Information-Integration of Manufacturing Systems Using Object-Oriented Technology

Gintas Jazbutis, Ching-Yang Wang, Bipin Chadha
Graduate Research Assistants

Dr. Robert E. Fulton
Professor

Material Handling Research Center
and the
School of Mechanical Engineering

Georgia Institute of Technology
Atlanta, GA 30332
USA

ABSTRACT

This paper describes an object-oriented approach to building an information system for optical fiber manufacturing. This object-oriented approach takes the following steps: defining the information and systems requirements, building the models of the manufacturing system, and implementing a prototype information system. This paper gives the details of the work done to date in the first two steps and compares the process with a previous effort in building a relational-based information system.

INTRODUCTION

A manufacturing information system plays a leading role in the manufacturing industry. Information is now a corporate asset, and hence the timeliness, accuracy, and integrity of this information is very important. A generic information framework of optical fiber products has been done [1,2]. The functions studied include: order processing and shipping, product tracking, quality control and testing, scheduling and inventory control. The implementation of this information system using the relational data model approach has been done in a phased implementation approach in collaboration with AT & T [3]. The approach includes: understanding the layout of the optical fiber manufacturing processes, building the functional model (IDEF0), constructing the data flow model (DFD), building the extended entity-relationship model (EER), designing the relational database system (RDBMS), designing user-interactive interface window screens, prototyping, testing, and delivering to the production floor.

A major issue identified in the information system implementation at AT & T was the need to manage exceptional events. An exceptional event is one in which a normal, successful operation is interrupted (as by, for example, a machine failure). Upon such an occurrence, the process is diverted to a new procedure, with attendant information requirements. Object-oriented technology has the ability to encapsulate and deal with the data *and* the method to manipulate that data within an object; in this case, to encapsulate and deal with the data of exceptional events and the methods of handling exceptional events. Thus it is believed that object-oriented technology is suitable for managing exceptions; there are also other functionalities object-oriented technology offers: user-defined data types, encapsulation, inheritance, explicit relationships, and data integrity [9]. These features can be used to improve the semantic expression, support of heterogeneous data types, representation of complex objects, the defining of a dynamic schema, and incorporation of exceptional events [9]. The integration of the information system using an object-oriented methodology is underway.

The methodology chosen is based upon the Object Modeling Technique (OMT) [4]. This method consists of analysis, system design, and implementation. Analysis consists of modeling the system in different ways; namely, object modeling, dynamic modeling, and functional modeling. System design determines the approach to implementation, and implementation is the building and execution of the design of the system. This paper describes efforts done in the analysis phase. Details of the prototype implementation are described in [5].

The approach taken to performing a prototype implementation consists of a number of steps. The first is to identify a manageable piece of the problem--the part chosen at AT & T is the optical fiber product manufacturing line. It is then necessary to identify the user's and customer's information needs. Then a study of the manufacturing process is required (generally, people know some of what goes on in their facility, but not all of what goes on). It is important to understand and improve the process before developing an information system.

The next phase is analysis: modeling the system. As the case study is being done on a manufacturing facility, it is important that the modeling methods reflect that environment. Hence, IDEF0 was chosen to do the functional modeling, and IDEF2 was chosen to do the dynamic modeling (the IDEF methodologies being tailored to the manufacturing environment). A variation of EER is used to do the object modeling. With the functional and dynamic models, the manufacturing process has been described, as well as the data flows within that process. With the object model, the relationships between the objects have been described. The methods must next be incorporated in the objects; the method is how each object manipulates its data.

The implementation phase consists of building a shared object database, developing the appropriate application interfaces (using SMALLTALK), and testing and evaluating the system.

A comparison is made between the object oriented methodology developed here and the relational methodology previously used. Advantages of the object-oriented approach are discussed.

