Task Decomposition Methodology for the Design of a Coal Mining Automation Hierarchical Real-Time Control System

Hui-Min Huang   Richard Quintero

Unmanned Systems Group
Robot Systems Division
National Institute of Standards and Technology
Gaithersburg, MD 20899

Abstract

This paper describes a systematic approach to hierarchical task decomposition and planning. Such a methodology can be used to design a complex system which receives goals from the external world, performs intelligent planning, and commands the actuators to achieve the goals. The application of this task decomposition methodology is illustrated in this paper through the design of a coal mining automation hierarchical real-time control system.

1. Introduction

The Mining Automation Standard Reference Model (MASREM) adapts the National Institute of Standards and Technology's (NIST) Real-Time Control System (RCS) [Al 82, Al 88] architecture to coal mining. The basic model behind the RCS technology is the intelligent machine system, as shown in Figure 1. The intelligent machine system has sensors that detect external events, and actuators that execute planned tasks to achieve the system's goals. The Sensory Processing (SP) modules perform filtering, correlation, and integration functions so that external world features and situations can be extracted. The Task Decomposition (TD) modules decompose and execute the system's goals. The World Modeling (WM) modules store information, answer queries, evaluate situations and make predictions. The MASREM is a hierarchical architecture.

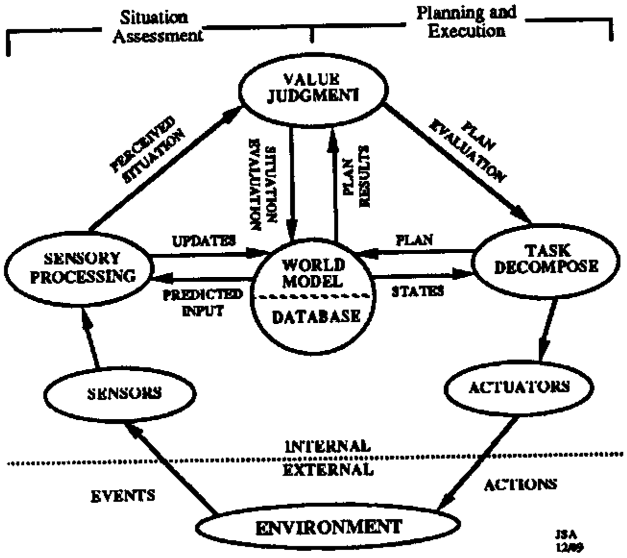


Figure 1: The Model of an Intelligent Machine System

Volume I of MASREM [Al 89, Hu 90-1] was developed to define the overall conceptual framework of the hierarchical architecture for the mine automation control system. The functional hierarchy contains seven control levels, as shown in Figures 2 and 3 (in the figures, all blocks are subsystems to their parent blocks except for the highest block). Conceptually each block has the three above-mentioned functions: SP, WM, and TD. It would be an enormous task to design a full system. Therefore it is essential to use specific examples to illustrate the

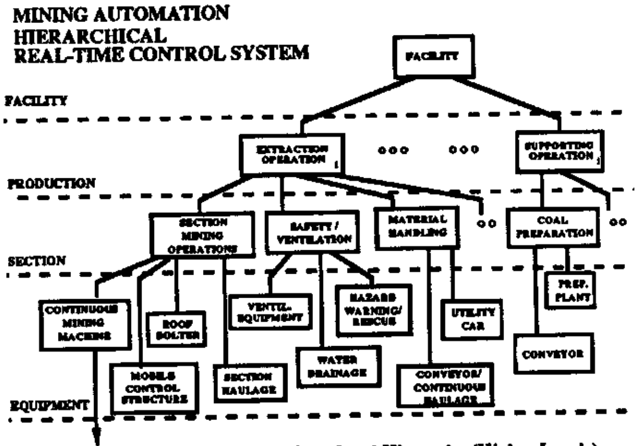


Figure 2: Mining Automation Functional Hierarchy (Higher Levels)

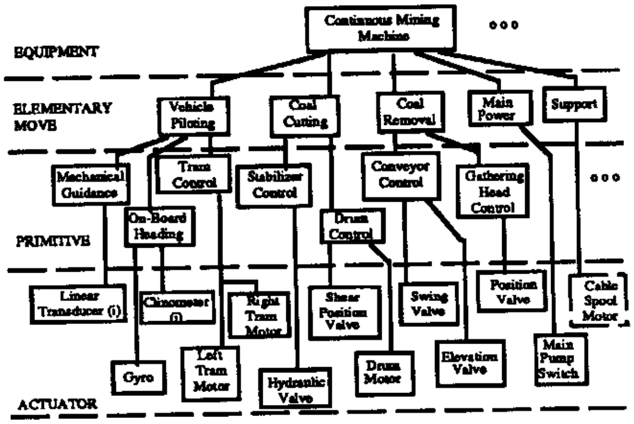


Figure 3: Mining Automation Functional Hierarchy (Lower Levels)

application of RCS, so that interested parties can pursue an incremental implementation and integration towards a complete system.

