Carnegie-Mellon University

THE INTELLIGENT MANAGEMENT SYSTEM
AN OVERVIEW

Mark S. Fox

Intelligent Systems Laboratory
The Robotics Institute
Carnegie-Mellon University
Pittsburgh, Pennsylvania 15213

THE
ROBOTICS
INSTITUTE

CMU-RI-TR-81-4

The Intelligent Management System
An Overview

Mark S. Fox

Intelligent Systems Laboratory
The Robotics Institute
Carnegie-Mellon University
Pittsburgh, Pennsylvania 15213

11 August 1981

Abstract This report describes the Intelligent Management System (IMS) project, which is part of the Factory of the Future project in the Robotics Institute of Carnegie-Mellon University. IMS is a long term project concerned with applying artificial intelligence techniques in aiding professionals and managers in their day to day tasks. This report discusses both the long term goals of IMS, and current research. It describes research in the modelling of organizations, job-shop scheduling, organization simulation, user interfaces, and system architecture. Examples of working systems are provided.

This research was supported by the Robotics Institute, Carnegie-Mellon University, and, in part, by the Westinghouse Corporation.

Table of Contents

List of Figures

1. Introduction

An important change is taking place in industry and industrial organizations. Products and services are increasing in complexity. The technology to produce them is becoming correspondingly more complex. Machine automation of physical processes progressively reduces the number of machine tenders and tedious low-skill jobs while simultaneously increasing the requisite number of high-skill, more challenging jobs. Highly skilled individuals cannot be interchanged as readily as the less skilled, leading to a more complex organizational structure with more demanding management tasks. Organizations, whether factories, ships, or hospitals, seem more difficult to manage. Managers typically respond to the situation by creating additional, specialist management jobs. Because of this new-found complexity, serious communication problems among the levels and subdivisions of management arise. Long range planning is often slighted because the complexity makes it difficult to pull all of the pieces together.

Classical research in the area of factory automation has been concerned more with production processes than with management. Yet, it has been observed that in many factories, white collar labor accounts for more than 50% of the cost of producing goods. In high volume, large batch-size production, such as lamp manufacturing, white collar labor costs account for much less than 50%. But in small batch size production, it can run as high as 75%. Contrary to popular belief, small batch-size production accounts for 50%-75% of the dollar value of goods produced in the United States. In metal-cutting, job-shop production environments, it has been found that only 20% of the time an order is in a factory, is it actually mounted on a machine. And during only 5%-10% of its time on the machine, are value-adding operations being performed. There are (at least) two approaches to dealing with this problem. The first is to discover new methods of producing products that do not suffer from these inefficiencies. The second approach is to increase the effectiveness of professional and managerial personnel. The project described herein is concerned with the latter. In the summer of 1980 we began the design and construction of what we call an *Intelligent Management System* (IMS). It is no longer the case that simple, single-function management systems, providing information access capabilities, are sufficient. Management systems must be more effective in the tasks they perform. They must integrate and communicate the knowledge and skill of the whole organization, making them available for management decisions. More importantly, they must aid profesionals and managers in carrying out tasks. Management systems must become more *intelligent*.

The goal of the IMS project is to aid supervisory personnel in their day to day decision making. It is concerned with both the type and level of functionality required by supervisory personnel, and with the cost of creating, maintaining, and adding new functionality. It is not sufficient to increase the effectiveness of one part of an organization, e.g., managerial, by increasing costs in another, namely programming and system support. Yet much of the software constructed today, while providing increased functionality, also requires increased programming support. Research and development must be concerned not only with functionality but with adaptability.

2. The Intelligent Management System

The goals of the Intelligent Managment System Project are to

1. Provide expert assistance in the accomplishment of professional and managerial tasks.

2. Integrate and coordinate the management of the organization.

To accomplish this the Intelligent Management System should:

- *Sense*: Automatically acquire state data. Sense the location of objects, state of machines and status of activities both on the plant floor and in supervisory departments.

- *Model*: Model the organization at many levels of abstraction. For example, machines, people, materials, orders, departments, need to be modelled in detail from both an attribute and a process view, including their interactions, authority, and communication.

- *Operate*: Provide expert assistance in the accomplishment of complex professional tasks within the organization.

- *Manage*:

    - Analyse and manipulate the model to schedule production and resource utilization, and answer short and long term state and planning questions. The system in this role is *passive* in that it responds to user initiated queries.

    - *Actively* monitor the organization and inform responsible personnel when important events occur. For example, when a machine break down occurs, not only is the foreman informed, but also maintenance, and the salesman who must inform the client that the order will be delayed.

- *Analyse-Optimize*: Analyse how the structure and the processing of the organization should be changed to further optimize some criteria such as cost, throughput, and quality.

Such a system, if it is to succeed in a business environment, must have the following characteristics:

Accessability: Interfaces to computer systems are usually idiosyncratic and difficult to learn and use. Also, systems that change require that their users be continually re-educated. Our goal for IMS is to enable all personnel to meaningfully communicate with it. The interface will gracefully interact with the user and provide guidance and help in deciding what the user needs.

