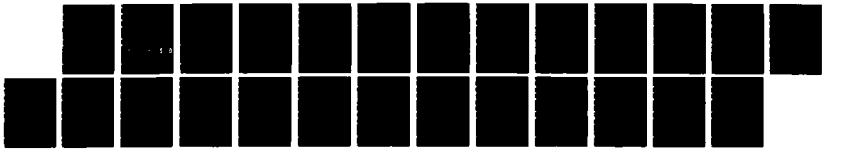
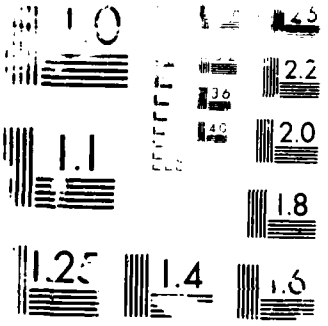
AD-A189 113 ON FAULT TOLERANCE IN MANUFACTURING SYSTEMS(U) MARYLAND 1/1
UNIV COLLEGE PARK INST FOR ADVANCED COMPUTER STUDIES
P R CHINTAMANENI ET AL OCT 87 UMIACS-TR-87-93
UNCLASSIFIED N00014-87-K-0463 F/G 13/8 NL




1.0
1.1
2.2
2.0
1.8
1.4

(4)

AD-A189 113

UMIACS-TR-87-53
CS-TR-1939

October, 1987

# On Fault Tolerance in Manufacturing Systems†

Prasad R. Chintamaneni
Pankaj Jalote
Yuan-Bao Shieh‡
Satish K. Tripathi

Department of Computer Science and
Institute for Advanced Computer Studies
University of Maryland
College Park, MD 20742

SELECTE J.

# COMPUTER SCIENCE TECHNICAL REPORT SERIES






# UNIVERSITY OF MARYLAND
## COLLEGE PARK, MARYLAND
## 20742

(4)

UMIACS-TR-87-53
CS-TR-1939

October, 1987

# On Fault Tolerance in Manufacturing Systems†

Prasad R. Chintamaneni
Pankaj Jalote
Yuan-Bao Shieh‡
Satish K. Tripathi

Department of Computer Science and
Institute for Advanced Computer Studies
University of Maryland
College Park, MD 20742



## ABSTRACT

An important issue in manufacturing systems involved in factory automation is the support for fault tolerance. This paper describes the hierarchical physical and control structure of manufacturing systems and proposes a hierarchical model for fault tolerance support. The usefulness of the hierarchical fault tolerance model is shown in the manufacturing system domain and the main issues involved in the general applicability of the model are discussed.



A-1

† This work is supported in part by contract N00014-87-K-0463 from the Office of Naval Research to the Department of Computer Science, University of Maryland.

‡ Yuan-Bao Shieh is with the Department of Computer Science, University of Maryland, Baltimore County, MD 21228.

# On Fault Tolerance in Manufacturing Systems[1]

*Prasad R Chintamaneni, Pankaj Jalote, Yuan-Bao Shieh[2], and Satish K Tripathi*
Department of Computer Science and
Institute for Advanced Computer Studies,
University of Maryland,
College Park, MD 20742.

## Abstract

An important issue in manufacturing systems involved in factory automation is the support for fault tolerance. This paper describes the hierarchical physical and control structure of manufacturing systems and proposes a hierarchical model for fault tolerance support. The usefulness of the hierarchical fault tolerance model is shown in the manufacturing system domain and the main issues involved in the general applicability of the model are discussed.

## 1. Introduction

Increasing competition, higher production costs, and greater demands for diverse and customized products are some typical problems that plague manufacturing businesses. Factory automation aimed at efficient, cost effective, and flexible production is thus a major concern of manufacturing systems. A variety of automated equipment including robots, numerically controlled (NC) machines, automated guided vehicles (AGV), and intelligent sensors have been developed for use at the factory floor. New computer based technologies like Computer Aided Design (CAD), Computer Aided Planning (CAP), and Computer Aided Manufacturing (CAM) are being incorporated in the manufacturing design, decision-making and production. The main motivation of manufacturing systems is the development of flexible, automated, integrated, and efficient manufacturing environments. Such environments are typically composed of a variety of computer hardware and software units coordinating with sophisticated automated equipment on the factory floor.

---

[1]This work is supported in part by contract N00014-87-K-0463 from the Office of Naval Research to the Department of Computer Science, University of Maryland.

[2]Yuan-Bao Shieh is with the Department of Computer Science, University of Maryland, Baltimore County, MD 21228.

One major concern in achieving this goal is the need for rapid and reliable communication among the wide variety of computers and automated equipment involved in the manufacturing process. General Motors in 1980 enlisted the support of its suppliers and started the specification of a network standard for factory communications called Manufacturing Automation Protocol (MAP)[GM85][GM86]. MAP complies with the definition of the Open Systems Interconnection (OSI) Reference model[ISO85], as defined by the International Standards Organization. To support technical information interchange and office automation, Boeing Industries started the development of a network standard for office communications called Technical Office Protocol (TOP)[TOP87]. TOP also complies with the definition of OSI reference model. The specifications of MAP and TOP networks are designed to allow a simple interface between the two. The composite networks, dealing with office automation and factory automation (based on TOP and MAP respectively), will define a set of communication standards that should provide reliable and rapid communication among the wide variety of equipment in a manufacturing system.

The protocols focus on the aspect of reliable communication in manufacturing systems. Another major issue for such systems is fault tolerance. Fault tolerance implies the ability of the system to continue performing certain activities even after the failure of some system components (hardware or software). The probability that one or more components may fail in a distributed system increases with the number of components that comprise the system. Clearly, the situation where failure of one component disables the entire system is unacceptable in the manufacturing systems domain. Furthermore, the cost of lost production while manual repairs are ongoing may be unacceptably high. Fault tolerance can be therefore be considered as a basic requirement for manufacturing systems.

This paper addresses the issue of providing support for fault tolerance in manufacturing systems. Physically, manufacturing systems can be viewed as large distributed systems consisting of interconnected networks of a number of computing nodes and automated devices. Functionally, manufacturing systems are involved in a variety of activities including planning, design, process

scheduling, tool and materials handling, product assembly, process quality control and inventory control. These activities are closely related to one another. Faults encountered during the processing of any of these activities will have repercussions on all the others, and hence should be handled gracefully. Manufacturing systems often have to satisfy constraints like meeting production deadlines, holding inventory levels at acceptable levels, and using the available resources optimally. Faults in the system can cause these constraints to be violated. Support for fault tolerance, together with a reasonable performance, is therefore essential in manufacturing systems.