This paper addresses key issues of the Task Decomposition function of RCS. The overall control architecture developed in this paper is generic and therefore may be applied to any mining operation. The emphasis here has been placed on the Joy CM continuous mining (CM) machines (reference to a company's or a product's name is for identification only and does not imply government endorsement) [Pa 88, Jo 82], as this is in line with the current research focus of the United States Bureau of Mines (BOM), as is outlined in [Sch 89-1].

This paper is a condensed version of [Hu 90-1], which was written as a part of the "Architecture for Internal Control of Mining Machines" project, sponsored by the United States Bureau of Mines (Interagency Agreement No. J0189027).

2. Task Decomposition Methodology

The RCS Task Decomposition function is responsible for planning and executing the decomposition of goals and/or tasks at each level of the system's hierarchy. A RCS intelligent control system interacts with the environment and receives a compound goal at its highest level, and at the lowest level acts on the environment to achieve the goal. Internal to the system, hierarchical and heterarchical task decomposition as well as temporal and spatial task decomposition are occurring. Therefore, it is sufficient to describe a task decomposition between any two successive levels as the higher level sending down "what needs to be done," and the lower level generating "how it can be done."

In designing a RCS, the definition of context is the first task. This task includes the establishment of the system's objectives, the problem domain, the constraints, and the overall assumptions. The task must also include a narrative description of the approaches used to achieve the goals and the system's typical scenarios to be performed. Once the context is defined, the development of the system's Task Decomposition can be approached by the following steps:

* Develop a Functional Hierarchy. The first step in setting up the basis for development of task decomposition is a structure that takes into account the system's goals, the environment, the existing facility, and other constraints.

* Perform Task Analysis and Develop Task Commands for Each Subsystem at Each Level. Answers to the following questions must all be specified: what can each subsystem perform, what are the constraints, and what information is required in order to perform a given task?

* Develop State Transition Diagrams. Task commands defined above are used to develop State Transition Diagrams to describe how higher level tasks are decomposed into lower level tasks, and how the constraints for the commands are implemented as transition requirements among different states.

Other steps such as:

* Design Algorithms and Establish Requirements for the Job Assignment Managers, the Planners, and the Executors [Hu 90-1, Hu 90-3]

* Develop Interface Specifications for the Integration of Task Decomposition with the World Model

are considered the second stage of task decomposition (since their objective is to define the processing necessary to execute the machine intelligence developed in the previous three steps) and will not be discussed in this paper.

The above five steps are tightly coupled. The guidelines for developing the functional hierarchy (Section 3) establish the basic functional requirements for each subsystem within each level, which will be used to define the system's activities (Section 4). The task commands and constraints developed are basic elements to develop intelligent plans (Section 5). Likewise, all the results developed from Sections 3 through 5 dictate the requirements for the processes (the job assignment module, etc.) that execute the plans. They also dictate the world model support requirements.

3. Functional Hierarchy Development Guideline

A functional hierarchy, Figures 2 and 3, lays out all the necessary functional modules (as well as subsystems) and the relationships among them. The following is a guideline for developing a RCS functional hierarchy.

3.1 Hierarchical Levels and Their Pre-Defined Functional Requirements

In the RCS methodology, the hierarchical levels are established and their functions are defined by taking into consideration the following:

* Logical Decomposition of Tasks -- In executing a task, component dynamics should be computed before actuator commands can be generated. Kinematic consideration takes precedence over dynamic consideration. Therefore RCS typically has an Actuator/Servo level for actuator commands as level 1 (the lowest level), a Primitive Level for component dynamics as level 2, and an Elemetary-Move (E-Move) Level for kinematics as level 3 to perform the above-mentioned functions.

- Consistency with Natural Spatial and Temporal Boundaries -- The existence of distinct physical entities dictates the need to have an Equipment Level in RCS. The fact that machines work together to form groups and groups coordinate to form super-groups dictates the need for multiple higher levels beyond the Equipment Level. In RCS, planning and response time intervals increase by one order of magnitude per level (Figure 4). This facilitates task planning and event summarizing, as well as smooth transition in time during task execution at different levels.

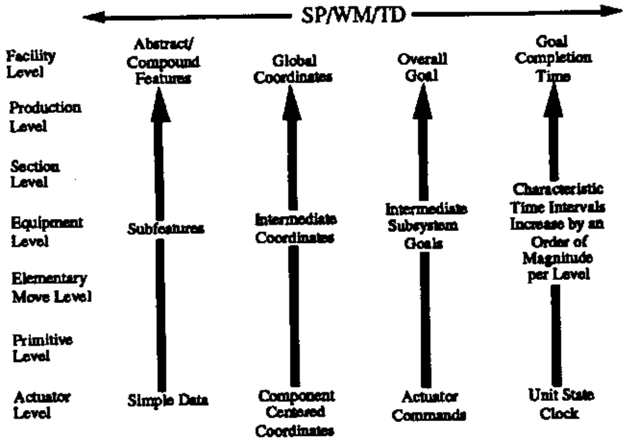


Figure 4: Transition in Temporal and Spatial Scales in a RCS Hierarchy

In designing a RCS for the automation of coal mining, a CM is modeled as an Equipment Level controller. The major subfunctions of the CM are modeled as its associated E-Move Level subsystems (including pilot/guidance, coal cutting, etc., as shown in Figure 3). The Production Level is created to deal with the large difference in planning horizons between the Section Level and the Facility Level and to account for the fact that multiple extraction operations may run concurrently in a large coal mine. Refer to [Hu 90-1] for detail.