Accountability: A major obstacle to computer acceptance is that users are unable to question how and why output was generated. Our goal is to construct an explanation system which will allow IMS to explain its actions at various levels of detail.

Adaptability: Currently, software systems are tightly coupled, requiring extensive re-programming whenever changes are required. This has resulted in the growth of the number of programmers needed to create and maintain computer systems. The goal of our research is to construct a theory of system design which will allow the users to modify the model, analysis, and processing functions without the aid of programmers. The end user introduces changes via dialogue.

Reliability: The system will not fail if one of its parts fails. No component is critical.

Reactability: The system, via its sensors and data monitoring will be able to detect changes in the organization. Detrimental changes are corrected if possible. Interested personnel are informed of the change, but receive only the information they require.

The construction of IMS requires the integration of a variety of technologies, many which require further research. While the total concept of IMS may take 10-15 years to reach fruition, many useful results will appear in the near future.

Research began in the spring of 1980. Three Westinghouse Corporation plants were visited[1]. Extensive analyses were carried out to determine problem areas and possible solutions. Three problems were chosen: job-shop scheduling, factory simulation, and process diagnosis. Systems were demonstrated for the first two applications in December of the same year. Research in all these areas and others, described later in the paper, continues.

In the following sections, what IMS looks like from a user's point of view is described. The remaining sections then describe the various areas of research underway in the project.

3. User's View

What the architecture of this system will look like, depends upon the type of organization. But there are many features that are organization invariant. Logically, the system architecture is distributed (figure 3-1). Most employee's will have a User Interface Process (UIP) that will act as an intelligent aide. A UIP is composed of a personal computer, graphics display, keyboard, microphone, and network interface (e.g., a SPICE machine (CMU-CSD, 1979)). The UIP will have either voice or typed natural language input. It will act as an aide in the sense that it will interpret and implement user requests and queries. All UIPs will be inter-connected via a communication network allowing them to cooperatively interact to solve problems and communicate information. The UIP will also carry out many of the employees well-structured tasks automatically. Each machine will have a Machine Interface Process (MIP) which monitors and controls it. It is also connected to the network, and can reply to queries and commands initiated by other MIPs or UIPs on the network. Lastly, there are Task Interface Processes (TIP). A TIP provides the focus for task management. It does most of the mundane task monitoring and control, freeing managers to do the more complex decision making tasks.

Examples of the types of interactions we would like to see are described in the following:

"Tell me when ..."

The marketing manager is under heavy pressure to get a rather large order out of the factory. He wants to be informed the minute it is shipped. He turns to his terminal and types the message: "Inform me when order X is shipped." His UIP translates the request into a rule "IF order X is Shipped

---

[1] A turbine component plant, a printed circuit board plant, and a flourescent bulb plant.