GENERIC SOFTWARE FRAMEWORK

A conceptual sketch of the initial integrated system configuration is given in figure 1. The user interacts with an executive, which controls the application programs within the system. This will ensure that all the application programs within the manufacturing system will be highly consistent with each other by having the same user interface. Flexibility and modularity are achieved by interconnecting the applications loosely using a database manager. Note that the information framework can encompass tasks involved in the design of manufacturing systems (including material handling systems) as well as tasks involved in the operation of manufacturing systems. While the specifics of these two activities may be different, the integration frameworks are quite similar. A shared information system framework should provide:

A Shared Data Base: to eliminate redundancies and inconsistencies.

A Data Base Management System: to store, arrange, and retrieve data from the shared data base. The DBMS should provide a mechanism for data independence and integrity checks. An Object-Oriented DBMS is being used for this project.

Interfacing Programs: to act as pre- or post-processors between the application programs and the shared data base. Use of Abstract Data Types (ADT) with prespecified operations should help standardize and simplify these interfaces.

A Data Dictionary: to describe the data and their functions in the system.

An Executive: to manage the integrated manufacturing system. Besides providing a good user interface, an executive should manage multiple users and multiple tasks.

A Geometric Modeler: to provide geometric representation facilities. Geometric representation forms an integral part of most engineering applications and hence deserves special treatment.

A Knowledge Based System: to capture and use knowledge in a structured manner. The knowledge base would interact heavily with the data base and use the information stored in the DBMS to draw inferences.

An Information Management Capability: to perform various information management functions, such as constraint management, exception management, change management, etc.. Based on experiences in case studies and the Generic Integration Model (GIM), our original framework was found deficient in certain respects. In addition to the above components the framework should provide a set of managers for the interacting applications to use as tools. The utilities that these managers provide could be part of the DBMS. In case the DBMS doesn't provide these utilities, these have to be provided externally.

BACKGROUND OF OBJECT-ORIENTED MODELING METHODOLOGY

Object-oriented modeling methodology is loosely addressed in the issues of object-oriented analysis (OOA) and object-oriented design (OOD). The main issues can be 7ategorized in the following items: (a) to describe and define the statement of requirements among which are: a set of objects, object attributes, object relationships, and object responsibilities in a specific problem domain; (b) to systematically provide modeling tools which serve as guidelines to help the designer translate unambiguously into an object-oriented programming language and carry out the implementation.

The pioneer object-oriented modeling methodologies are addressed in the works of Shlaer and Mellor, [19]; Coad and Yourdon, [15]; McIntyre and Higgins, [17]; Rumbaugh et al., [4]; Booch, [6]; Wirfs-Brock et al., [7], etc.. Among these works three modeling methodologies including the Object Modeling Technique (OMT) methodology [4]; the Booch methodology [6]; and the Classes Responsibilities Collaborations (CRC) methodology [7] support systematic tools/notations for the designer to conceptually model the information in a problem domain.

The OMT methodology [4] consists of three models: the object model, the dynamic model, and the functional model. An object model is a static structure which describes a set of objects, object attributes, and object relationships that can exist in a system. A dynamic model describes the concepts which deal with flow of control, interactions, and sequencing of operations in a system of concurrently-active objects. A functional model consists of multiple data flow diagrams which show the flow of values from external inputs, through operations and internal data stores, to external outputs. According to our experience derived from an earlier case study using relational approach [1,8,9], we observe that the OMT methodology is likely evolved from the concepts of database design. Comparing it to our modeling methodology using relational approach, we can broadly treat the object model as the extended extended entity-relationship model (EER) with the addition of object-oriented programming concepts and treat the functional model as the data flow model (DFD) and process model (IDEF0) with the addition of object-oriented programming concepts. (In our earlier modeling methodology, we did not build the dynamic model.)

The Booch methodology [6] includes: class diagrams, state transition diagrams, object diagrams, timing diagrams, module diagrams, and process diagrams. It is difficult to find similarities between the modeling methodology in Booch and in our earlier relational approach. Generally speaking, the Booch methodology is likely to have evolved from the design concepts of software development.