3.2 Operation Requirements and Functional Coherence

The closely coupled face area operations in a coal mine dictate the need for a Section Mining Operation subsystem at the Section Level to coordinate operations such as coal cutting (performed by CM's), coal haulage (performed by shuttle cars or continuous haulage units), bolting, etc. In another example, the Elementary Move Level is developed by observing the major operations of equipment (Figure 3).

3.3 Autonomy and Modularity

The RCS methodology emphasizes maximizing the autonomy and the modularity of all subsystems. To achieve subsystem autonomy and modularity, the functional hierarchy is developed so that each process (such as the tramming function at the Primitive level, or the Section Mining Operation at the Section Level) will have a closed loop at the lowest practical level. By doing so, independent (autonomous) subsystems are formed. Subsystems may themselves be composed of several hierarchical levels. Each level of subsystem decomposition includes sensory information input, data storage, data manipulation routines, state space models, control laws, and output commands. Sensors and actuators are connected through SP, WM, and TD modules to form a closed loop. At each level, a loop is closed through the SP, WM, and TD modules at that level, so that the control hierarchy forms a set of nested control loops. The loop band-width decreases about an order of magnitude per level from the bottom to the top of the hierarchy. Therefore the autonomy and modularity guideline promotes self-sustained modules and locally maximized communication traffic as well as system extensibility.

By closing the loops at the lowest practical levels, changes in any module would have minimal effects on other modules. Without this autonomy and modularity approach, data queries may logically pass through longer routes [Hu 90-2].

3.4 The Bottom-Up Design Approach versus the Top-Down Approach

The autonomy and modularity guideline referenced above is one reason the development of a RCS application often starts with the lower levels. Each autonomous module is used as a building block and therefore the system is built up from the bottom (see [Hu 90-1] for other reasons). But on the other hand, the development of a system from the top down also has advantages. One advantage is that generic assumptions and overall strategies for certain major operations can be established up front for easier reference.

The RCS approach described here starts top-down in defining the functional hierarchy then moves to a bottom-up (object-oriented) approach in developing the Actuator/Servo Level, Primitive Level, E-Move Level, etc.

3.5 Concurrent Computing Timing Requirements and Software Module Sizes

The predefined cycle time (Section 3.1) for each level affects the development of the hierarchy by constraining the sizes of the software modules. The synchronization requirements also affect the hierarchy development. For example, in the CM, the fact that the cutting drum motor should be turned on before the machine can sump in the coal face implies frequent synchronization will be required; therefore these operations should be performed by parallel subsystems belonging to the same parent subsystem.

3.6 Achieving Automation Through Integration

It is desirable for a coal mining control system to have a functionally distributed but integrated architecture. A mine facility may have multiple extraction operations running concurrently and numerous pieces of equipment spread throughout. Each would have its own local controller and would be connected by certain network schemes. In addition, there has been major development work at many institutions such as the Bureau of Mines, Carnegie Mellon University, and West Virginia University, and from this work resources in the form of software/hardware will be produced. A key to automation is to have a comprehensive system architecture to integrate these resources. Such an architecture must be designed for the mining industry. MASREM, as a generic conceptual architecture, can serve this purpose.

There are several ways to integrate components developed by multiple institutions. One is to have all the work conform to one such structure. A more flexible approach is to standardize interface formats and required interface information (derived from the functional requirements specified in the reference architecture). A reference architecture can then be used to check to ensure the existence of all necessary functions from a system's point of view, so that there will be no missing pieces in the resulting system.

For the MASREM to be a mining automation reference architecture, the flexible integration approach involves:

* consistent data modeling, particularly with regard to a dynamic mining plan representation (advantageous in that it helps to facilitate easy data communication);

* sharing the same design concept, such as having a distributed World Model to compute the best estimated world states to enable response to Planner/Executors' queries, or having Planners that plan, update, and prioritize task commands; and

* establishment of the interfaces between systems developed by other researchers and the various subsystems in a RCS architecture.

In this approach RCS is a conceptual structure for all the essential functional elements in an automated coal mining system, and is used as a systematic approach in designing such a complex system. By this approach, a distributed but integrated system architecture can be developed.

3.7 Application Specific Factors

There are application specific factors which can affect the development of the functional hierarchy. They include: the system's goals, the existing facilities and resources, environment, and cost [Hu 90-1, Hu 90-3].