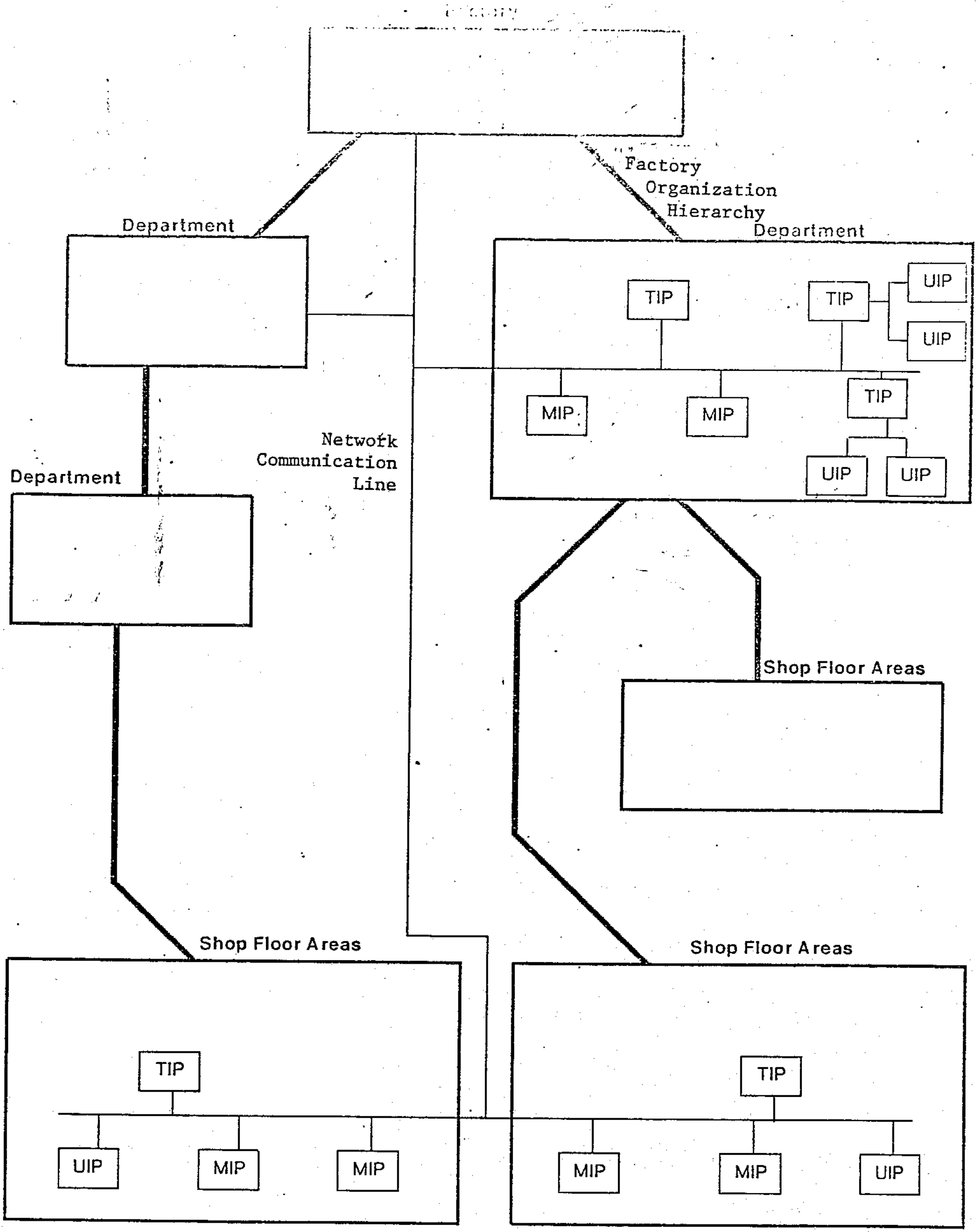

Figure 3-1: IMS System Architecture

THEN send message to manager Y", and sends it to the shipping TIP. The TIP monitors the system to determine when the rule condition occurs. When the order is shipped, the TIP interprets the rule, resulting in the shipping message being sent.

"You've got problems ..."

The milling machine breaks a tool while cutting a high priority order. The machine and order are damaged. The machine's sensors transmit the information to its MIP. The MIP analyses the problem, and shuts down the machine. It then informs the floor supervisor and the scheduler TIP of the breakdown; the scheduling TIP re-routes orders. A message is also sent to the maintenance TIP which allocates a maintanence person to fix the machine. Lastly, the MIP checks the importance of the order, and informs marketing and other personnel of the problem, if it affects their tasks.

"What if ..."

The manager in charge of production is considering the problem of a continually large back-log of orders. Should another machine be bought, or should the orders be subcontracted? He/she turns to his/her terminal and types in: "What are the effects on orders over the next six months if we buy another machine X?" The system then enters into a dialogue with the manager determining other information required to analyse the question. The UIP then scans the system wide functions that may help answer the problem. It finds a simulation module that can analyse structural changes in an organization. It gathers the initialization data and alters the factory model. It then runs the simulation, analyses the output and provides the manager with the answer, and further explanations.

4. User Requirements

In determining the functionality IMS is to exhibit, over 30 managers from three plants in the Westinghouse Corporation were asked to record the types of questions they frequently are faced with during the performance of their jobs. The following is a sample of their replies:

- What is the effect of changes in engineering specifications: designs, materials, process specifications, etc?
- What is the effect of defective work?
- What is the proper inventory level at various levels, i.e., raw, work in progress, finished parts, etc?
- What is the productivity level of employees in all categories and how does it compare to expected levels and trends?
- When should preventive maintenance be scheduled for a particular facility?
- What if the jig grinder goes down with maintenance problems?
- Based on downtime, cost of maintenance and cost of replacement parts, how do I determine when to buy a new piece of production equipment?

- What equipment should be added?

- Charlotte moves one (1) BB72 rotor two (2) months ahead of schedule and Lester moves two (2) BB22 rotors back in schedule.

  1. How are all promised dates for next six months affected?

  2. What manpower changes are needed by cost center to accomplish changes?

  3. What material delivery changes need to be made?

- I am a tool planner. How do I decide what tools are required to manufacture a new blade?

- How do I select the process to be used? What if the shop changes the plan and needs to perform a particular job on a different machine from the standard?

- Given orders and time frame information, it is difficult to plan whether to produce the product in-house or to subcontract it, where to acquire materials, or when to schedule production.

- Material X is not available, what schedule changes are required and develop new schedule?

- Internal Communication. Not enough information concerning the current state of production on the floor, orders in process, problems etc. is communicated quickly and succinctly to interested parties within the plant.

- Engineering Feedback: changes to products or tools by the engineering division are not adequately communicated and coordinated with changes taking place in the factory.

- Tooling Problem. The plant contains about thousands of tools scattered throughout Operator and machine time is wasted searching for tools necessary to carry out a task. Tools are "lost" by losing track of their location. Scheduling cannot be accurately done without knowledge of tool availability.

- Machine, materials, schedule, maintenance, personnel Status.

- If I have a database containing lamp prices, sales volume, ics (internal product cost) transportation cost, overhead allottment, .. for all lamp types. What will the profit be if I change transportaiton, or change the source of manufacturing.