The CRC methodology [7] is a responsibility-driven methodology. In the CRC methodology, the designer first establishes who is responsible for each action that is to be performed. That is, first identify a class and its responsibilities to perform actions. Following the scenario progress from the known to the unknown, the designer can list another class as a collaborator based on the degree of connection or cooperation with respect to a specific responsibility in a known class. In this way, the designer can build the representation of a class and its several pairs of responsibility and collaboration class. A 4 inch by 6 inch card, first introduced by [10], is adopted to record a class and its responsibilities and collaboration classes. The size of a CRC card could be used as a measure of appropriate complexity. A class that is expected to perform more tasks than can fit easily in the CRC card space is probably too complex, and it may require either moving some responsibilities elsewhere or dividing the tasks between two or more separate classes. The CRC methodology can help the designer to carry out the detail implementation. Thus, the CRC methodology is likely to have evolved from the design concepts of object-oriented programming languages.

A hybrid object-oriented modeling methodology taking the OMT methodology [4] as a kernel part and adding the process model (IDEF0) [11,12] and CRC methodology [7] has been implemented in [5] as one solution to model optical fiber manufacturing. In this hybrid modeling methodology, two object-oriented concepts are emphasized: inheritance and encapsulation. Inheritance can highlight the semantic meaning in the steps of conceptually modeling manufacturing processes. During the modeling steps, there could exist either an is-a relationship or a part-of (or has-a) relationship. The is-a relationship defines superclass-subclass hierarchies, whereas the part-of relationship describes data to be maintained within a class. In OMT object model notations there are two different symbols to represent the is-a relationship by generalization concepts and the part-of relationship by aggregation concepts. A class cannot make direct modifications to other classes, because it violates the encapsulation concepts of object-oriented technology. The CRC responsibility-driven approach provides a guideline during the definition of object. According to our experience [1,8,9], the IDEF0 methodology can provide a good understanding of the manufacturing processes. Details of this implementation can be found in [5].

NEED FOR A MANUFACTURING-ORIENTED METHODOLOGY

Manufacturing systems have their unique requirements regarding processes and material flow. It was realized that earlier approaches utilized either process flow or data flow as their drivers, but manufacturing systems require an understanding of both process and data flows. Our approach utilizes both modeling methods to drive the system development.

Manufacturing environments are complex and involve numerous processes and functions. Models are therefore needed to logically break these environments down to manageable pieces. The environment can be seen from various viewpoints in the model. The viewpoint can be from a process standpoint, information flow standpoint, or material flow standpoint. All these views are needed for the designer to completely understand the manufacturing system.

A facilities layout is very important for understanding and modeling a manufacturing system. It is also important to model the disposition of resources because of the dynamic and complex nature of resource allocation and deallocation typical of a manufacturing system. It is necessary to understand how the material flows through the system, how the products are processed, what resources are utilized, and also how information flows through the system while manufacturing activities are taking place. It usually requires more than one modeling method to capture this information. This is a significant step as it not only helps in identifying what is needed for the information system, but it also helps clarify and document the manufacturing processes for everyone to use and critique. This can help eliminate redundant and ad-hoc codes, standards, and practices that may have developed over time. It is important to note that the process must be well understood and improved before attempting to automate or develop an extensive information system.

Success of the information system depends on how little it affects the ongoing production and organizational structure of the shop floor, because manufacturing is an ongoing process and it is not cost-effective to stop production over periods of time to install a new system. Hence, as a first step, the information system must support the ongoing manufacturing process. Once the system is running and supports the manufacturing activities, then other relevant information management activities should be addressed. Thus, a categorization of the information for phased implementation is very useful. The following phases/categories are proposed to ensure smooth transition: Product/Process information, Quality information, Process/Product improvement information, and Exceptional Events information.

OBJECT MODELING APPROACH

The approach taken in this paper is based on the Object Modeling Technique (OMT), an approach described by James Rumbaugh, et al [4]. It consists of three phases: Analysis, Design, and Implementation. The first phase, Analysis, focuses on understanding the particular system under study. The user's and customer's goals and requirements are determined; these form the framework and direction for the system development and implementation. A major portion of analysis is in modeling the system in different ways. A *functional model* is developed to help in understanding all of what goes on in the manufacturing process; people do not know all of what goes on in the manufacturing process, but only part, generally in their own sphere of work. A *data flow model* is developed to identify and describe the data and information flows in the manufacturing process; an *object model* is developed, in which objects and their relationships to other objects are determined; and a *dynamic model* is developed to understand and describe the time-dependent elements of the system.