4. Task Analysis and Task Command Lists

In hierarchical real-time control, the system's overall goal or task is received at the highest level. The goal is decomposed into detailed tasks at lower levels and is executed by controllers at and for those levels. To achieve this, each level's functions must be identified first. Machine activities (and system activities at the higher levels) have to be defined specifically by means of a complete list of task commands. Each command is described later in this Section and again using State Transition Diagrams (STD) in Section 5. Task command definition involves the way each individual machine behaves, as well as the way machines coordinate among themselves. The combination of individual behavior and cooperative behavior specify the system's capability. Any mining plan or mining scenario developed can then be described by the task commands and by the State Transition Diagrams.

4.1 Considerations for Task Analysis

Task analysis seeks to resolve the following questions for a system and its subsystems: what tasks are implied, how can these tasks be performed within, and what are the requirements and constraints of the system. In particular, when multiple subsystems are involved, the complexity of cooperative behavior makes the definition of system and individual machine activity even more necessary. In performing task analysis, the following issues should be considered:

4.1.1 Spatial Coordination Strategy The spatial coordination strategy determines how individually automated machines cooperate. The spatial coordination strategy is a crucial step in achieving system integration. One example is the alignment problem between a haulage unit and a continuous miner.

After the CM has arrived at the face area, either it already carries the continuous haulage unit with it from a previous entry, or the haulage unit would approach and align to the CM. The CM would not tram to approach the haulage unit.

The criteria for the alignment may include the relative angle between the center lines of the machines, the clearance for the machines at their facing ends, and the CM conveyor boom positions. The alignment criteria are affected by the types of the on-board sensors as well as the type of haulage units used. If a continuous haulage unit is used, the coordination strategy may be defined as:

* prior to cut: The continuous haulage unit would physically engage with the CM. The CM will then position the boom to complete its ALIGN TO THE HAULAGE UNIT task (an E-Move Level command that has been defined [Hu 90-1]).

* during the cut: The CM COAL LOAD E-Move command will swing the boom and load the coal. Fault conditions may occur in the continuous haulage unit: the system may become jammed, it may be stretched to its limit, it may come in contact with corners as it turns, etc. Under these situations the required action for the continuous haulage unit is to stop and send a pause signal to the CM.

4.1.2 System/Machine Capability Definitions and Assumptions There are two types of information characterizing an automated machine:

- Behavioral Characteristics: what commanded operations can it perform?

- Physical Characteristics: machine dimensions, tramming speed, boom reach, etc.

This paper focuses on the first issue. A SUMP is defined here as the tramming of the continuous miner (CM) into the coal face for a distance of less than the cutting drum diameter. But physical characteristics are only symbolically defined in this paper.

4.1.3 Real-Time Planning vs. Predefined Script Planning In a real-time planning application, the generic plans can be described in advance by using State Transition Diagrams, but the selection of plans and the computation of the target values for the involved state variables are done in real-time based on sensory feedback information. Replanning may also be necessary when the system does not approach the goal as expected by executing the preselected plan. On the other hand, a more primitive format for task planning is to have pre-defined scripts. Capabilities such as plan selection and replanning may not be available.

4.1.4 Emergency Reaction Capability of Each Machine When Encountering Unexpected Problems Do all machines contain the same level of intelligence, or are there one or two dominant machines? For example, the haulage system may only be able to react to certain given commands whereas the CM is able to resolve more complex situations involving the haulage system, and may send commands to the haulage system to resolve the haulage system's problem.

4.1.5 Coordinate Reference Frames, Resolutions and Task Command Complexity for Different Levels In RCS, higher levels are concerned with larger areas but coarser resolution (Figure 4). Higher level tasks also cover a greater period of time but less spatial and temporal detail. In general, at the higher levels, a global coordinate frame is used, and in the lower levels, machine centered local frames are used. In a global frame, further subclassification in terms of resolution typically is required for different levels. Therefore successive transitions in coordinate frames or resolutions can be seen among different hierarchical levels. A mining operation can be specified by the production tonnage required at the highest level. Tasks are then decomposed at a lower level; to accomplish numerous subgoals, at some level a task may be defined in terms of commands to produce coal in a certain area of the coal seam, while at a lower level tasks are decomposed in terms of the coordinate position where coal is to be cut. A Mobile Control Structure coordinate system may be used at this level. At even lower levels, the mining sequences are defined as to the number of cuts to make, the number of sumps to make, all the way down to the amount of tram motor current needed for the distances involved. Lower level tasks must also deal with functions to be performed in a three dimensional world. For example, lower level coal mining tasks must be concerned with the height of the coal seam to be cut and with transforming commands from higher tasks requiring the cutting of coal (of some thickness) into tasks defined in terms of an initial boom position angle and the number of degrees of shearing angle to cut. Another example is that in dealing with the machine heading, '45 degrees' means northwest in the higher levels, but may mean counter-clockwise 45 degrees relative to last heading in the lower levels.

RCS, in most cases, is flexible as to which reference frame each level should use. But the key point is that at the highest level, a global coordinate frame is used, and at the lowest level, the individual actuator coordinate frames are used. Developers should be aware that coordinate transformations are used throughout a RCS design in order to simplify computations and to facilitate sensor fusion and model matching.