- A priority of the entire division is to reduce the working capital. Amount of products in the channels, work-in-process, etc.. Want a system to speed the product through the distribution channels. High inventory, in transit, in process cost.

- Reduce the inventory levels by upgrading the forecast, reorder quantities, and lamps on demand.

- Analyse the effect of increasing throughput on profitability: include speed up of machines, increases shrinkage, wear and tear, greater inventory levels, etc.

- Analyse the effect on profitabililty of the increased sales of a certain product, including individual product margins, product promotions and scheduling.
- Keep records of: production, shrinkage, down time, absenteeism, safety purchases, perpetual inventory of materials and machine parts.
- Analyse down time by machine part number failure, shift, and by operator.
- Predict machine failures and prescribe preventative maintenance by sound viabration, machine timing, shrikiage trends, and end of normal part life.
- A program to determine correlations between gas, fill, temperature, stack up, and shrinkage.

The questions and problems listed above can be categorized into three broad classes:

1. Information access and communication.
2. Professional tasks.
3. System analysis.

Existing vendor systems can handle most of the information access problem, e.g., product tracking, reporting, billing, etc., by providing databases and appropriate access functions. But for many of the managerial level questions and problems, systems do not exist.

The following sections describe our first steps towards constructing an Intelligent Management System. Our approach has been to construct a solid system architecture and organization modelling base upon which to construct interesting functionality, whose choice has been motivated by the needs found in the plants visited. The problem of modelling organizations is discussed first because of the foundational role it plays in IMS. Next, sections on organization analysis and management are presented. They discuss the various analysis and management functions in IMS. How users interact with IMS is described in the section on User Interfaces. And finally the architecture of IMS is described in the section on integrating distributed systems.

5. Organization Modelling

5.1. Introduction

The purpose of organization modelling is to provide the information base upon which intelligent processes rely. What the contents of a model should be, and how it is represented is dependent upon the processes that use it. It is safe to say that an "Intelligent Management System" will require at least as much information as humans. The richness and variety of this information cannot be found in the databases of current management information systems. Nor is the form of the organization model related to current notions of database structures.

For example, a simulation system requires knowledge of existing processes including process times, resource requirements, and its structural (routing) relation to other processes. It must also know when routings for products are static, or are determined by a decision process such as a scheduler. In the latter case, it must know when and where to integrate the scheduler into the simulation. If IMS is to generate the sequence of events to produce a new product, it must have knowledge of processes (e.g., machines) which includes the type of processing it can do, its operating constraints, the resources it consumes, and its operating tolerances. If data is to be changed in an interactive, possibly natural language mode, IMS must have knowledge of generic processes such as machines, tasks, and departments if it is to understand the interaction. It must also know what information is important and how it relates to other information in order to detect missing information and inconsistencies. Hence, the organizational model must be able to represent object and process descriptions (structural and behavioral), and functional, communication and authority interactions and dependencies. It must represent individual machines, tools, materials, and people, and also more abstract concepts such as departments, tasks, and goals.

Consider a process model of a florescent lamp production machine group. The purpose of a process model is to represent each physical process and the causal relations which link it with other processes. For example, the Lehr[2] process comprises of hundreds of subprocesses concerned with bulb grasping, positioning, heating, cooling, etc. Each is sequentially related with others in time. The performance of each is related to the performance of previous subprocesses. Each subprocess is described in terms of how it physically performs, its three dimensional movement, what and how it may grasp and heat an object, what object it expects, operating constraints, etc. The scope of the description is limited only by the uses to be made of it, and the information available.

One use of such a model is for machine diagnosis. Engineers spend a great deal of time watching a malfunctioning machine to determine what has gone wrong and what variables to alter to improve performance. One of the major stumbling blocks in the systematic analysis of such systems is the unavailability of process instrumentation for data collection. Yet the availability of the data solves only half of the problem. The other is the automatic analysis of data to find relations between system parameters and productivity. Statistical correlations are only a small part of the analysis. Statistics alone can only *suggest* relations. *Understanding* whether the relation is valid requires a thorough knowledge of the process itself and all its interactions. This cannot be performed without a sufficiently rich model of the process describing physical, functional, and time relations.

Another use of a model is to provide process cost analysis. Given a model, questions about the resource consumption and production of each process and subprocess can be answered. The individual process descriptions can be integrated across the group to provide summary cost information. But more importantly, because the model represents not only the process but the effects and relations among processes, questions related to process alterations can also be analysed. For example, what will the effect on cost be of changing a Lehr process to use microwave heating. And from a diagnosis point of view, we would want the system to determine whether the change would have any significant effect on shrinkage[3]. For example, the temperature of the glass may be lower

---

[2] Baking lacquer out of lamp phosphor.

[3] loss through defects.

after microwaving, hence the following process, end sealing, may not have glass at a high enough temperature.

In summary, the modelling system should provide:

1. A rich source of information.
2. A context within which all IMS modules can interact.

But if it is to be used in a changing environment and by non-programmers, it should also provide the following features:

Accessability: the model user should be able to easily peruse the model to determine its attributes and structure.

Extensibility: the model should be alterable by its users.

