intelligent monitoring and control system that can react quickly to short-term environmental changes while planning and scheduling for long-term effects of current actions. A CELSS must sustain a moderate size crew for a number of years with minimum re-supply of mass. Distinctive of this type of environment is the active participation of biological agents (e.g., humans and plants) in a system that possesses mass-closure. From the perspective of integrated planning, scheduling and control, important characteristics of advanced life support systems are:

- The simultaneous presence of difficult constraints including conservation / regeneration of mass, crew availability, space availability, and energy limitations (Leon 1995).
- Non-linear system behavior and long-term dynamics together with agents that respond indirectly to control signals (Auslander 1981; Colombano 1981; MacElroy 1981).
- The requirement of rapid response to environmental changes that pose danger to the crew, with consideration to the effect on long-term system stability of the corrective actions.
- The need to automate labor intensive tasks to offload the crew from time consuming and hazardous working conditions.

Each of these four areas is mission-critical, in the sense that an intelligent architecture must deal with all of them to ensure success. Dealing with the first problem requires reasoning about time and other resources, and scheduling those resources to avoid conflicts while managing dynamic changes in resource availability (e.g., decreasing food stores). Dealing with the second problem requires the ability to plan for a set of distant goals and adapt the plan, on-the-fly, to new conditions. Dealing with the third problem requires tight sense-act loops that maintain system integrity in the face of environmental changes. Finally, dealing with the fourth problem requires an integration of robotic control and scheduling with the overall monitoring and control of the life support system.

Requirements for an integrated planning, scheduling and control architecture

The effectiveness of an integrated planning, scheduling and control architecture for CELSS will depend on to what degree the following requirements are satisfied:

- Interactive planning and scheduling. Given the uncertainties associated with long-missions it would be impossible to automatically generate a plan, pass it to the scheduler, and be done with it in one iteration. The architecture must support the ability of move easily between planning and scheduling in an iterative process, allowing the user to provide significant feedback, if desired.

- User-definable abstraction levels for planning and scheduling. The decisions under consideration may be short-term in the order of minutes, to long-term in the order of years. The architecture must be able to move between various time and granularity scales presenting the correct level of information to the user at different decision situations.

- Flexibility. The user should be able to revise implemented solutions, as well as generate new solutions. The architecture should allow the user to play "what-if" games, evaluating different hypothetical scenarios while, at the same time, the architecture is executing the current scenario.

- The architecture must present a common view of the system to the user. Even though the architecture may be built of many different modules, the user's view should be centered on the kinds of functions the user desires, the problems they want to solve, and the kinds of information the want to see.

- The architecture must explicitly recognize multiple performance criteria. The user must be presented with information that depict the dependencies and compromises among these often conflicting criteria.

- The architecture must be open to facilitate its integration with other systems.

Our motivation is to produce a architecture that satisfies the above requirements. The system will allow the crew to build plans and schedules, play "what-if" games with those plans and schedules, then have the architecture execute them while presenting a consistent view of the state of the system to the use. Such an architecture will greatly increase the effectiveness of the crew in CELSS environments and greatly increase the scientific results of CELSS experiments.

An Integrated Architecture

The integrated architecture is based on the several lines of on-going research at NASA, Metrica and Texas A&M University. Figure 1 shows the complete architecture. The important components of the architecture are:

- A multi-tiered intelligent control system called 3T. The control system combines a reactive tier with a deliberative planning tier, both mediated by a middle tier conditional sequencer. This allows for long-range planning to take place while, at the same time, the system can react to immediate environmental events. This system is being used in several on-going NASA JSC projects (Bonasso *et al.* 1995).

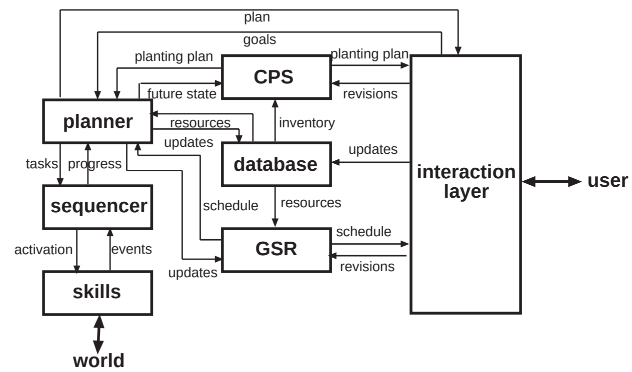

Figure 1: The complete planning, scheduling and control architecture for advanced life support systems.