Several techniques have been proposed in literature addressing the issues of fault tolerance in distributed systems[PP83][BB83][TS84][JB86][KT87]. However, little work has been done to address the specific issues of fault tolerance in manufacturing systems. We propose a model for supporting fault tolerance in manufacturing systems. We first identify a unique set of features characterizing manufacturing systems. These features of manufacturing systems denote a special subset of distributed systems. The proposed model for fault tolerance is applicable to any distributed system belonging to the identified subset.

The rest of the paper is organized as follows. Section 2 describes the characteristics and related issues of manufacturing systems relevant to fault tolerance. In section 3, the fault tolerance issues, requirements and constraints in manufacturing systems are discussed. The hierarchical model for fault tolerance is described in section 4. Support for fault tolerance in manufacturing systems is discussed in the context of the hierarchical model. In section 5, the design issues involved in hierarchical fault tolerance are summarized. Conclusions and future work are addressed in section 6.

## 2. Manufacturing Systems

Manufacturing systems define a specific subset of distributed computer systems. In section 2.2 we present the features that are characteristic of manufacturing systems. Section 2.3 discusses the control issues and section 2.4 describes the communication issues in manufacturing systems.

## 2.1. Characteristics of manufacturing systems

The following features typically characterize manufacturing systems:

**Distributed nature:** Manufacturing systems are inherently distributed in the sense that one cannot assume a single thread of control.

**Hierarchical structure:** The control structure of the different nodes in a manufacturing system is hierarchical (see Section 2.2).

**Event ordering:** The concurrent execution of processes in a manufacturing system may have precedence relationships associated with certain task sets. Thus synchronization between processes is an issue in manufacturing systems.

**Real time constraints:** Some segments of the network comprising the manufacturing system operate under real time constraints. For example, the automated equipment at the factory level (i.e., sensors, AGV's, robots, NC machines and PLC's) typically have real time controls. Other segments of the network may not require real–time response.

**Heterogenous nodes:** An assumption frequently made in supporting fault tolerance is that a process can execute at any operational node of the distributed system. In certain segments of the manufacturing system networks, this assumption does not hold. For example, automated equipment at the factory level perform specialized functions that cannot be interchanged. Such segments should be identified and appropriately supported.

**Varying consistency requirements:** Consistency maintenance is an important issue in a distributed system. From a hierarchical viewpoint, consistency requirements at different levels of a manufacturing system hierarchy are different. For example, consistency requirements are stringent at the office level and practically non–existent at the factory level.

**Heterogenous traffic patterns:** In the hierarchical model of a manufacturing system, communication traffic intensity at different levels of the hierarchy is different. For example, the bursty traffic at the office level is typically composed of large data and graphics file transfers. At the factory level, equipment operating under real time constraints communicate via short and frequent command exchanges.

**Hostile environments:** In proposing a model for fault tolerance, a common assumption made is that of an error free communication channel. In the factory floor of a manufacturing system, the communication error rate can be high because of high electromagnetic interference, high temperature and possible sparking. Manufacturing systems include some communication channels that are susceptible to transient errors.

## 2.2. Control Issues in manufacturing systems

Distributed manufacturing systems are typically modelled to have a hierarchical control system[AC86]. There are four main levels in the control hierarchy (see Figure 1).

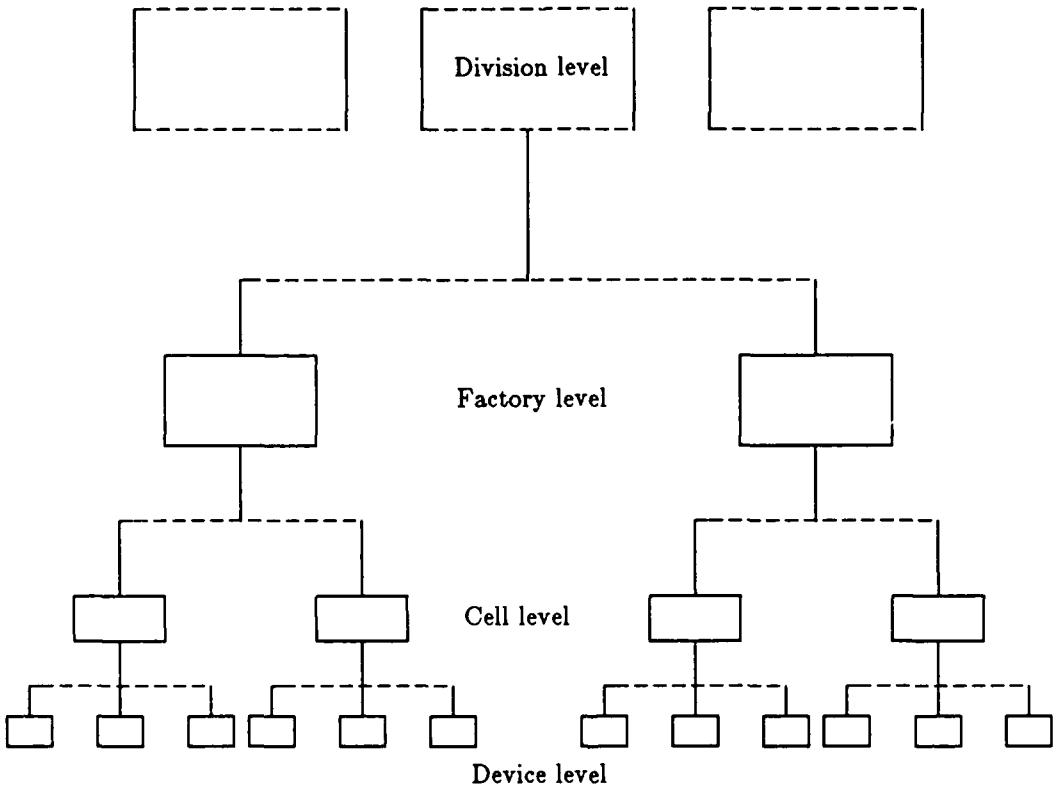



Figure 1: Control Hierarchy in Manufacturing Systems

**Device control level:** This is lowest control level of the hierarchical networking scheme. A device controller is attached to control the operation of each automated device on the factory floor. The various devices include robots, NC machines, sensors and AGV's. The device controllers are generally supplied by different vendors and therefore not compatible with one another. A device program therefore cannot be arbitrarily executed on any device controller.