Consistency: when a model is created or extended, the modelling system should be able to determine when the model contains inconsistent information.

5.2. Basic Modelling System

The above discussion presents a case for a modelling system that supports a variety of functionality such as simulation, cost analyses, and process diagnosis, and a flexible user interface. Therein lies the question: does such a modelling system exist? Traditional, computer-based modelling systems fall into four categories:

1. Mathematical.
2. Discrete event, facility based.
3. Continuous.
4. Arbitrary program.

In looking at these modelling systems, we found:

- They are too problem specific, hence inflexible.
- The modelling techniques are difficult to learn by managers and engineers who are the prime users.
- Alteration may require substantial change to the model and related systems.
- Once constructed, the models are difficult to understand, peruse and verify.

Each use of the Intelligent Management System has at its core, a model of the organization. This model is shared by all subsystems to achieve their tasks. The modelling system provides the following features:

- The model is composed of declarative objects and relations which match the users conceptual model of the organization.

- The model incorporates a variety of representational techniques allowing a wide variety of organizations to be modelled (continuous and discrete). And it is extensible, allowing the incorportation of new modelling techniques.

- The user interactively defines, alters, and peruses the model.

- The model can be easily instrumented. For example, the model can be diagramatically displayed on a color graphics monitor at different levels of abstraction. The complete organization, or parts thereof, can be viewed with summaries (e.g., queue lengths, state).

- The modelling system is simple to learn to use because the modelling tools match the concepts people use to think about problems.

The modelling system is based on the knowledge representation system SRL (Fox, 1979a; 1981). SRL is a schema (frame-like) based representation language (Minsky, 1975; Bobrow & Winograd, 1977). One of its primary features is its allowance for the creation of user-defined (inheritance) relations. From a modelling point of view, this allows the model builder to define both objects and relations amongst objects that display inheritance and mapping properties that closely match the model building view of an organization. A schema describes the attributive, behavioral, and relational characteristics of a concept.

```
{{ Machine

        CAPACITY:
        QUEUE:
        OPERATOR:
        CONTENTS:
            Restriction: (TYPE #is-a #product)
        LOAD:
            Restriction: (SET (TYPE #is-a #rule))
            Default: #load-rule
        UNLOAD:
            Restriction: (SET (TYPE #is-a #rule))
            Default: #unload-rule
   }}
```

Figure 5-1: Machine Schema

In figure 5-1 the basic schema for a MACHINE is defined having many slots, e.g., CAPACITY, QUEUE. None of them are filled with values or rules, though restrictions and defaults for the values of some of the slots are specified. Figure 5-2 defines a CONTINUOUS-MACHINE which works much like a pizza

```
{{ Continuous-Machine
   { IS-A Machine
       USED-CAPACITY:
       LOAD: { #INSTANCE # rule
                  IF: USED-CAPACITY < CAPACITY
                  THEN: add 1 to USED-CAPACITY, add object to CONTENTS.
                  ELSE: add object to QUEUE }
       }
   }}
```

Figure 5-2: Continuous-Machine Schema

oven, it can be continuously filled up to capacity. A **Continuous-Machine** IS-A **Machine.** The IS-A relation between the two schemata allows **Continuous-Machine** to inherit attributes (slot names) and their values from the **Machine** schema. The LOAD slot defines the behavior of the machine when a load event ocurrs. The loading rule tests whether the machine has capacity, if so the object is placed in the machine, otherwise it is queued.

The **Machine** and **Continuous-Machine** schemata are generic schemata that form part of the basic modelling system. The system contains a variety of basic schemata which the model builder can use to model an individual organization. An organization model is constructed by instantiating the basic schemata with appropriate attribute values, e.g., capacity, and possibly, new behavioral rules. Schemata are structured (linked) through a user extendable set of relations: IS-A, INSTANCE, PART-OF, etc. For example, a circuit board baking oven (figure 5-3) can be instantiated as an INSTANCE of a **Continuous-Machine.** It has a CAPACITY of 10, and inherits its loading rule from **Continuous-Machine.** It is also part of a **Work-Area** in the plant called the **Baking-Shop.** The PART-OF relation allows the inheritance of locational information.

```
{{ PcOven

   { INSTANCE Continous-Machine
       CAPACITY: 10     }

   { PART-OF Baking-Shop  }

   }}
```

Figure 5-3: PcOven Schema

5.3. Model Accessability and Extensability

Current management information systems require the use of programmers when changes are made to the organization model. While such an approach is fine for domains where the model changes seldomly, the dynamics of a factory organization require continual updating of the model; both parametric and structural changes are constantly occuring. In lieu of employing a brigade of programmers to implement changes, the alternative is to construct a model acquisition system that allows the person initiating the change, to directly inform IMS of the modifications.

The IMS modelling system currently provides the following functionality:

- ***Accessability***: The user may interactively view each schema in the model using a variety of schema printing functions. Relational hierarchies can also be displayed. Color graphic display of the model is supported at various levels of abstraction.

- ***Extensibility***: An interactive schema editor allows the user to create and alter schemata in the model.