4.1.6 Existing Constraints, Existing Practices, and Flexibility The existence of certain equipment dictates the existence of certain fixed tasks. Examples can be seen in the Actuator Level, where capabilities of the valves, motors, or sensors are basically fixed, and thus their definitions can be viewed as the descriptions that conform to the existing capabilities. Examples can also be seen in the Primitive Level, where the Joy 16 CM Tram Control can have only twelve hard-wired commands. Another part of existing constraints is regulations. For example, in the Section Level, ventilation has to be set up before the CM equipment can operate at the coal face, therefore these two task commands must have synchronization built in (besides the existing constraints, designers are given flexibility to define task commands).

Existing mining practice may be used as a reference to identify task commands. For example, the sump and shear cycle is such a typical mining practice that the sump and shear commands described in MASREM correspond to it. But the MASREM does not intend to entirely follow the existing practice. The RCS "Section Mining Operation" (Figure 2) subsystem is responsible for fewer pieces of equipment than a section foreman [Us 68]. The distinguishing criteria is the computing efficiency versus human control efficiency.

All these issues (from Sections 4.1.1 to 4.1.6) must be addressed by the establishment of the task commands. The format, the parameters, and the description for each task depict the requirements.

4.2 General Syntax

The general syntax has to be defined to establish a reference for the task commands. Coordinate frames and the task command formats are among those defined. Refer to [Hu 90-1] for detail.

4.3 Task Commands for the Continuous Miner and for the Higher Levels

A set of task commands for a Continuous Miner (CM) and related Section Level and Operation Level task commands have been defined [Hu 90-1]. They are used in Section 5 to describe RCS plans. These plans represent a minimum set of commands that are required for the system to perform automated operations. Therefore, the described task commands include those commands that the current equipment is not capable of performing, but are desirable for automation purposes and those commands that are currently performed by human operators.

The CM receives Equipment Level task commands from the Section Mining Operation subsystem CM Planner, and outputs actuator commands, as shown in the following sections.

4.4 The Integration of BOM/NET Commands

The United States Bureau of Mines Pittsburgh Research Center has developed a control and communication specification, BOM/NET [Us 90], for the Joy CM mining machines. BOM/NET defines a complete set of Continuous Miner primitive functions. It also defines a set of commands for, and responses from, each sensor package (refer to Figure 3 for the specific sensors). The commands and responses are all implemented as message packets with a standard format so that they can be sent across the bitbus network to their destinations. The BOM/NET commands are used in this paper in defining the inputs to the Actuator Level for the following reasons:

- To preserve a coherent interface between the RCS and the current BOM automation research work.
- In the RCS, the Primitive Level deals with system's dynamics. In the BOM/NET protocol, commands specify dynamic characteristics for each actuator, such as velocity, maximum time limits, etc. Therefore a correspondence can be found between these two systems.

5. State Transition Diagrams for Plan Description

The methodology of state transition diagrams and how they are applied in a RCS architecture to represent RCS plans is described below. A series of examples illustrating a vertical swath of task decomposition are also explained.

5.1 Methodology

State transition diagrams (STD) are used in task decomposition, in plan description, and in machine activity description. One assumption of STD's is that the system's states will not change unless all transition requirements are met and that no action will take place until all activation prerequisites are met. Generally there is a one-to-one correspondence among a 'state,' a 'command,' and an 'activity'; in other words, the system enters into a certain 'state' when a corresponding 'command' is being executed and the corresponding 'activity' is being exhibited.

A similar one-to-one correspondence also exists between a 'transition condition' represented by a status data set and an 'event' occurring in the external world.

In a State Transition Diagram, a bubble with an enclosed name is used to represent a system's state (and the implied command), and the arrowed edges entering or leaving a bubble are used to describe the system's state transitions. Together the bubbles and edges completely describe how the system is to enter, to stay, and to leave any particular state (by following the direction of the arrows). Each edge has a definition attached to it and is typically described internally by a condition list which contains multiple state values, special flags, predicate function values, or other aspects of the system's status.

One general assumption in the state transition diagrams is that each command has a timeout limit. If the command can not be accomplished within the time limit, the system automatically branches out to a 'suspend' state and fault reports are issued. Proper actions need to be taken either by human action or by certain emergency recovery processes such as the Executor emergency planning routines. The execution of the commands is also subject to interrupts (generated by either human or computer) that suspend certain commands and send the system into the required states.

5.2 A RCS Plan

A RCS plan can be described by one or a series of STD's. The TD module for any subsystem has a Planner (PL) for each of its next lower level subsystems (or actuators) that it controls. This Planner generates (or selects) a plan for the subsystem (or actuator) that it controls. The commands in the plan are passed down sequentially by the Executor (EX) associated with the PL to the Job Assignment Module (JA) of the next lower Task Decomposition module (or to the actuator).

5.3 Plan Frame

Each state transition diagram uses a plan frame which includes the following slots to identify the plan:

- Plan Name -- The name of the command to be described by the current state transition diagram is used as the name for this plan.
- Plan Number -- The first segment is the number of the level which generates (not executes) the plan; it is followed by a dot and a second segment which is a serial number (more segments may be needed if the complexity of the system grows).
- Generated By -- The module generating the plan is described by
  - the name of the hierarchical level
  - the name of the subsystem
  - the name of the functional module.