**Cell control level:** This is second level of control. The machines at this level control the manufacturing devices within a cell. A cell controller consists of a combination of a programmable controller and a processing unit. The task management and activity coordination within a cell are managed by the cell controllers.

**Factory control system level:** This is the main control level of a plant network. The operation of individual manufacturing cells, the storage, retrieval and assembly systems

are managed at this level. Inventory management and work assignment are the responsibility of the factory control subsystem. The factory subsystem also maintains a database of the individual device programs of the factory front–end devices. Downloading of appropriate device programs to individual cells is the responsibility of the factory control subsystem.

**Division control system level:** This is highest level of control. Nodes at this level are connected to divisional networks. Global management and administration, interfacing individual factory control systems and inter–factory communications are the control functions of this level.

### 2.3. Communication issues in manufacturing systems

From the hierarchical view, different levels in the manufacturing system have different information and control requirements. The environmental interferences are higher and and the time constraint requirements are more critical for the lower levels (i.e., device control level and cell control level). Thus reliability and response time are more important at the lower levels. The control and application programs at the device and cell levels communicate via short, frequent and time–constrained message transfers. To support these characteristics, high speed carrier band coax communication networks are used at the lower levels. The network functions are performed by time critical protocols (e.g., token bus[IEEEa]) that guarantee message transfer in finite time durations. At the higher levels in the hierarchy, (i.e., factory control level and division control level) the reliability and real–time constraints may be less critical. The communication characteristics include bursty and high volume traffic and non–critical message delivery requirements. To support these characteristics and utilize the communication channel effectively, broadband networks with appropriate protocols (CSMA/CD[IEEEb]) may be used at these levels. The broadband network forms a network backbone for the entire factory floor. Communications with other backbones are achieved through wide band links at the division level. The interconnections between the different segments of the manufacturing system at the various levels of the hierarchy is through the use of intermediate systems like gateways, routers and bridges. A general communication model for manufacturing systems, consisting of a set of computer systems communicating over local area networks (LAN's), is shown in Figure 2.

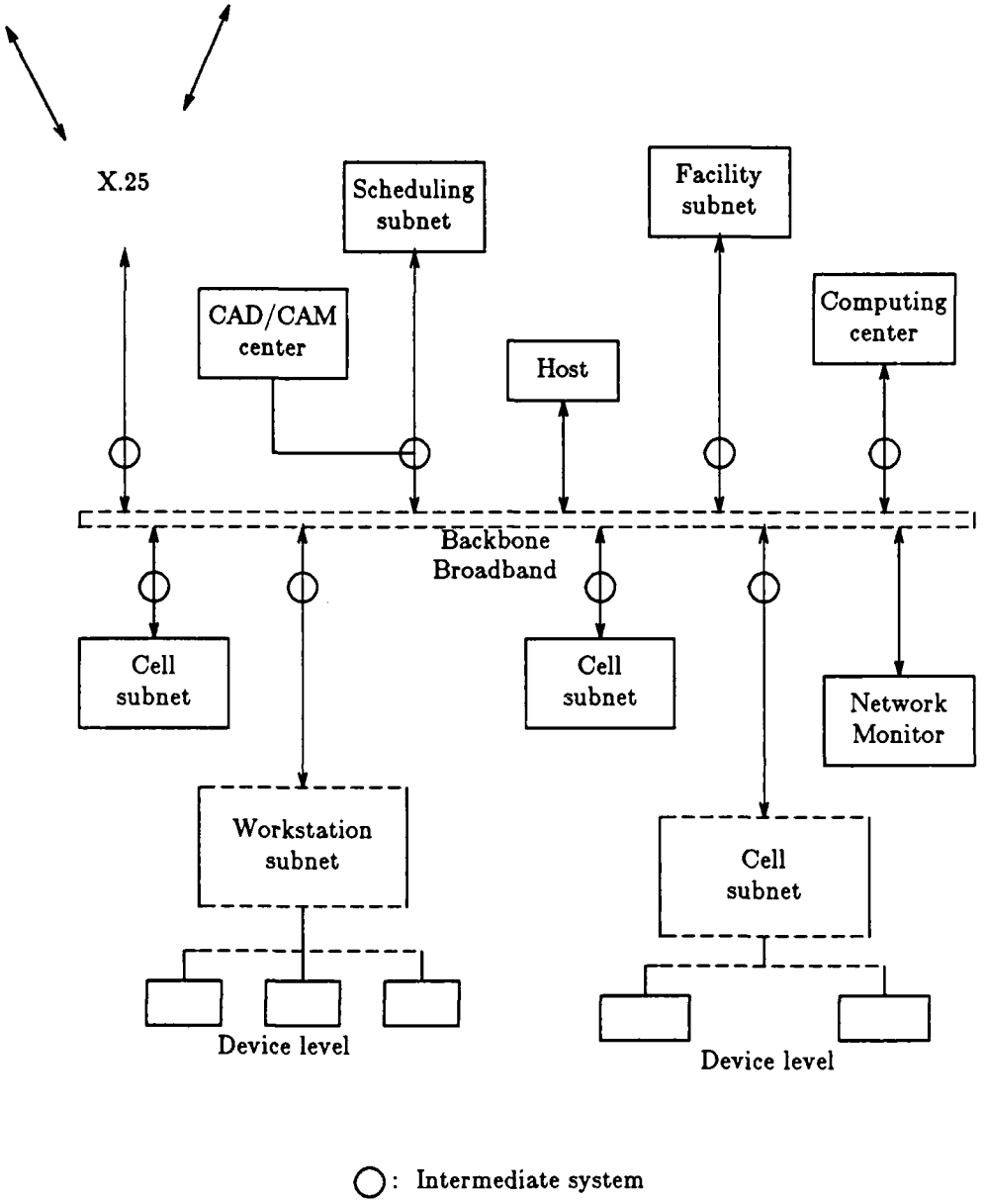



Figure 2: Communication Structure of Manufacturing Systems

### 3. Fault tolerance issues in manufacturing systems

The primary goal of a manufacturing system is efficient control and smooth operation in a factory environment. The system comprises of a number of components including computing and automated devices. The basic goal of fault tolerance in manufacturing systems is to ensure

proper control and operation in the system despite the failure of some components.