- ***Consistency***: A first version of a model consistency language has been developed. A consistency checker uses the consistency specifications to check a model and reports inconsistencies to the user.

Though the above mechanisms provide a good user interface to the modelling system, our ultimate goal is to allow managers to alter the model directly. By means of a natural language interface, the manager should be able to describe to IMS changes that have ocurred in his area of responsibility. To achieve this, the modelling system interface must:

1. Determine what part of the factory model is affected by the new information.

2. Check to see if the new information is consistent with what it already knows about the factory.

3. Determine the effect of the change on other parts of the model, and make the appropriate changes.

4. Query the manager when inconsistencies appear in the reconciliation of the new information with the existing model.

A natural language understanding and discourse modelling system to support model acquisition, factory layout, and job-shop scheduling is currently under development.

5.4. Flow-Shop Modelling

A complete model of a printed-wire-board (PWB) bareboard production plant[4] has been constructed to the machine level (August, 1980). Figure 5-4 provides a glimpse of part of the relational structure of the model without each schema's attributes. You will note that information on

---

[4] under design by the Westinghouse Corporation.

how to simulate and display the factory is embedded in the model. The model also includes schemata for operations, process sequences (lineups), products, personnel and others. The model has been used directly by our simulation system and to display the factory on a color graphics monitor. Figure 5-5 displays the complete layout of the plant. Each work-area is color coded according to type of processing. Under each name is the number of orders in process and queued for the work-area. At 1:15 there are 9 orders in the inspection area. The UIP allows the user to specify what part of the plant to display. Figure 5-6 shows a blow up of the inspection area. At 1:36 there are 6 orders in the inspect1 station and 8 orders in the touchup station.

Process flow is defined by a production (operation) lineup defined for each product type. When an order is unloaded from an object such as a machine, work-area, etc. the next operation is determined by information inherited by the object. For some objects access to a scheduling system is inherited via the PART-OF relation, for others the next operation is defined directly due to physical coupling of operations.

5.5. Job-Shop Modelling

A model of part of a turbine component production plant[5] has also been constructed to support simulation (section 6.2) and job-shop scheduling functions (section 7.2). The model contains information about machines, products, tools, work-centers, labor and cost data, and factory layout. Much of the schemata used to model the circuit-board plant were used in the turbine plant model. In some cases additional slots (attributes) were specified, e.g., cost data, and operation sequences were expanded to operation graphs to include alternate processsing routes in the plant. Job-shop scheduling was provided by inheriting a different scheduling system.

5.6. Machine Modelling

Our third modelling project, currently underway, is to construct a model of a flourescent lamp production line; including enough detail to support machine process diagnosis (section 7.4). This model focusses on the pre and post-conditions of individual machines in the production process, and on the causal relations that exist between and within the machines.

6. Organization Analysis

6.1. Introduction

The effectiveness of an organizational is determined by its ability to deal with environmental uncertainty and complexity. Environmental uncertainty and complexity coupled with organizational uncertainty and complexity results in sub-optimal, and even sub-satisficing organizational behavior. To produce requisite behavior, an organization must analyse its environment and adapt. But it is too often the case that management lacks the time or ability to carry out the analysis; hence the organization internalizes more rigid behavior. One of the most important but least understood aspects of an organization is its ability to adapt. Much of Organization Theory has been concerned

---

[5] Westinghouse Turbine Component Plant, Winston-Salem NC.