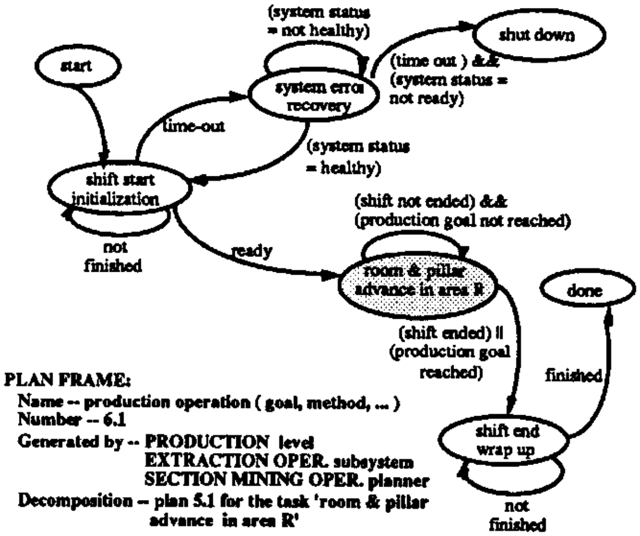
- Decomposition -- This slot cross references the commands associated with each state on the current diagram to the corresponding next lower level diagrams.

An optional "Description" slot can also be inserted to describe the activity.

5.4 A Vertical Swath of RCS Plans

A series of illustrative RCS plans, in hierarchical order, has been developed to show the successive task decomposition process. Bubbles were shaded to indicate that they have been decomposed in this paper:

5.4.1 A Production Operation Plan This plan (Figure 5) is generated by the Production Level Extraction Operation #i subsystem Planner. The first step of this plan is a shift start initialization. Detected inconsistencies such as equipment non-readiness would be allowed a certain time to recover. At the end of this period the Executor would decide whether to proceed to the next step on the normal plan, or to do one of the following two tasks due to un-recoverable system faults: perform emergency planning, or suspend operation altogether and report the status to the Facility Level Planner.



PLAN FRAME:
Name -- production operation (goal, method, ...)
Number -- 6.1
Generated by -- PRODUCTION level
EXTRACTION OPER. subsystem
SECTION MINING OPER. planner
Decomposition -- plan 5.1 for the task 'room & pillar advance in area R'

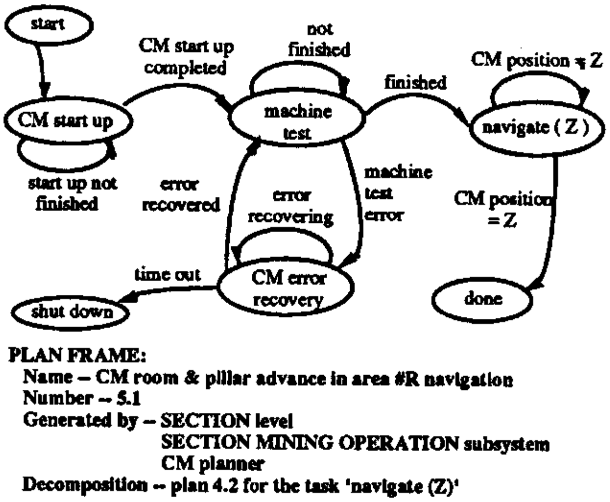
Figure 5: A Production Operation Plan

After the initialization finishes satisfactorily, the next task to be carried out is a ROOM AND PILLAR ADVANCE IN AREA R ("room and pillar" is a mining method), the criteria to end this task are that either a shift has come to an end, or the production goal has been met. Afterwards a SHIFT END WRAP UP task is executed before the plan is completed and the next plan is generated.

The ROOM AND PILLAR ADVANCE IN AREA R command is decomposed into the following Plan 5.1 and Plan 5.2.

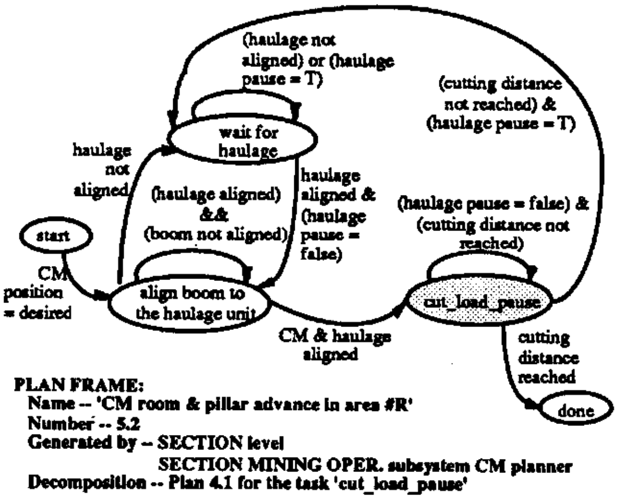
5.4.2 A Room and Pillar Advance Navigation Plan This plan (Figure 6) is generated by the CM Planner residing in the Section Mining Operation subsystem of the Section Level. The Job Assignment Module (JA) assigns the CM a room and pillar operation in area R. The CM plan that is selected would start with a machine test and a start-up procedure, followed by a navigation task to reach a more specific location in that area (the "specific location" has a resolution that is of about one order of magnitude finer than that of "area R," as a result of the hierarchical decomposition).