As explained in section 2, different segments of a manufacturing system have different characteristics. Examples of these characteristics include communication traffic intensity, real-time requirements, node homogeniety, communication channel reliability, and consistency requirements. In addition, the fault tolerance requirements at different segments may be different. Examples of these requirements include the fault resiliency (number of failures to be sustained), and the fault coverage (types of faults to be sustained).

Given the characteristics of the segment in the manufacturing system and given the fault tolerance requirements for the segment, a fault tolerance scheme for the segment has to be devised. Different fault tolerance schemes have different amounts of overhead associated with fault tolerance support. For example, consider the two main schemes for supporting fault tolerance in a distributed system: the centralized[PP83] and the distributed[BB83][JB86][KT87] approaches. In the centralized approach a single fault manager, located at a node in the system, is responsible for fault detection and recovery. This approach is cheap involving little overhead. It has a low reliability because a single point failure can disrupt the system. In the distributed approach, several nodes cooperate in providing support for fault tolerance. The approach is more expensive involving greater overhead. However, it has a higher reliability. In the manufacturing systems context, in segments that have real-time constraints, low communication traffic intensities, and low consistency requirements, the centralized approach is preferable to the distributed approach for fault tolerance support. In segments that have high communication traffic intensities, high consistency requirements, and no real-time constraints the distributed approach is preferable to the centralized approach. Specific techniques in each of these approaches can further be evaluated based on the segment characteristics and requirements.

Since manufacturing systems are organized hierarchically, the model used for supporting fault tolerance should also be hierarchical. As in any hierarchy, the view of the system is different at different levels. For fault tolerance, this means that the system characteristics and the

reliability and consistent requirements at different levels may be different. A fault tolerance model should be flexible enough to allow different techniques to be employed at different levels. Fault tolerance at each level can then be tailored according to the services to be supported at the level, and the consistency and reliability requirements at the level, based on the system characteristics of the level.

The major parameters to be considered at each level are reliability, consistency and acceptable overhead for supporting fault tolerance. The reliability and overhead are affected by the choice of either a centralized or a distributed approach and in addition to the choice of a specific technique in either of the approaches. Stringent consistency can complicate fault tolerance considerably and adds to overhead. Since the smooth control and operation of the factory level is the basic goal of manufacturing systems, the support for fault tolerance should be driven *bottom-up* from the factory or device level towards the topmost division level. Each level must either be capable of providing the required support needed at that level, or request additional support from levels higher than the current level.

## 4. A Hierarchical Fault Tolerant Model

Given a distributed system that can be modelled at the physical and control level hierarchically, we propose a hierarchical model for fault tolerance support. In section 3.1, the hierarchical fault tolerance model is described. The applicability of the model to manufacturing systems is described in section 3.2.

### 4.1. Model Description

Hierarchical fault tolerance models for software applications have been proposed in the past[Rand75][Hech76]. These models concentrated on the software hierarchy and handled only software faults. Hierarchical fault tolerance models for distributed systems to handle hardware failures have not been considered. The model we propose takes into consideration the hierarchical physical and control aspects of a distributed system.

Let the distributed system be modelled by a hierarchical network of interconnected sub–LAN's (see Figure 3). A sub–LAN (i.e., segment) at level $i$ in the hierarchy is directly under exactly one segment at level $i+1$ and each level of the hierarchy can have more than one segment.

The problem of providing fault tolerance in the entire distributed system is decomposed into:

1) Providing fault tolerance in each individual segment of the network

2) Integrating the fault tolerance schemes in the different segments of the network.

The fault tolerance scheme selected on a segment at level $i$ in the hierarchy should optimally support the fault tolerance requirements of the segment and serve as a higher level support for the fault–tolerance requirements of all segments directly below it (i.e., at level $i-1$). For example, consider a two level hierarchy with two segments, one on each level. Assume the segment at higher level has a distributed fault tolerance scheme and the segment at the lower level has a centralized fault tolerance scheme. The centralized fault manager is responsible for fault detection and recovery on any node in the lower segment. Faults that cannot be detected or recovered from are signaled to the segment at the level above. The distributed scheme at the higher level is not only responsible for fault detection and recovery of any node at that level but is also responsible for fault detection and recovery of the centralized fault manager at the lower level and any signaled faults from the lower level.

Several techniques have been proposed for supporting fault tolerance in distributed systems[AK83][BB83][PP83][Svob84][BJ85][LC86][KT87]. Most of these are very concerned with consistency and therefore have a high associated overhead. To select a specific technique for an individual segment in the network, two aspects need to considered:

1) The characteristics of the segment have to be evaluated with respect to the fault tolerance technique. Such characteristics include the communication traffic on the segment (e.g., bursty or continuous, high or low volume), the application constraints on the segment(e.g., time constrained or not), the channel behavior (e.g., high or low error rates), and the segment node features (e.g., homogeneous or heterogeneous nodes).