The final results of this phase are an understanding and description of 1) what is required of the system to be developed, and 2) the manufacturing process, especially from an information point of view. This paper describes efforts in this analysis phase.

The Design and Implementation phases are where the system is designed, having already developed the requirements for the information system and an understanding of the manufacturing system, and then implemented. The objects are designed--the attributes and methods are specified--and the database is built. The system is also designed--the hardware requirements are determined (e.g., quantity and location of terminals on the shop floor, etc.), the dynamics of the objects are implemented, and the user interface processes are designed and developed. The final step of the implementation is testing the system, to ensure that it operates as properly and meets the requirements of the users. It is here that many unforeseen bugs in the system arise and can be dealt with.

The methodology developed for the system development can be summarized in the following steps:

1. Identification of the requirements and information needs of the manufacturing system.

2. Development of functional models for the product using two of the process modeling techniques (IDEF0 and Data Flow Diagrams).

3. Development of an object model for the product using a hybrid of Extended Entity Relationship diagramming.

4. Development of a dynamic model using IDEF2.

5. Implementation of a prototype information system to evaluate and demonstrate the capabilities needed in a system. This would involve database design and its implementation on a Database Management System, design of user/application interfaces, etc.

THE CASE STUDY: INFORMATION SYSTEM DESIGN FOR OPTICAL FIBER PRODUCTS MANUFACTURING

Fiber Optics technology is evolving rapidly with new LAN, space, and military applications. An optical fiber can transmit all forms of communications (voice, image, data) by digital light signals. Light signals are generated by tiny laser devices that pulse millions of times per second. The messages are carried over hair-thin glass fibers. A cable less than 0.6 inches in diameter containing 216 fibers suitable for placement in aerial, buried, underground, or building environments could transmit and receive more than 2.5 million telephone conversations simultaneously. This technology needs advanced manufacturing systems supported by integrated information systems. A joint project was started with the AT&T's Atlanta Works plant which is part of the Network Cable Systems division of AT&T, and the case study focuses specifically on activities at that plant. The Georgia Tech team has been working very closely with the AT&T team to design a prototype information system for optical fiber products manufacturing, which is serving as a case study for this project.

The optical fiber manufacturing process as AT & T is shown in Figure 2. A hollow glass tube is heated and collapsed to a solid rod. This rod is then drawn out to form the optical fiber and is wound on a reel. The reels of optical fiber are then either sold as a product of used in making a number of other products, such as optical fiber cables (bundles of individual insulated optical fibers).

The project was started with the following objectives: understand the information needs of the manufacturing system, develop process and data flow models of a representative part of the manufacturing system, develop an information model for the selected subsystem, design and implement the database and application/user interfaces, and provide specifications/requirements for the future system.

ANALYSIS: REQUIREMENTS DEFINITION

The first step in the analysis of the system is to define the requirements. These generally take the form of what the users need the system to do. Users include: the customer, management, accounting, procurement, order handling departments (order receiving and shipping), engineering, manufacturing, quality control, testing, and information systems developers. The projected system needs to provide such following information on demand: product data, inventory status, order status, efficiency reports, etc.. The sum of these determine information requirements; system requirements, which will define the system configuration, include such criteria as flexibility, speed, specific hardware, user interfaces, maintainability, etc.. Once the requirements are defined, the system development proceeds, the goal being to meet these requirements.

ANALYSIS: INFORMATION NEEDS

Some of the information needs identified for the integrated optical fiber system are:

* Optical fiber products information.
* Product and material tracking information
    - Real time status.
    - Automation of unique product identification.

Bar code identification.
Data collection.
Data storage.
Data maintenance.
Long term data archiving.
Data retrieval.

* Inventory management information
  - In-process inventory accounting/control.
  - Finished product inventory accounting/control.
* Information from support functions
  - Order processing.
  - Shipping.
  - Scheduling.
  - Routing.
  - Testing.
* Quality related information
  - Defects.
  - Scrap.
* Exception management information
  - Machine Start up.
  - Machine shut down.
  - Fiber breakages during manufacturing.
  - Machine breakdowns.
  - Operator injuries.