5.4.3 A Room and Pillar Advance Plan This plan (Figure 7) is also generated by the CM Planner residing in the Section Mining Operation subsystem of the Section Level. At the beginning of this plan, the machine is expected to be at some desired location; otherwise a time-out signal would be issued and Plan 5.1 would be selected and executed first.



PLAN FRAME:
Name -- CM room & pillar advance in area #R navigation
Number -- 5.1
Generated by -- SECTION level
SECTION MINING OPERATION subsystem
CM planner
Decomposition -- plan 4.2 for the task 'navigate (Z)'

Figure 6: A Room & Pillar Advance Navigation Plan



PLAN FRAME:
Name -- 'CM room & pillar advance in area #R'
Number -- 5.2
Generated by -- SECTION level
SECTION MINING OPER. subsystem CM planner
Decomposition -- Plan 4.1 for the task 'cut_load_pause'

Figure 7: A Room & Pillar Advance Plan

Plan 5.2 (Figure 7) describes a cut and load operation. The conveyor boom at the rear of the CM is to be aligned with the haulage unit first (this means the haulage system has to be in place already), then the CUT_LOAD_PAUSE activity can begin. Pause signals can be generated and can happen in various situations, for example if the haulage unit is away, is full, is jammed, or has other problems. The conditions for the CM to exit the cut state are that either the cutting distance is reached (refer to Section 4.4.5 CUT_LOAD_PAUSE command definition for the possible replanning of the desired cutting distance) or an external pause signal is received. In the former case the system goes into a 'wait' state, and in the latter case the plan is completed.

5.4.4 A CUT_LOAD_PAUSE Plan This plan (Figure 8) is generated by the CM equipment Piloting/Guidance (P/G) subsystem Planner. The P/G subsystem is reset according to the types of tasks that need to be performed ('navigation,' 'cut,' or 'wait,' as described in Section 4.4.4). The plan includes a wait in order to synchronize other parallel tasks. For example, the stabilization jack needs to be in a lifted position. After all the CM subsystems are reset, the P/G planner issues a tram command to align the machine to a desired cutting angle. A sump task will follow. During the sump, several situations may happen that would lead to an exit of the state: first of all, on-line Coal Interface Detection (CID) devices may discover an unexpected layer of rock. The Piloting/Guidance subsystem then has to go into a WAIT state, and the emergency replanning activity will take place to decide what to do next. The replanning may result in a request for the P/G subsystem to resume the sump, or it may simply ask the P/G subsystem to back out. A second situation to terminate a sump is that either the sump distance or the cut distance is reached. If a sump distance is finished, the P/G subsystem would typically wait for the Coal Cutting subsystem to perform a SHEAR operation, then proceed for a trim operation to cut the ridge. The 'align to cut - sump - shear & load - trim & load' cycles (performed by different E-Move subsystems) are continued until the cutting distance is reached.

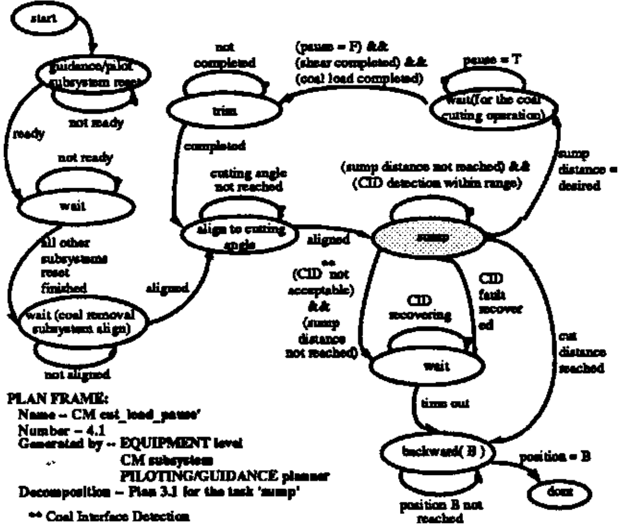


Figure 8: A Cut_Load_Pause Plan

5.4.5 A Sump Plan This plan (Figure 9) is generated by the E-Move Level Piloting/Guidance subsystem Tram Control Planner. A sump is basically a CM tramming forward task, with the cutter motor and the Coal Interface Detection devices running. Wet or loose ground may cause the CM to slip, which in turn causes the CM STRAIGHT distance to not be reached within a given time limit. The system may need to be shut down at this point.