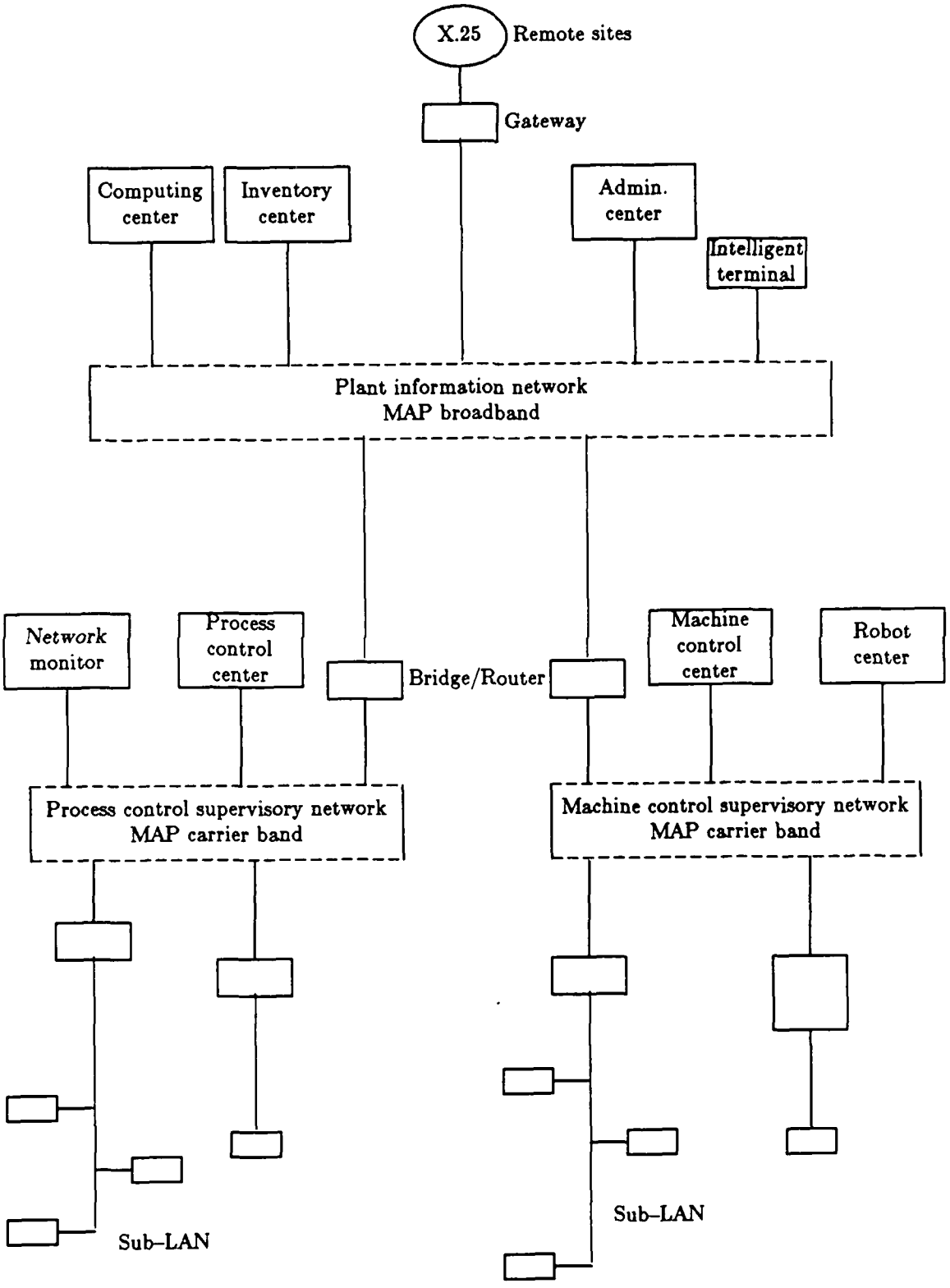



Figure 3: LAN Hierarchy in Manufacturing Systems

2) The fault tolerant invariants as dictated by the application domain, in each segment of the network, must be supported by the corresponding fault tolerance technique. Examples of invariants include the amount of resiliency that needs to be provided, the type of failures that need to be supported and the deadlines for recovery on failures.

Depending on the characteristics and invariance requirements, fault tolerance techniques can be chosen independently for each segment.

Integration of fault tolerance schemes at the different segments is necessary since all the segments are part of a single distributed system. There are two issues involved in this integration:

1) Message transfer between segments needs to be addressed since some fault tolerance techniques[PP83][BB83] record inter–process communication messages for recovery purposes. Message transfer is restricted in the model to be across a single level (i.e., messages from a node on a segment at level $i$ can be addressed to another node on the same segment, a node on the segment immediately above at level $i+1$ or a node on any of the segments immediately below at level $i-1$).

2) The control structure of the fault tolerance support for the entire distributed system should be hierarchical. This implies that the fault tolerance technique used in a segment at level $i$ is partly responsible for the fault tolerance support for all the segments immediately below it (at level $i-1$).

One possible approach to the problem of integration is illustrated by the ensuing example. Consider a hierarchical distributed system with two levels. Assume that the fault–tolerance is achieved by a centralized scheme at one level and a distributed scheme at the other level. In the centralized scheme, a single fault manager is responsible for fault detection and recovery. In the distributed scheme, all the nodes on the segment cooperate in the enabling fault detection and recovery. Four different cases can be identified:

1) The segment at level $i$ has a centralized fault tolerance scheme. The segment at level $i+1$ (i.e., immediately above in the hierarchy) also has a centralized fault tolerance scheme. The fault managers at both the segments are responsible for independently monitoring all communications between the two segments. This ensures recoverability of any process on the two segments. In addition, the centralized fault manager at level $i+1$ is responsible in the fault detection and recovery activities of the fault manager at level $i$, and any signaled failures from level $i$..

2) The segment at level $i$ has a centralized fault tolerance scheme. The segment at level $i+1$ has a distributed fault tolerance scheme. The fault manager at level $i$ is responsible for monitoring all communications between the two segments. This ensures recoverability of any process at level $i$. At level $i+1$, the node containing the fault manager of the segment at level $i$ is considered to be logically a node of the segment at level $i+1$. This

ensures recoverability of any process in the segment at level *i+1* and provides fault tolerance for the fault manager at level *i*.

3) The segment at level *i* has a distributed fault tolerance scheme. The segment at level *i+1* has a centralized fault tolerance scheme. This can be handled as in case 2.

4) The segment at level *i* has a distributed fault tolerance scheme. The segment at level *i+1* has a distributed fault tolerance scheme. One of the nodes on the segment at level *i* is selected to monitor all communication between the two segments. This ensures recoverability of any process in the segment at level *i*. A similar policy is adopted at level *i+1*. The failure of the monitoring node at level *i* (*i+1*) is supported by the distributed scheme at level *i* (*i+1*).

The advantage of the hierarchical model is that the types of the fault tolerant invariants provided at individual segments (i.e., amount of resiliency, type of failures supported) and the details of the techniques chosen at each segment (i.e., centralized or distributed control, consistency maintenance algorithms, recovery mechanisms, etc) can be chosen optimally depending on the requirements of the segment. The overhead involved is mainly in integrating the fault tolerant techniques at different levels.