As the list suggests, these information needs are largely generic except for the product specific data noted first. The manufacturing process itself is interesting in the sense that it has characteristics of both flow and discrete manufacturing processes. Within a manufacturing line the product behaves as a flow process, but in between lines it is a discrete entity in the form of reels of fiber and cable. Several fibers can be “assembled” to form a cable, or the product can be cut into pieces at any of several workstations. It may undergo testing and repair at any point in time. Thus, optical fiber manufacturing provides and excellent case study for a large class of problems.

In general the information to be captured can be classified into the following categories:

Product/Process Information.
Quality Information.
Process Improvement Information.
Security Information.
Exceptional Events Information.

This is also the approximate order in which the information system should be implemented, i.e., initial implementation should focus on the information that is necessary to support the ongoing manufacturing process. Success of the information system depends on how little it affects the ongoing production and organizational structure of the shop floor. Once the system is running and supports the manufacturing activities, then other relevant information management activities should be addressed.

Figure 3 shows the framework that has been developed for this case study. The framework is consistent with the Generic Integration Model [1].

ANALYSIS: SYSTEM REQUIREMENTS

The most important factors considered in developing the basic system configuration are the following:

Flexibility. Since manufacturing is a very diverse function, involving various equipment from design through operation, there are a number of software packages designed to do specific tasks. The envisaged system should be flexible enough to accommodate the diversity in the software packages.

Modularity. The system should be modular in nature, to accomplish the addition and/or deletion of software packages with minimum effort.

Consistency. Various applications within the system should be consistent with one another. This will help the users to learn the system faster and to use it more efficiently. One way to accomplish this is to have the same front end interface for all the programs.

Reusability. Parts of or the whole of the system should be reusable for different applications. This becomes important due to shorter product life cycles.

Maintainability. The system should be easy to maintain. Since these systems and their requirements are continuously changing, their maintenance becomes an important issue.

Multitasking. The system should support multiple users as well as allow multiple tasks to be carried out simultaneously.

ANALYSIS: MODELING

Modeling the Manufacturing Process

After establishing the information needs, the manufacturing process was studied (see figure 2). At this point the information of interest was how different products go through their manufacturing steps. The motivation behind this is to help clarify the manufacturing process to those people who need to know it in order to design the information system. In general, everyone has a good idea of how particular areas (especially their own) work, but no one really knows how the whole system works. The modeling exercise not only helps clarify the current process but also help in identifying places where the process can be improved. The modeling tool IDEF0 [11,12] was chosen to model the activities taking place in the plant. An IDEF0 diagram hierarchically represent the activity flow. An IDEF0 diagram was continuously decomposed to finer levels of detail, until the team was satisfied that the process was fully modeled. The information was gathered by interviewing engineers, shop floor supervisors and operators. It was noticed (as expected) that each person knew to a good level of accuracy the function that he was dealing with. Operators knew very well the detailed operations, while engineers knew better about the higher level activities. It took several iterations to assimilate the knowledge into the model. The information captured was the flow of activities, the events that controlled these activities, and the resources utilized by those activities. Since IDEF0 does not capture the flow of information between several functions, data flow diagrams (DFD) were developed simultaneously to capture the flow of information. A set of IDEF0 diagrams are shown in figures 4, 5, and 6.

The process should be carefully examined for inefficiencies before proceeding further. The control system should be decoupled from the information system. IDEF0 provides a convenient way of differentiating information requirements from control requirements.