- A scheduling methodology called the Generalized Scheduler/Rescheduler (GSR) that allows for the use of static, off-line schedules in uncertain environments (Leon & Balakrishnan 1995).

- A crop scheduling and planning methodology that can be used by the planner to decide what crops to plant at what time (Leon 1995; 1996).

- An interaction layer that mediates user interaction with the entire system. The interaction layer hides the various components of the architecture from the user, allowing him or her to concentrate on achieving tasks (Kortenkamp *et al.* 1997).

The rest of this paper will concentrate on the integration of the 3T control system with GSR, as this is the most mature part of the architecture. Brief mention of the other architectural components will be given, followed by an example scenario.

3T architecture

The 3T architecture separates the general intelligent control problem into three interacting layers or tiers:

- A set of hardware-specific situated skills that represent the architecture's connection with the world. The term "situated skills" is intended to denote a capability that will achieve or maintain a particular state in the world.

- A sequencing capability that can activate the situated skills in order to direct changes in the state of the world and accomplish specific tasks. This tier of the architecture is implemented using Reactive Action Packages (RAPs) (Firby 1989).

- A deliberative planning capability which reasons in depth about goals, resources and timing constraints.

We are using a state-based non-linear hierarchical planner known as AP (Elsaesser & MacMillan 1991) for this portion of the architecture.

The architecture works as follows: the deliberative layer takes a high-level goal and synthesizes it into a partially ordered list of operators. Each of these operators corresponds to one or more RAPs in the sequencing layer. The planner places these RAPs on the RAP agenda. The RAP interpreter (sequencing layer) decomposes each RAP on the agenda into other RAPs and finally activates a specific set of skills in the reactive layer. A set of event monitors is also activated to detect certain world conditions and notify the sequencing layer. The activated skills will move the state of the world in a direction that should cause the desired events. The sequencing layer will terminate the skills, or replace them with new skills, when the monitoring events are triggered, when a timeout occurs, or when a new message is received from the deliberative layer indicating a change of plan.

Related work The integration of intelligent planning and control is not a new topic in the artificial intelligence research community. The Cooperative Intelligent Real-time Control Architecture (CIRCA) (Musliner, Durfee, & Shin 1993; 1995) is designed to support both hard real-time response guarantees and unrestricted AI methods that can guide those real-time responses. CIRCA has an AI subsystem (AIS) reasons about high-level problems that require its powerful but uncertain reasoning methods, while a separate real-time subsystem (RTS) uses its predictable performance characteristics to deal with low-level problems that require guaranteed response times. For a good overview of real-time AI work see (Musliner *et al.* 1995).

Simmons' Task Control Architecture (TCA) (Simmons 1990) has been successfully used on a number of real-world robots, but it is very different from our architecture. There are essentially no tiers in TCA. A task net is constructed for the robot which is similar to a task-net in RAPs. Each node in the task tree can be decomposed further or can be a primitive which interfaces with the robot, or other nodes, through a sophisticated message-passing algorithm. These messages are processed through a central router, and thus TCA is more like a robot operating system. There are no explicit representations for expressing relationships among tasks. TCA task trees are manipulated directly by C function calls. Therefore, it is incumbent on the programmer to mentally compile the desired control constructs into the appropriate calls.

The Guardian architecture of Hayes-Roth (Hayes-Roth 1995) is a blackboard architecture designed for controlling embedded (though not necessarily embodied) agents. The architecture is divided into a cognitive component and a perception/action component. The perception/action component is controlled by the cognitive component. Thus, the Guardian architecture is similar to ours, but with sequencing and deliberation performed by the same mechanism. The deliberative component can modulate the performance of the perception/ action component, as well as its own performance, according to the current situation in the world. Guardian embraces traditional AI representation, but does not commit to any particular representation for describing the interrelationships among tasks.

The 3T architecture shares many aspects of Cypress (Wilkins *et al.* 1995). Our AP planner has similar expressive power at an abstract level as SIPE; RAPs compares favorably with PRS. But because RAPs were designed to allow integration with conventional AI planners, we did not have to write an interlingua such as ACTs to achieve such integration. Additionally, Cypress does not specify a canonical interface to the control tier as does 3T.