### 4.2. Model Applicability to Manufacturing Systems

The hierarchical physical and control structure of manufacturing systems is well suited for the proposed fault–tolerance model. We use a manufacturing system composed of a network of interconnected LAN's to illustrate the applicability of the hierarchical fault tolerance model (see Figure 3). Each sub–LAN of the manufacturing system is referred to as a segment. The use of the hierarchical fault tolerance model decomposes the problem of providing fault tolerance in the entire network into the problems of providing fault tolerance support in individual segments of the network. Each segment is initially considered in isolation. Based on segment characteristics and requirements, schemes for fault tolerance support including recovery mechanisms and concurrency control strategies are chosen. The support schemes are finally integrated taking into account the hierarchical nature of the network.

Identification of the types of faults that need to be sustained in a distributed system is a prerequisite to designing a fault tolerance technique. There are three main types of faults that can be identified in manufacturing systems: software faults, hardware faults and communication faults. Software faults are caused by software failures. Techniques to handle such faults include recovery blocks[Rand75], exception handling[AL81][Cris82][Lisk82], and N–version programming[Aviz85]. Hardware faults are faults caused by failures of either computing nodes or automated devices in the system. Techniques to handle such faults include checkpointing[Bar81][BJR85], and multiple executions of identical processes[Coop85][Wens78][JT86][Jalo87]. Communication faults are caused by failures in the communication channels. One technique to handle such faults is by using specialized protocols supporting multiple channels[Jaco86] (channel redundancy). The hierarchical model for fault tolerance we have proposed can include any or all of these techniques to coexist in a manufacturing system. The model adds an additional control structure on top of the individual techniques for purposes of integration in the entire network. The need for different fault tolerance techniques and the use of the model in manufacturing systems is illustrated below.

The lowest level of the manufacturing system hierarchy is the device controller level. Device controllers are typically heterogeneous[Rous85]. Consistency requirements are very low. Thus, fault tolerance schemes based on checkpointing are not applicable. The communication traffic at this level is composed of short, frequent and time–constrained messages. The communication channel is prone to transient errors. The device programs are typically cyclic[Rous85], and they execute a small number of instructions each cycle. There are three options for providing fault tolerance under these circumstances:

1) Cluster the controllers at the device level so that the nodes within a segment are homogeneous. In this case, a centralized fault manager[PP83] can be used, subject to satisfying the real time constraints.

2) With heterogeneous nodes on a segment, employ hardware redundancy for fault tolerance support. Each process and its backup execute on tightly coupled processors. If a processor fails, recovery is instantaneous since the other processor can continue without any interruption. This is an expensive option, since the computational capability of the

backup processors is used only in the event of failures.

3) Since the processes are typically cyclic, with short cycle times, ignore errors occurring within a cycle and proceed to the succeeding iteration.

The cell control level is the second level of the manufacturing system hierarchy. The traffic intensity at this level is between that of the device level and the upper two levels of the hierarchy. Nodes on segments at this level are typically homogeneous. The application programs do not generally have real time constraints. Under these circumstances, each segment at this level can have a centralized fault manager. The communication overheads involved in fault detection and recovery are lower in centralized approaches as compared to the distributed methods.

The higher two levels of the manufacturing system (i.e., factory control level and division control level) have communication traffic patterns characterized by bursty and high volume data transfers. The nodes on these levels are typically homogeneous and the applications do not have real time constraints. Consistency maintenance is an important issue at these levels. With these characteristics, each segment at these levels is provided with distributed support[BB83][KT87]. A lazy update scheme for replicated data[JB86] and complex recovery mechanisms are justifiable at these levels.

Integrating the fault tolerance schemes at the different segments is approached as indicated in the model.

## 5. Design Issues

The hierarchical fault tolerance model is applicable in distributed systems that are characterized by a hierarchical physical and control structure. The main design issues are identification of system characteristics at the segment level, specification of the desired fault tolerant invariants at the segment level, evaluation of different fault tolerance techniques for a segment given its characteristics and desired invariants, selection of fault tolerance techniques at the segment level, and integration of fault tolerance techniques at the system level.

### 5.1. Fault tolerance at the segment level

At each segment of the network, one needs to identify the characteristics of the segment, the fault tolerant invariants needed at the segment and the fault tolerant mechanisms used at the segment. The segment characteristics and desired fault tolerant invariants are defined by the specific application domain of the distributed system. The fault tolerant mechanisms are defined by the specific fault tolerance technique chosen.

#### 5.1.1. Segment characteristics

The segment characteristics are used in evaluating the appropriateness of a particular fault tolerance technique is for a specific segment. Included in the list of segment characteristics are:

**Real–Time requirements:** Whether real time constraints are applicable or not in the segment.

**Node characteristics:** Whether nodes on the segment are homogeneous or heterogeneous.

**Consistency requirements:** Whether consistency on the segment is an important issue or not.

**Traffic patterns:** A measure of the communication workload on the segment.

**Fault characteristics:** The types of faults expected to be encountered on the segment.

**Redundancy:** The amount of redundancy allowable in the segment

#### 5.1.2. Invariance requirements:

The fault tolerant characteristics that need to be sustained within each segment constitute the invariants. For each segment in the network, the list of invariants that need to be identified include:

**Fault resiliency:** The number of faults to be sustained in the segment.

**Fault coverage:** The types of faults to be sustained in the segment.

**Local fault coverage:** The faults on a segment that can be detected and recovered within the same segment.

**Nonlocal fault coverage:** The faults on a segment that can be detected and/or recovered by signaling to another segment.

**External fault coverage:** The faults external to a segment that can be handled by the segment.

### 5.1.3. Fault tolerance techniques

The fault tolerance techniques are chosen based on the segment and invariant characteristics. Each fault tolerance technique specification should include the following features:

**Fault detection:** The mechanism for fault detection.

**Fault recovery:** The mechanism for fault recovery.

**Redundancy:** The redundancy employed in the technique to support fault tolerance.

**Consistency maintenance:** The mechanism used for maintaining consistency within the system.

Finally, the issue of integration of the different fault tolerance techniques selected at individual segments needs to be addressed.

## 5.2. Metrics

Different fault tolerance techniques can satisfy a given set of fault tolerant invariants. The need for metrics in evaluating fault tolerance techniques is obvious. To be useful, metrics should be quantifiable and measurable. Metrics and the evaluation of fault tolerance techniques should be included in the design of a fault tolerance system. We identify a set of features that merit consideration in the comparative evaluation of different fault tolerance techniques.