Modeling the Data Flows

Data Flow Diagrams (DFD) were utilized to capture the specific information needs of the processes and activities of the system. A set of DFDs [13] for the system are shown in figures 7 and 8. The DFDs and IDEF0s were compared frequently to find any discrepancies in the models. The DFDs helped define a set of attributes of information of interest. This created a data dictionary for the system. A set of codes and standard definitions were developed for use in the system. These codes were adapted from the codes being used on the shop floor. Over a period of time redundant and often inconsistent codes become prevalent. Different departments use different names to identify the same items. Many redundant data and definitions were identified and eliminated by this exercise. Some of these standards developed were: product type codes, operational area codes, defect codes, disposition codes, transaction codes, etc.. It is very important to standardize these definitions early in the information system design process to avoid costly and lengthy delays in application development. Sample data dictionary entries are shown in figure 9. A Computer-Aided Software Engineering (CASE) tool was utilized for the models which helped considerably in clarifying the data model and reducing the complexity of the task.

Modeling the Object Relationships

Process models and DFDs yield a set of objects that are of interest. These objects are logically related to each other. An EER-type object diagram was used to model the relationships between these objects. Hierarchical structures of objects as well as relationships between objects are described. A list of objects identified in the study is given in figure 10; a representative object is shown in figure 11; and an object relationship diagram for the study is shown in figure 12.

Objects models describe the static data structure of objects, attributes, and responsibilities and their relationships to one another. A manufacturing object (Insulated-Fiber-Manufacturer) represents an abstraction with crisp boundaries and meaning for a manufacturing application. A part-of relationship between this manufacturing object (Insulated-Fiber-Manufacturer) and its component objects (Machine etc.) shows the linkages among these objects. All Objects have identity and are distinguishable. Thus, object relationships are tightly coupled and lead to enhancing the semantic meaning. An object class (Optical-Fiber) describes a group of objects with common attributes, operations, and semantics. The is-a relationship allows designer to organize classes into a hierarchical structure based on their similarities and differences.

Modeling the Object Dynamics

The methodology used to capture the time-dependent aspects of the manufacturing system is IDEF2 [14]. This methodology was designed for the manufacturing environment, thus making it suitable for use in this analysis. The IDEF2 model actually consists of four submodels: the facility, the entity-flow, the control, and the resource disposition submodels. The facility submodel is essentially a plant layout diagram, identifying all the resources within the system and their arrangement. It also can show the general flow of entities through the system. The entity-flow submodel describes the flow of entities through the manufacturing process, describing the sequence and timing of events each entity encounters, and the required resources in performing the events. The control model is a network that describes activities or conditions that affect system status (such as machine breakdowns) and flow of entities through the system, but do not in themselves cause entity flow (generally, initiating entity, resource, and facility changes). The resource disposition submodel is a tree diagram describing the allocation of a particular resource upon completion of an activity (it may become idle, or be allocated to another task, etc.). Examples of IDEF2 diagrams for this study are given in figures 13, 14, 15, and 16.

Upon completion of the analysis phase, the identified objects are designed--the data attributes and methods are built; the database is designed and built; and user/process interfaces are designed and developed, all using SMALLTALK [5]. The system is then tested and evaluated. This work has been done and is described in detail in another paper.

COMPARISON OF RELATIONAL AND OBJECT-ORIENTED METHODS

The case study under consideration has a working product tracking system with a relational database management system (RDBMS) [3]. The analysis phase of the methodology to develop the object-oriented system for optical fiber manufacturing is similar in many ways to the one used for the RDBMS-based system. Both initially define the requirements and information needs (in fact, they are basically the same for both methodologies); both develop functional models (IDEF0 and DFD) to understand to manufacturing system and information flows in that system; both develop a data dictionary; both develop a data model to represent the relationships. The relational methodology identifies data entities, and uses an Extended Entity-Relationship (EER) diagram to model the relationships between data entities. The object-oriented methodology identifies objects, and uses a modification of EER to model relationships between objects. The relationships between entities are different than the relationships between objects; the object model incorporates IS-A and PART-OF relationships, and may show a hierarchical structure. The object-oriented methodology differs significantly from the relational in that a dynamic model is built for the O-O method. The dynamic model is important in determining the methods for each object, and shows the interactions of the objects and their various data.

While the analysis phase of the object-oriented methodology as discussed in this paper is similar to that of the relational-based, the design and implementation phases differ significantly; these are discussed in [5].