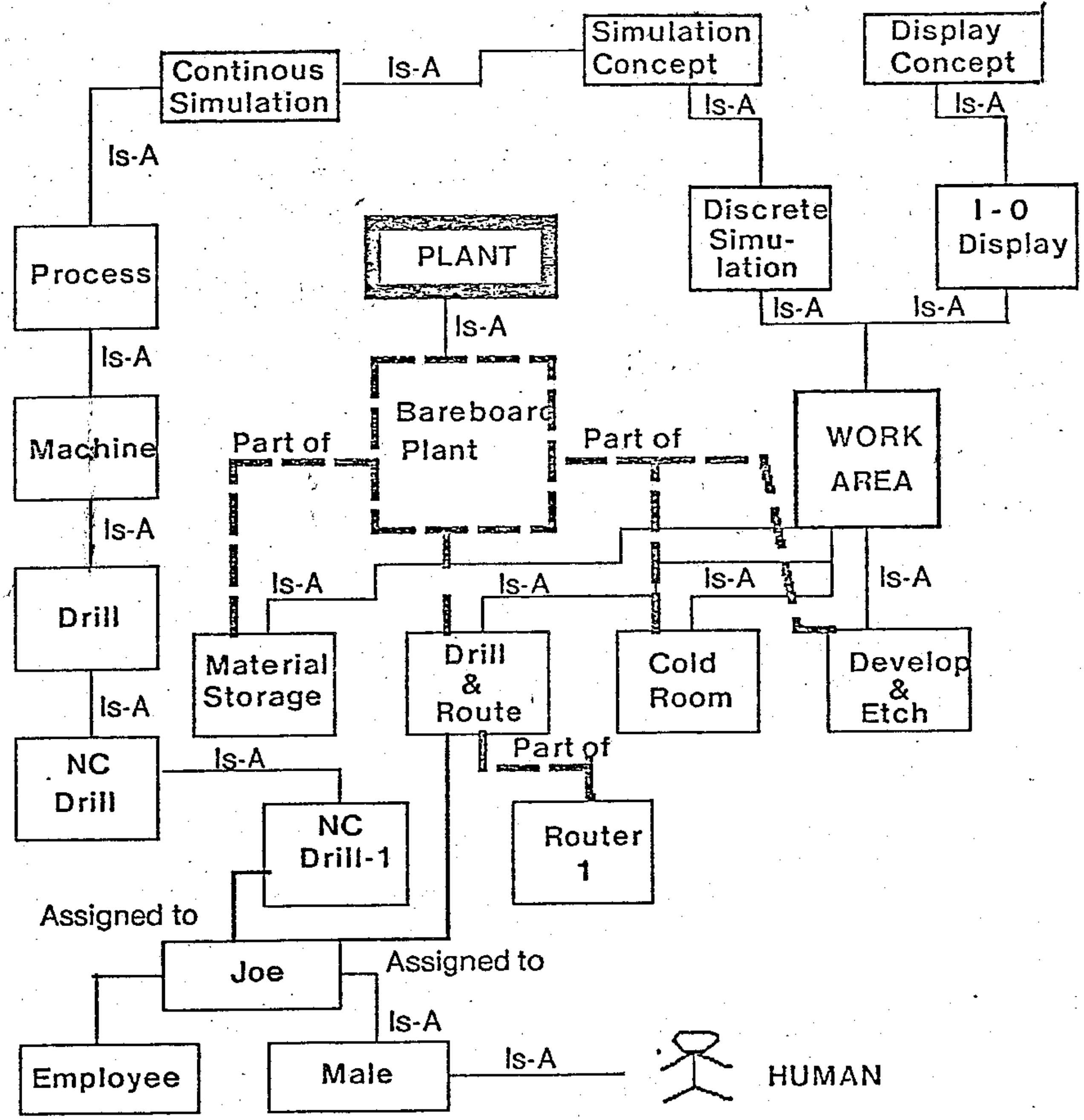

Figure 5-4: Flow-Shop Partial Model

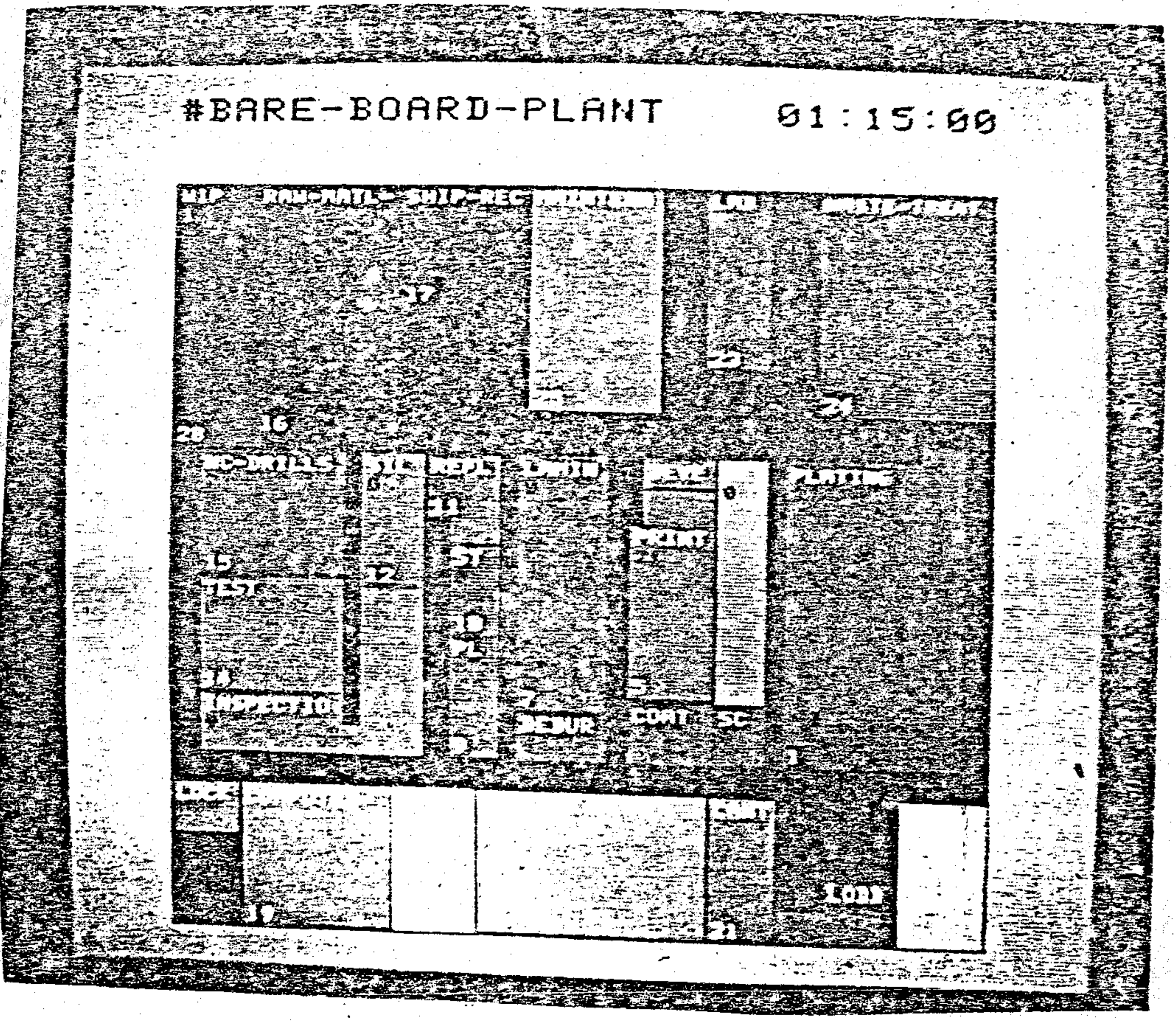