Generic Scheduler / Rescheduler (GSR)

Given a world's current state-of-affairs, planning specifies a sequence of tasks and required resources to achieve a given goal. Scheduling refers to the time ordering of activities given in plan such that a given objective function is optimized. This objective function may comprise multiple performance evaluation criteria. The capacity of each resource is explicitly considered, as well as, any other conditions required by the plan. Rescheduling refers to the "repair" of a given schedule. The "repair" can be triggered (by the user or automatically) by the occurrence of disrupting events, or when "sufficient" new information about the state-of-affairs becomes available. Rescheduling must minimize the impact that the proposed changes have on system performance. Also note that rescheduling assumes the existence of a schedule.

GSR incorporates multiple-objectives and implements user-interactive search for efficient solutions. The input to GSR is the plan generated by the deliberative layer of 3T. The plan contains information about the required tasks, potential resources required, time constraints, and precedence relations among activities. The output of GSR is a detailed schedule with exact start and finish times for each task and the corresponding resource assignments.

The scheduling engine is based on problem-space based neighborhood generation and search [19]. This approach is computationally efficient, produces good-quality solutions, it is easy to implement, and is very flexible allowing the incorporation of most operating conditions in the scheduling model. By carefully manipulating the attribute-weights and other parameters of the heuristic, and the weights of the criteria in the objective function, the user can "direct" the search to regions of the solution space that contain satisfactory schedules. The user is presented with graphical and numerical information about the quality of the solution during the interactive search process to aid him / her make decisions about the search direction.

The integration of GSR with 3T has significant potential for improved system performance. Test results suggest that reductions up to 50% can be achieved in terms of total completion-time and total tardiness metrics. These tests compared the results obtained with a planner using a rudimentary scheduler with that of the integrated 3T-GSR.

Related work Detailed reviews of control of manufacturing systems can be found in (Buzacott & Yao 1986; Gershwin *et al.* 1986). Previous research where explicit consideration is given to recovery from disruptions in schedules includes (Yamamoto & Nof 1985), where a new schedule is generated each time a disruption occurs, and (Bean *et al.* 1991), where recovery from disruption is made during a transient period of time, after which the new schedule matches up with the disrupted schedule. Wu et al (Wu, Storer, & Chang 1991) explicitly consider costs associated with rescheduling jobs before and after their original start times. However, none of this work integrates scheduling with a powerful planner to allow for long-range considerations to be taken into account by the scheduler.

Integrating planning and scheduling has received surprisingly little attention. There are two large efforts currently underway. The first is the OZONE/DITOPS project at Carnegie Mellon University (Smith, Lassila, & Becker 1996). The focus of this work, supported by the ARPA/Rome Laboratories Planning Initiative, is military crisis-action deployment scheduling. However, in this project planning does not mean full-fledged, state-based planning as we do with AP. Instead, planning refers to the preliminary phase of scheduling. This lends itself to problems in which the search-space is well-known and optimization is the important criteria. A second project is an integrated process planning and production scheduling system being developed by Raytheon and Carnegie Mellon (Sadeh *et al.* 1995). Again, the focus is on scheduling rather than planning. The system we propose has more powerful planning capabilities combined with a state-of-the-art scheduling engine.

Crop Planning and Scheduling (CPS)

This module is based on the work in [Leon 1995 and Leon 1996] and deals with the decisions of What, When, Where, and How Much to plant during a given planning horizon. What to plant decisions must select between various plant types (about 15 or more). When to plant consider a planning horizon measurable days. Where to plant is determined by the best growing conditions for each crop and affinity with other plants sharing the same growth chambers. How much to plant is restricted by the maximum planting area in the growth chambers (currently a few hundred square meters) and the size of the planting trays (currently about one square meter). The objective used is a function of the deviations from the ideal reservoir levels. This objective is used as a surrogate measure of the probability of survival. Other considerations are crew menu preferences, nutrition requirements, food stocks, and crew size changes. As a result, the decisions under consideration are non trivial due to the large size of the combinatorial solution space and inherent nonlinearity of the problem.

Interaction layer

Human intervention will be needed during both the planning and the scheduling phases. Human intervention at the planning level is needed to assist in generating and modifying plans and in viewing and comparing plans. Human intervention at the scheduling level is needed to allow the user to set preferences and constraints as well as to allow for changes in scheduled activities. In order to present a consistent interface to the user, all user interaction is with a separate module called the *interaction layer*. This layer maintains a relationship with the user, presenting the appropriate information at the appropriate time and allowing for user intervention into the system. It is not simply a graphical user interface; it is an intelligent agent that may contain a model of the user and their goals and intentions.