CONCLUSIONS

This paper has described an object-oriented approach to information-integration of manufacturing systems; in particular, it describes the steps in the analysis phase of system development. An information system is being developed to support the manufacturing processes in a optical fiber production facility. The system is being developed and implemented in a phased manner, according to the framework presented herein. The system and the methodology used to develop the system has been described in detail. The requirements of the users of the system were the driving force behind the system, specifying the information they require, and the form in which they require it. We have developed functional, object, and dynamic models of the manufacturing facility to help in understanding the processes and information flows, to identify the relevant objects in the system and their relationships to each other, and to help in identifying the methods of each object. We have compared the object-oriented methodology with a relational-based methodology used earlier: the analysis phase differs in that the object-oriented methodology also develops a dynamic model of the manufacturing system to identify the object-data interactions and methods. Significant differences between the two methdologies occur in the design and implementation phases [5].

We have concluded that there is a need for an information modeling methodology suitable for manufacturing environments. The methodology should include, for example, facility layout information, resource allocation and deallocation information, entity and data flows, resource and material use descriptions, exceptional event occurences, etc. A case study approach has been adopted to understand information-integration in manufacturing environments. Object-oriented technology offers many functionalities that relational-based technology does not in information-integration of manufacturing systems: user-defined data types, encapsulation, inheritance, explicit relationships, and data integrigy. These features can be used to improve the semantic expression, support of heterogeneous data types, representation of complex objects, and defining of a dynamic schema [5]. Exceptional events can also be incorporated into the information system. These make the object-oriented approach more suitable than the relational-based approach for manufacturing information systems.

The work described in this paper sets up the design and implementation of the object-oriented system, which is described in [5].

ACKNOWLEDGEMENTS

The authors wish to thank the Material Handling Research Center at Georgia Tech for providing funding for this research activity. Acknowledgements are due to Dr. Ron Bohlander at Georgia Tech as well as R.F. Foyle, J.C. Calhoun, A.J. Hart, and R. Lopez at AT & T for their suggestions and help in various stages of this project.

REFERENCES

1. Chadha, B, and Fulton, R. E., "Issues Involving Information-Integration of Material Handling", ASME Winter Annual Meeting, Dallas, Texas, Nov. 25-30, 1990.

2. Fulton, R. E., 1991, "Issues Involving Information-Integration of Design and Manufacturing", International Conference on CAD of Machinery 91, Beijing, China, Sep. 16-20.

3. Chadha, B., Fulton, R.E. and Calhoun, J. C., "Case Study Approach for Information-Integration of Material Handling", The 1991 ASME International Computers in Engineering Conference and Exposition, Santa Clara, California, Aug. 18-22, 1991.

4. Rumbaugh, J., Blaha, M., Premerla, W., Eddy, F., and Lorensen, W.,Object-Oriented Modeling and Design, Prentice Hall Inc., 1991.

5. Wang, C.Y., Chadha, B., Jazbutis, G., and Fulton, R.E., "Object-Oriented Approach to Modeling Optical Fiber Manufacturing", International Conference on Object-Oriented Manufacturing Systems, Calgary, Canada, May 4-6, 1992.

6. Booch, G., Object-Oriented Design with Applications, Benjamin/Cumming Publishing Company, 1991.

7. Wirfs-Brock R., Wilkerson B., and Wiener L., Designing Object-Oriented Software, Prentice-Hall Inc., 1990.

8. Chadha, B., Jazbutis, G., and Wang, C.Y., "Software Framework for Integrated Material Handling in a Plant", Material Handling Research Center, Georgia Tech, Spring and Fall Monitor Meetings, May and November, 1990.

9. Chadha, B., Jazbutis, G., Wang, C.Y. and Fulton, R.E., "An Appraisal of Modeling Tools and Methodologies for Integrated Manufacturing Information System", The 1991 ASME International Computers in Engineering Conference and Exposition, Santa Clara, California, Aug. 18-22, 1991.

10. Beck, K. and Cunningham, W., "A Laboratory for Teaching Object-Oriented Thinking", Proceeding of 1989 OOPSLA, Sigplan Notices, 24(10):1-6, 1989.