All fault tolerance techniques use some form of hardware and/or software redundancy. In evaluating a fault tolerance technique, we distinguish between the *facilities* provided by the technique and the *costs* associated with providing those facilities.

Included in the list of facilities are:

**Fault Resiliency:** A system with fault tolerance can continue to perform activities even with the occurrence of system failures. Fault resiliency refers to the number of failures that a fault tolerance mechanism can support, and still allow the system to continue functioning.

**Fault Coverage:** The type of faults tolerated in the system is an useful measure of the facilities provided by the fault tolerance mechanism. Hardware faults, software faults, resource and load dependent faults, and communication faults are examples of common types of faults in a system.

**Fault Transparency:** The degree to which the fault tolerance technique is transparent

to a user of the system is a measure of the ease with which it can be used. The possibilities include explicit invocation of fault tolerance mechanisms by a user, support via individual node operating systems and complete transparency.

The overhead costs associated with fault tolerance support can be classified into the following categories:

**Duplicate Resources:** Duplicate hardware resources used explicitly for the purpose of fault tolerance result in low resource utilization, though fault recovery is immediate. Duplicate hardware resources used for computation and fault tolerance are more efficient in terms of resource utilization, but system operation in the event of failures is suboptimal.

**Communication overheads:** Overheads due to communication are inevitable in any fault tolerance scheme. The communication overheads can be categorized into the overheads due to fault tolerance support (e.g., checkpointing), the overheads due to fault detection and recovery, and the overheads due to consistency maintenance.

**Time overheads:** Overheads involving a loss of time are directly related to the three categories of communication overheads. In addition, time is lost in process recomputation after failure and recovery.

Another perspective in the evaluation of a fault tolerance technique is the amount of *health data* that the technique provides. Heath data is defined as failure related data in a system collected and provided by the fault tolerance mechanism, to a system manager, useful in increasing the reliability of the system. The hierarchical model fits well with the notion of improving fault tolerance dynamically depending on the fault characteristics of a system. Each level in the hierarchy collects information about the failures it encounters and summarizes the failure data from the lower levels. The data available at the top levels of the hierarchy can be used to improve the reliability of the entire system. The hierarchical organization helps in that the information available at different levels impose a logical structure on the failure data.

## 6. Conclusion

We have proposed a hierarchical model for supporting fault tolerance in distributed systems. The model is applicable to distributed systems that have a hierarchical physical and control structure. Manufacturing systems are shown to have such a hierarchical structure. The

model is found to be useful in providing fault tolerance support for manufacturing systems. The main advantage of the model is that it allows different fault tolerance and recovery techniques to be employed at different segments of the network. To help in the choice of specific techniques in particular network segments, a set of metrics to evaluate fault tolerance techniques have been presented. Preliminary ideas about integrating different fault tolerance techniques have been presented.

Three main problems have been identified in supporting fault tolerance in hierarchically structured distributed systems. The first involves the analysis of the characteristics of the processes, nodes and communication within each segment of the hierarchy, as related to the the support of fault tolerance. The analysis should include the desired fault tolerant invariants associated with the segment. The second involves the selection of fault tolerance techniques optimally suited to the identified segment characteristics and capable of preserving the desired invariants. The notion of metrics to evaluate fault tolerance schemes is introduced. Finally, the integration of different selected fault tolerance techniques within a single distributed system needs to be addressed.

This work can be extended in two directions. Specific application domains like manufacturing systems, that can be modelled hierarchically, need to be analyzed at the segment level and appropriate fault tolerance techniques need to be proposed. Precise, quantifiable metrics for comparing fault tolerance techniques need to be formulated. Secondly, the problem of integrating different fault tolerance techniques within a single distributed system, independent of the application domain, needs to be investigated in detail.

### References

[AC86] T. Albert and R. Charles, " A Proposed Hierarchical Control Model for Automated Manufacturing Systems", in *Journal of Manufacturing Systems*, vol. 5, no. 1, pp. 15–25, 1986.

[AL81] Anderson, T. and Lee, P. A. *Fault Tolerance, Principles and Practice*, Prentice–Hall International, Englewood Cliffs NJ, 1981.

[AK83] T. Anderson and J. Knight, " A Framework for Software Fault Tolerance in Real–Time

Systems", in *IEEE Trans. Software Eng.*, vol. SE–9, pp. 355–364, May 1983.

[Aviz85] A. Avizienis "The N–Version approach to fault–tolerant software", IEEE Trans. on Software Engg., vol SE–11, no. 12, pp. 1491–1502, Dec. 1985.

[BAR81] Bartlett, J. F., "A NonStop kernel", *Proc. of 7th ACM Symp. on Operating Systems Principles*, pp. 22–29, 1981.

[BB83] A. Borg and J. Baumbach, " A Message System Supporting Fault Tolerance," in *The Ninth ACM Symposium on O.S. Principles; O.S. Review*, vol. 17, no. 5, pp. 90–99, Oct. 1983.

[BJ85] K. Birman and T. Joseph, "Implementing Fault–Tolerant Distributed Objects," in *IEEE Trans. Software Eng.* , vol. SE–11, pp. 502–508, June 1985.

[BJR85] Birman, K. P., T. A. Joseph, T. Raeuchle and A. E. Abbadi, "Implementing fault–tolerant distributed objects", *IEEE Transactions on Software Engineering*, June 85, vol. SE–11, no. 6, pp. 502–508.

[Coop85] E. C. Cooper, "Replicated Distributed Programs", *Proc. of the 10th ACM Symp. on Op. Sys. Principles, Op. Sys. Review*, vol. 19, no. 5, pp. 63–80, Dec. 85.

[Cris82] F. Cristian, "Exception Handling and Software Fault Tolerance", *IEEE Transactions on Computers*, vol. C–31, no. 6, pp. 531–540, Jun. 1982

[GM85] General Motors: MAP Specification version 2.1, Mar. 1985.

[GM86] General Motors: MAP Specification version 2.1.A and 2.2, Aug. 1986.

[TOP87] TOP Specification version 3.0, 1987.

[Hech76] H. Hecht, "Fault–Tolerant software for real–time application," in *Comput. Surveys*, vol. 8, no. 4, pp. 391–407, Dec. 1976.