An Example Scenario

This scenario will illustrate the interaction among 3T's AP planner, Generic Scheduler Rescheduler (GSR) and Crop Planner and Scheduler (CPS). Figure 2 illustrates the simplified version of a CELSS considered in this example. In this hypothetical scenario the system consists of a crew of 4 people, two autonomous robots, several plant types (e.g., wheat, lettuce, soybean, potato), a food processing machine, limited planting area, limited processing area, and limited holding area.

At the decision time the current state-of-affairs has the following potential goals:

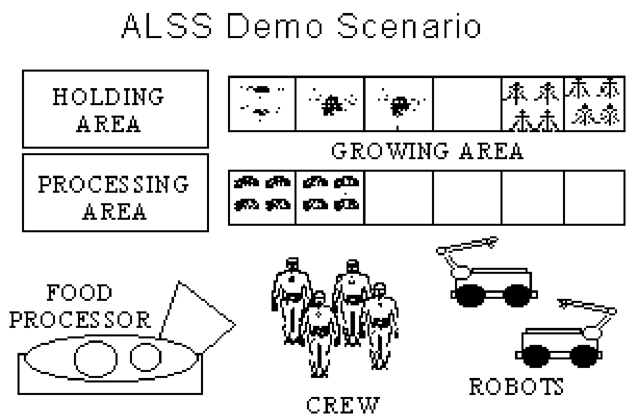

Figure 2: Simplified CELSS system.

- Some planting planting space is available
- A crop of soybean may be ready for harvesting
- A crop of potato is ready for harvesting
- The robots must be serviced within the next couple of days
- The food-processing machine must also be serviced within the next couple of days

Given this situation, the crew desires to generate a detailed schedule of activities that ensures long-term stability. Manual planning and scheduling may be complicated even for this simplistic scenario because: (1) the achievement of each goal requires the execution of numerous tasks, (2) these tasks are inter-related by precedence relations among them, and the usage of scarce resources (i.e., crew, robots, equipment, and space), and (3) the problem is further complicated by the fact that planting decisions will have an long-term effect on the system stability. The proposed architecture is aimed at aiding the decision maker in this kind of scenario.

Given the above state of the world, the crew decides to use the integrated architecture to generate a detailed schedule of crop-related activities and maintenance activities that will ensure the safe and stable operation of the ALSS. Recall that the crew is seeking for recommendations about what to do with the available planting space, soybean and potato harvesting, and robots and food-processing maintenance. In order to generate the detailed schedule, AP will interact with its scheduling (GSR) and crop planning (CPS) modules, in that order, as illustrated on Figure 3.

AP and CPS interaction. In order to decide WHAT, HOW MUCH, and WHERE to plant, such that long-

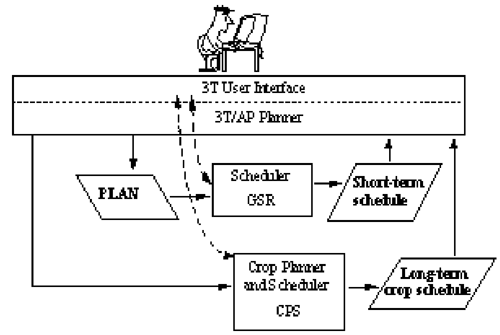

Figure 3: 3T - CPS - GSR Interaction.

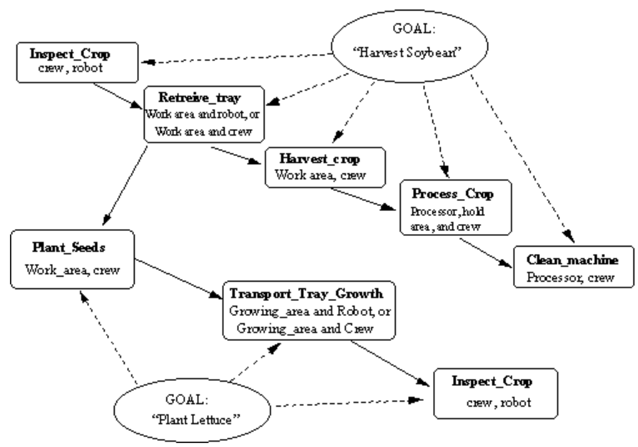

Figure 4: Partial plan generated by AP.