6. Summary and Future Work

The hierarchical real-time control task decomposition methodology was described through the development of an illustrative coal mining automation system. A guideline involving software modularity and subsystem autonomy, the system's operation, the environment, the computing technology (both hardware and software), etc., were discussed in developing the system's functional hierarchy. The guideline also establishes the functional requirements for each subsystem and each level. In performing the task analysis and defining the system's activities, the system's pre-existing capability is taken into account, the machine coordination strategy has to be decided, and a transformation of coordinate frames and resolutions has to be observed during the hierarchical task decomposition. State transition diagrams are used to define plans using the task commands (including constraints) developed using the RCS task decomposition methodology.

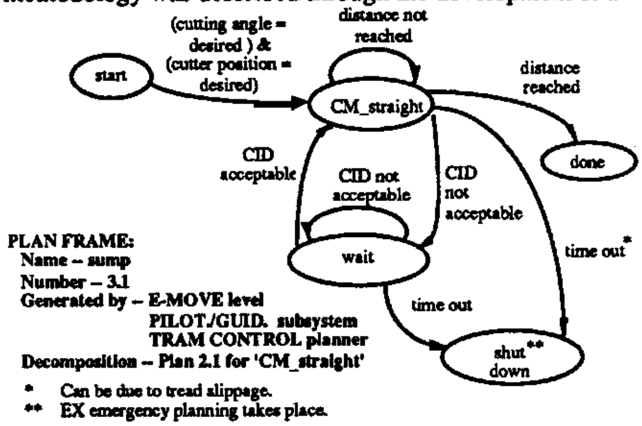


Figure 9: A Sump Plan

By following these steps, a system's capability, behavior and interaction with the external world can be described. An abstract high level goal for the system can then be logically decomposed, planned, and executed by the required equipment. The next stage of work includes designing software algorithms and requirements for the Job Assignment Modules, the Planners and the Executors to select and execute these activities, and the world model modules to support the task decomposition requirements.

7. Acknowledgements

The RCS methodology was originated by, and is evolving under, the leadership of Dr. James Albus, Chief, Robot Systems Division, NIST. The authors acknowledge Dr. Albus' comments and in particular, his permission to use Figure 1.

The authors also extend their appreciation to Dr. Christopher Jobes, Dr. George Schnakenberg, and the rest of the engineering staff involved, at the United States Bureau of Mines, Pittsburgh Research Center, as well as Mr. John Horst and Mr. Don Orser of the Robot Systems Division, National Institute of Standard and Technology, for their comments and suggestions. Also, Mr. Maris Juberts, who was with NIST and is now with NASA, for his early work on the functional hierarchy (Figure 2).

Our thanks to Cheryl Hutchins and Jhemon Lee for their careful editing of this paper.

References

[Al 82] Albus, J.S., McLean, C., Barbera, A., and Fitzgerald, M., "An Architecture for Real-Time Sensory-Interactive Control of Robots in a Manufacturing Environment," 4th IFAC/IFIP Symposium on Information Control Problems in a Manufacturing Technology, Gaithersburg, MD, October 1982.

[Al 88] Albus, J.S., "System Description and Design Architecture for Multiple Autonomous Undersea Vehicles," NIST Technical Note 1251, National Institute of Standards and Technology, U. S. Department of Commerce, September 1988.

[Al 89] Albus, J., Quintero, R., Huang, H., Roche, M., "Mining Automation Real-Time Control System Architecture Standard Reference Model (MASREM)," NIST Technical Note 1261 Volume 1, National Institute of Standards and Technology, U. S. Department of Commerce, May 1989.

[Hu 82] Huang, H., "Suppressing the Nonstationary Portion of the Transient Response in Digital Simulation," Master's Thesis, The University of Texas at Arlington, Arlington, Texas, 1982.

[Hu 90-1] Huang, H., "Hierarchical Real-Time Control Task Decomposition for a Coal Mining Automation Project," NIST Interagency Report NISTIR 90-4271, National Institute of Standards and Technology, U. S. Department of Commerce, March, 1990.

[Hu 90-2] Huang, H., "CASE Tool Implementation of RCS," NISTIR, National Institute of Standards and Technology, in process.

[Hu 90-3] Huang, H., "A Hierarchical Real-Time Control System for Use with Coal Mining Automation," Fourth Conference on the Use of Computers in the Coal Industry, the West Virginia University, Morgantown, West Virginia, June 1990.

[Jo 82] Joy Machinery Company, "Technical Service Manual, Joy 16CM Miner," Joy Technology Inc. Technical Publication, 1982.

[Pa 88] Penn State Mining Engineering Program, Course Notes on "Elements of Underground Coal Mining," The Pennsylvania State University, University Park, Pennsylvania, October 1988.

[Sch 89-1] Schnakenberg, G., "Underground Coal Mining Automation Research (Room and Pillar Mining) Justification and Research Program Plan," Ver. 1.1, Bureau of Mines, United States Department of the Interior, August 24, 1989.

[Us 68] U. S. Bureau of Mines, "a dictionary of mining, minerals, and related terms," 1968.

[Us 90] U. S. Bureau of Mines, BOM/NET Node Specifications, U. S. Bureau of Mines internal papers, June 1990.