Figure 5-5: Monitoring the Factory

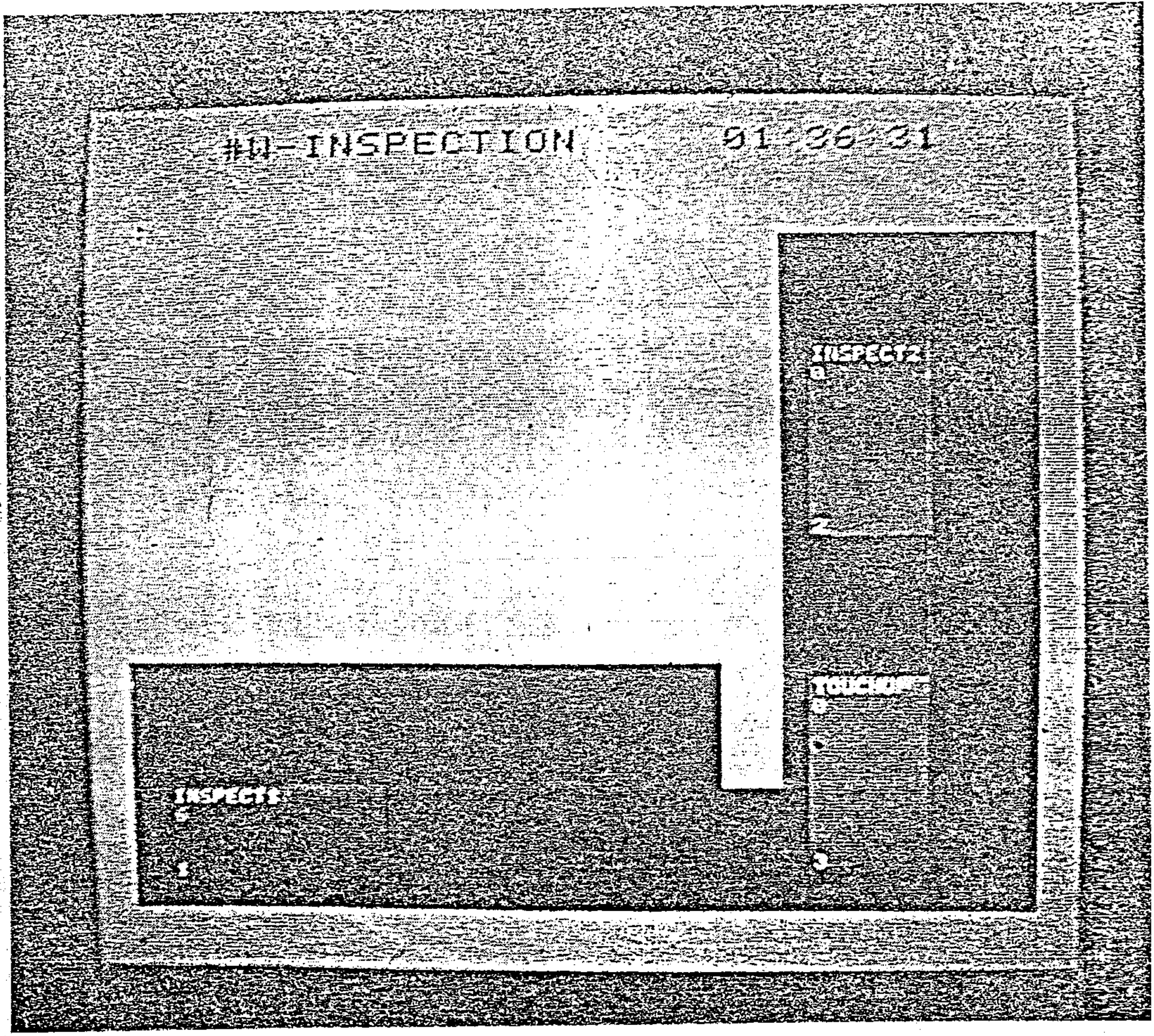

Figure 5-6: Monitoring the Inspection Area