term stability is warranted and the probability of survival is maximized, AP relies on CPS. CPS takes the current state of the system reservoirs (food stores, gases, water), the expected crew profile for the next few months/years, and the state of the in-progress crops, among other system information. For this example, let us assume that CPS recommends, through an interactive session with the decision maker, to plant Wheat, Lettuce and Tomatoes on the available areas. This information (What and how much to plant) is passed to the AP planner. At this point the AP planner has all the goals required to generate a plan of activities.

AP planning

Given the goals "Plant Wheat," "Plant Lettuce," "Plant Tomato," "Harvest Soybean," "Harvest Potato," "Service Robots," and "Service food-processor," the AP planner searches its space-state until finds the tasks that will be passed to the lower levels of the architecture. Figure 4 illustrates the tasks required to accomplish the goals Plant Lettuce and Harvest Soybean.

The planner also designates which resources are suitable for the execution of each task and estimated resource consumption. The planner also may require additional time and precedence constraints as specified by operational conditions or the user. Notice that, the planner specifies that the "Retrieve-Tray(Soybean)" task must be accomplished before "Plant-Seed(Lettuce)."

Clearly, there will be significant contention among the activities (or primitives) to utilize the available scarce resources; that is, crew, robots, processing machine, etc. In order to ensure the efficient utilization of the resources AP calls GSR for a detailed schedule and resource-to-task assignment. Figure 5 illustrates the complete plan generated by the AP planner as it is passed to the GSR scheduling module.

GSR Scheduling

In an interactive session between GSR and the decision maker, a detailed schedule containing the exact execution time of each of the tasks is generated. In this schedule, the resources required to execute each task is completely specified. The resulting schedule is one that compromises the total completion time, tardiness of the activities and the decision maker's desires.

From the scheduling perspective, GSR possesses five important characteristics. First, GSR can handle complex precedence relations between activities. Second, GSR resolves alternative resource-group assignments which were only partially specified by the planner. Third, GSR handles multiple-objective explicitly. Fourth, GSR can easily implement resource-time constraints. Finally, GSR can function in a user-interactive mode, or in a fully automated mode.

The current implementation of GSR has a graphical user-interface which allows the user modify the search direction by interactively adjusting the relative weights between performance criteria, and algorithmic parameters. Graphical and tabular displays of the solutions found so far guide the user through the decision process and eventually, in the selection of the schedule to implement. Two output displays are used. Specifically, a Summary Report and a Gantt Chart. The Summary Report form contains a graphical display that allows the user to determine the goodness of a solution in all performance criteria with respect to an "ideal" solution. This report also contains a table with the numerical values for the various performance criteria. The Gantt Chart form displays the activities to be carried out by each resource as a function of time. The user can easily navigate from one form to the next in a windows-like environment.

What-if Analysis and Rescheduling with GSR

The computational efficiency and structure of GSR allows for effective "what-if" analysis and rescheduling. The user could easily modify his / her preference for resource-groups to execute a given activity, or could impose new precedence relationships between tasks.

Execution, monitoring, control and replanning

Once a detailed schedule is generated and approved by the user, AP starts requesting its execution to the lower layers of 3T. AP monitors the execution of the plan. If the system seems to be deviating significantly from the nominal path or new information about events become available, then AP may automatically request for a replanning. The crew can request for replanning at desire.

Current applications to ALSS

We are applying the 3T control architecture (without the scheduling extension) during a 90 day manned test in September 1997 of advanced life support systems for the Lunar/Mars Life Support Technical Program (LMLSTP) at NASA Johnson Space Center (JSC). 3T will be used to control the transfer of product gases (oxygen and carbon dioxide) between multiple gas reservoirs, including a plant growth chamber, storage tanks, the crew habitation module, and an airlock from which the solid waste incinerator draws air and vents effluent. For this application, the planning tier is essential to manage complex system reconfiguration and changes in control strategy required to maintain these multiple gas reservoirs at required levels during a variety of activities including seed germination, plant growth, harvest, and incineration. Even in this constrained application, we have identified a need for the GSR to provide a finer time granularity in the plan (a detailed schedule) and more exact control of start and stop times for activities.