[IEEEa] IEEE Standard 802.4–1985, " Token Passing Bus Access Method and Physical Layer Specification," 1985.

[IEEEb] IEEE Standard 802.3–1985, " CSMA/CD Access Method and Physical Layer Specification", 1985.

[ISO85] The International Organization for Standardization, Data Processing, "Open Systems Interconnection Basic Reference Model", *ISO Doc. DIS7498*, 1985.

[Jaco86] D. W. Jacobson, " High Performance Reliable Token Bus for MAP Network Architecture", in *Proc.of XI Conf. on Local Computer Networks*, Minnesota, pp. 26–33, Oct. 1986.

[Jalo87] P. Jalote, "Resilient objects in broadcast networks", *IEEE Transactions on Software Engineering* (to appear, accepted 8/87).

[JB86] T. Joseph and K. Birman, " Low Cost Management of Replicated Data in Fault–Tolerant Distributed Systems", in *ACM Trans. on Computer Systems*, vol. 4, no. 1, pp. 54–70, Feb. 1986.

[JT86] P. Jalote and S. K. Tripathi, "Fault–tolerant computation in synchronous message passing systems", Technical Report, University of Maryland, Department of Computer Science.

[Lisk82] Liskov, B. H. *On Linguistic Support for Distributed Programs, IEEE Transactions on Software Engineering*, vol. SE–8, no. 3, pp. 203–210, May 1982.

[LC86] A. Liestman and R. Campbell, " A Fault–tolerant Scheduling Problem", in *IEEE Trans. Software Eng.*, vol. SE–12, pp. 1089–1095, Nov. 1986.

[KT87] R. Koo and S. Toueg, " Checkpointing and Rollback–Recovery for Distributed Systems", in *IEEE Trans. Software Eng.*, vol. SE–13, pp. 23–31, Jan. 1987.

[Rous85] N. E. Rouse, "Developing Network Standards", in *Machine Design*, pp. 69–73, Dec. 1985.

[PP83] M. Powell and D. Presotto, " Publishing: A Reliable Broadcast Communication Mechanism", in *The Ninth ACM Symposium on O.S. principles; O.S. Review*, vol. 17, no. 5, pp. 100–109, Oct. 1983.

[Rand75] B. Randell, "System structure for software fault tolerance," in *IEEE Trans. Software Eng.*, vol. SE–1, pp. 226–232, Mar. 1975.

[Svob84] L. Svobodova, "Resilient Distributed Computing," in *IEEE Trans. Software Eng.*, vol. SE–10, pp. 257–267, May 1984.

[TS84] Tamir Y. and S'equin C., " Error Recovery in Multicomputers Using Global Checkpoints," in *IEEE Symposium on Parallel Processing*, pp32–41, 1984.

[Wens78] J. Wensley et. al., "SIFT: design and analysis of a fault–tolerant computer for aircraft control", *Proc. IEEE*, vol. 60, pp 1240–1245, Oct. 1978.

# REPORT DOCUMENTATION PAGE

| | |
|---|---|
| 1a REPORT SECURITY CLASSIFICATION<br>UNCLASSIFIED | 1b RESTRICTIVE MARKINGS<br>N/A |
| 2a SECURITY CLASSIFICATION AUTHORITY<br>N/A | 3 DISTRIBUTION / AVAILABILITY OF REPORT<br>approved for public release; distribution unlimited |
| 2b DECLASSIFICATION / DOWNGRADING SCHEDULE<br>N/A | |
| 4 PERFORMING ORGANIZATION REPORT NUMBER(S)<br>UMIACS-TR-87-53<br>CS-TR-1939 | 5 MONITORING ORGANIZATION REPORT NUMBER(S) |
| 6a NAME OF PERFORMING ORGANIZATION<br>University of Maryland<br>6b OFFICE SYMBOL (*If applicable*)<br>N/A | 7a NAME OF MONITORING ORGANIZATION<br>Office of Naval Research |
| 6c ADDRESS (*City, State, and ZIP Code*)<br>Department of Computer Science<br>University of Maryland<br>College Park, MD 20742 | 7b ADDRESS (*City, State, and ZIP Code*)<br>800 North Quincy Street<br>Arlington, VA 22217-5000 |
| 8a NAME OF FUNDING / SPONSORING ORGANIZATION<br>8b OFFICE SYMBOL (*If applicable*) | 9 PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER<br>N00014-87-K-0463 |
| 8c ADDRESS (*City, State, and ZIP Code*) | 10 SOURCE OF FUNDING NUMBERS<br>PROGRAM ELEMENT NO / PROJECT NO / TASK NO. / WORK UNIT ACCESSION NO |

11 TITLE (*Include Security Classification*)
On Fault Tolerance in Manufacturing Systems

12 PERSONAL AUTHOR(S)
Prasad R Chintamaneni, Pankaj Jalote, Yuan-Bao Shieh, and Satish K Tripathi

| 13a TYPE OF REPORT | 13b TIME COVERED | 14 DATE OF REPORT (*Year, Month, Day*) | 15 PAGE COUNT |
|---|---|---|---|
| Technical | FROM _____ TO _____ | October 1987 | 21 |

16 SUPPLEMENTARY NOTATION

| 17 COSATI CODES | | | 18 SUBJECT TERMS (*Continue on reverse if necessary and identify by block number*) |
|---|---|---|---|
| FIELD | GROUP | SUB-GROUP | |
| | | | |
| | | | |

19 ABSTRACT (*Continue on reverse if necessary and identify by block number*)

An important issue in manufacturing systems involved in factory automation is the support for fault tolerance. This paper describes the hierarchical, physical and control structure of manufacturing systems and proposes a hierarchical model for fault tolerance support. The usefulness of the hierarchical fault tolerance model is shown in the manufacturing system domain and the main issues involved in the general applicability of the model are discussed.

| | |
|---|---|
| 20 DISTRIBUTION / AVAILABILITY OF ABSTRACT<br>☐ UNCLASSIFIED/UNLIMITED ☐ SAME AS RPT ☐ DTIC USERS | 21 ABSTRACT SECURITY CLASSIFICATION<br>UNCLASSIFIED |
| 22a NAME OF RESPONSIBLE INDIVIDUAL | 22b TELEPHONE (*Include Area Code*) / 22c OFFICE SYMBOL |

END

DATE

FILMED

MARCH

1988

DTIC