The 3T control architecture also has been selected for controlling computer-controlled machines (robotic and regenerative life support) in the BIOPlex facility to be completed at NASA JSC in 2000. The BIOPlex facility will be a ground-based, manned test facility for advanced life support technology destined for use in lunar and planetary bases, and planetary travel (such as Mars Transhab Project). It consists of five connected modules - two plant growth chambers, a crew habitation module, a life support module, and laboratory. Regenerative life support systems include water recovery, air revitalization, solid waste management, and thermal/atmospheric control. Plant support systems

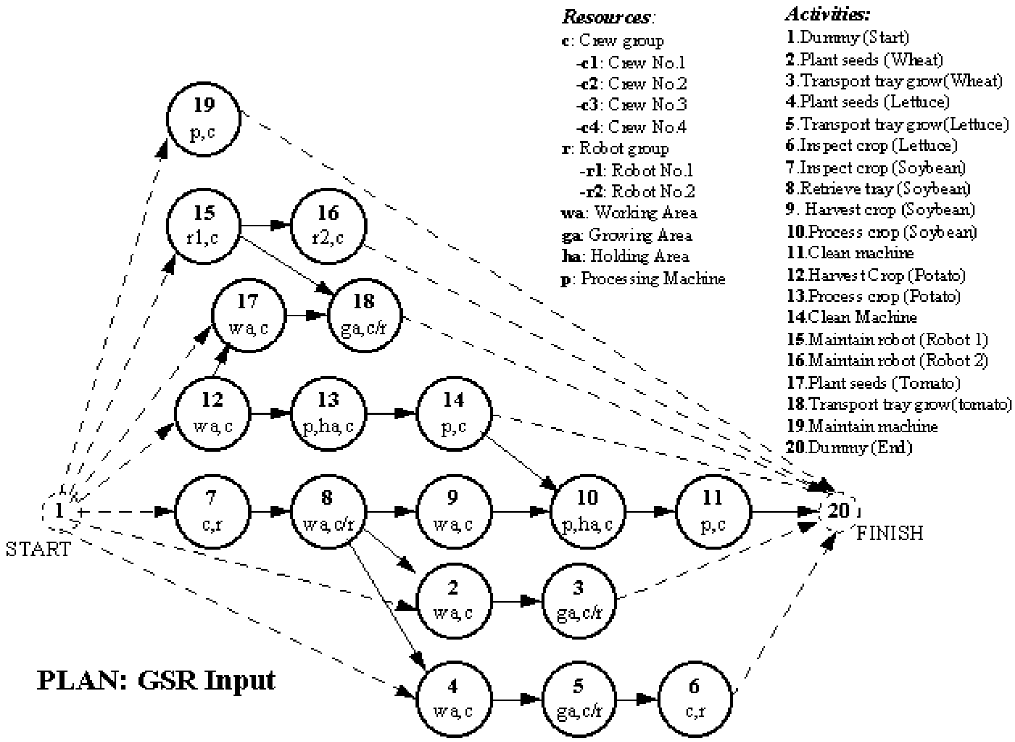

Figure 5: GSR input: the complete plan.

include nutrient delivery, gas management, and thermal/humidity control. Robotic systems include transport, manipulation, and sensor/video scanning. Controlling these heterogeneous systems to maintain food supplies and water and gas reservoirs, while minimizing solid waste reservoirs (inedible biomass and fecal matter) poses a challenging set of problems for planning and scheduling. The planner must balance conflicting system needs and account for cross system coupling, at time scales varying from hours to months. In this facility, human and robots will jointly execute tasks and must coordinate their efforts. A common/shared schedule for both crew and computer-controlled machines is needed to guarantee such coordination. This schedule must be sufficiently flexible to adapt to crew preferences while stable and robust for computer control. An integrated planning, scheduling, and control architecture that includes both fine time grain scheduling and optimization (GSR) as well as long term crop planning (CPS) will be required for BIOPlex.

Concluding Remarks and Future Work

This paper discusses the main characteristics of a CELSS and the requirements for intelligent planning, scheduling, and control architectures. An architecture developed jointly by Metrica Inc. and Texas A&M University for NASA-JSC is described and its functionality illustrated via a scenario. Preliminary testing of different components of the architecture are showing promising results. Future work includes the integration of the crop planning and scheduling module, the development of a coherent and intelligent interface layer, and enhancement of all modules.

References

Auslander, D. M. 1981. CELSS system control overview. In *Controlled Ecological Life Support Systems – Proceedings of First Investigators Meeting (NASA Publication 2247).*

Bean, J.; Birge, J.; Mittenthal, J.; and Noon, C. 1991. Matchup scheduling with multiple resources, release dates and disruptions. *Operations Research* 39(3).

Bonasso, R. P.; Kortenkamp, D.; Miller, D. P.; and Slack, M. 1995. Experiences with an architecture for intelligent, reactive agents. In *Proceedings 1995 IJCAI Workshop on Agent Theories, Architectures, and Languages.*

Buzacott, J., and Yao, D. 1986. Flexible manufacturing systems: A review of analytical models. *Management Science* 32(7).

Colombano, S. 1981. Control problems in autonomous life support systems. In *Controlled Ecological Life Support Systems – Proceedings of First Investigators Meeting (NASA Publication 2247)*.

Elsaesser, C., and MacMillan, R. 1991. Representation and algorithms for multiagent adversarial planning. Technical Report MTR-91W000207, The MITRE Corporation.

Firby, R. J. 1989. *Adaptive Execution in Complex Dynamic Worlds*. Ph.D. Dissertation, Yale University.

Gershwin, S.; Hildebrant, R.; Suri, R.; and Mitter, S. 1986. A control perspective on recent trends in manufacturing systems. *IEEE Control Systems Magazine* 1(1).

Hayes-Roth, B. 1995. An architecture for adaptive intelligent systems. *Artificial Intelligence* 72.

Kortenkamp, D.; Bonasso, R. P.; Ryan, D.; and Schreckenghost, D. 1997. Traded control with autonomous robots as mixed initiative interaction. In *AAAI Spring Symposium on Mixed Initiative Interaction*.

Leon, V. J., and Balakrishnan, R. 1995. Strength and adaptability of problem-space based neighborhoods for resource-constrained scheduling. *OR Spektrum* 17(1).

Leon, J. V. 1995. Intelligent planning and scheduling for controlled ecological life support systems. Technical report, NASA Summer Faculty Fellowship Program.

Leon, J. V. 1996. Integration of CELSS simulation with long-term crop scheduling. Technical report, NASA Summer Faculty Fellowship Program.

MacElroy, R. D. 1981. Current concepts of the CELSS programs. In *Controlled Ecological Life Support Systems – Proceedings of First Investigators Meeting (NASA Publication 2247)*.

Musliner, D.; ; Hendler, J.; Argrawala, A.; Durfee, E.; Strosnider, J.; and Paul, C. 1995. The challenges of real-time AI. *IEEE Computer* 28(1).

Musliner, D. J.; Durfee, E.; and Shin, K. 1993. CIRCA: A cooperative, intelligent, real-time control architecture. *IEEE Transactions on Systems, Man and Cybernetics* 23(6).

Musliner, D. J.; Durfee, E.; and Shin, K. 1995. World modeling for the dynamic construction of real-time control plans. *Artificial Intelligence* 74(1).

Sadeh, N.; Laliberty, T.; Bryant, R.; and Smith, S. 1995. Development of an integrated process planning/production scheduling shell for agile manufacturing. In *Proceedings of the IJCAI-95 Workshop on Intelligent Manufacturing*.

Simmons, R. 1990. An architecture for coordinating planning, sensing and action. In *Proceedings of the DARPA Workshop on Innovative Approaches to Planning, Scheduling and Control*, 292–297.

Smith, S. F.; Lassila, O.; and Becker, M. 1996. Configurable, mixed-initiative systems for planning and scheduling. In Tate, A., ed., *Advanced Planning Technology*. Menlo Park CA: AAAI Press.

Wilkins, D. E.; Myers, K. L.; Lowrance, J. D.; and Wesley, L. P. 1995. Planning and reacting in uncertain dynamic environments. *Journal of Experimental an Theoretical AI* 7.

Wu, S.; Storer, R.; and Chang, P. 1991. A rescheduling procedure for manufacturing systems under random distributions. In *Proceedings of Joint US/German Conference on New Directions for Operations Research in Manufacturing*.

Yamamoto, M., and Nof, S. 1985. Scheduling/rescheduling in the manufacturing operating system environment. *International Journal of Production Research* 23(4).