11. Bravoco, R.R. and Yadav, S.B., "Requirement Definition Architecture --- An Overview", Computers in Industry, Vol. 6, pp 237-251, 1985.

12. Bravoco, R.R. and Yadav, S.B., " A Methodology to Model the Functional Structure of an Organization", Computers in Industry, Vol. 6, pp 345-361, 1985.

13. Gane, C., Rapid System Development Using Structured Techniques and Relational Technologies, Prentice-Hall Inc., 1989.

14. Bravoco, R.R. and Yadav, S.B., " A Methodology to Model the Dynamic Structure of an Organization", Information Systems, Vol. 10, No.3, pp 299-317, 1985.

15. Coad, P. and Yourdon, E., Object-Oriented Analysis, Yourdon Press, 1990.

16. Fulton, R.E., and Yeh, C.P., "Managing Engineering Design Information", AIAA/AHS/ASEE Aircraft Design, System and Operations Conference, 1988.

17. McIntyre, S.C. and Higgins, L.F., "Object-Oriented Systems Analysis and Design : Methodology and Application", Journal of Management Information Systems, Summer 1988, Vol. 5, No. 1, pp 25-35, 1988.

18. Objectworks\Smalltalk user's guide, ParcPlace Systems Inc., 1990.

19. Shlaer, S. and Mellor, S.J., Object-Oriented Systems Analysis : Modeling the World in Data, Yourdon Press, 1988.

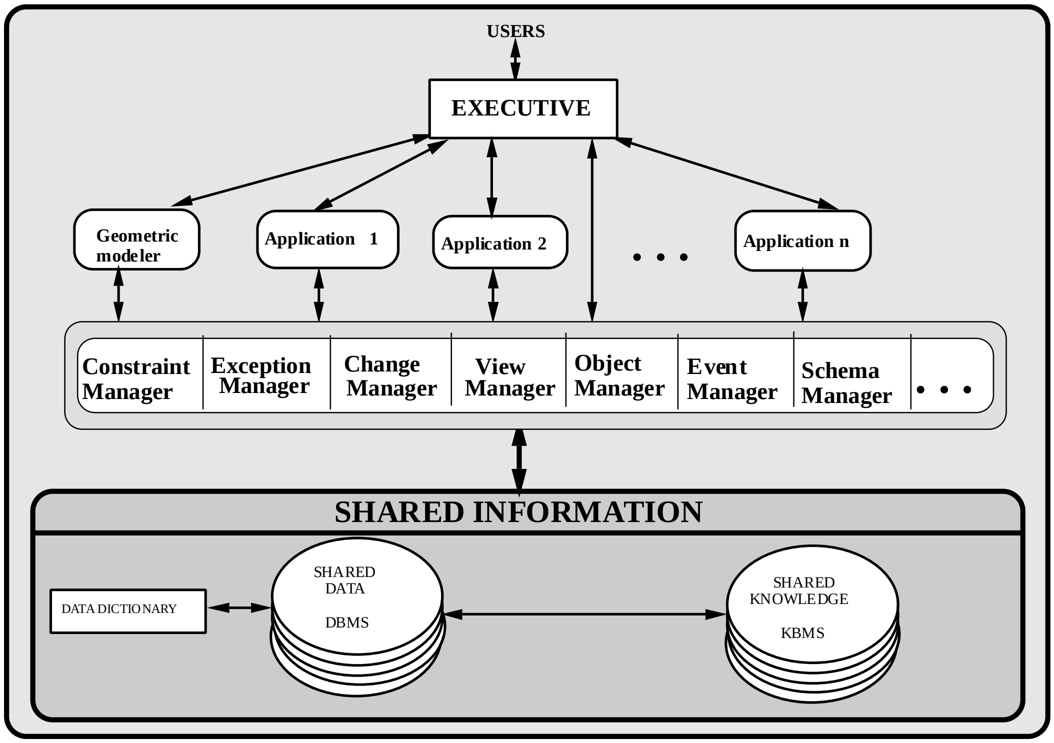


FIGURE 1. PROPOSED FRAMEWORK FOR INTEGRATION

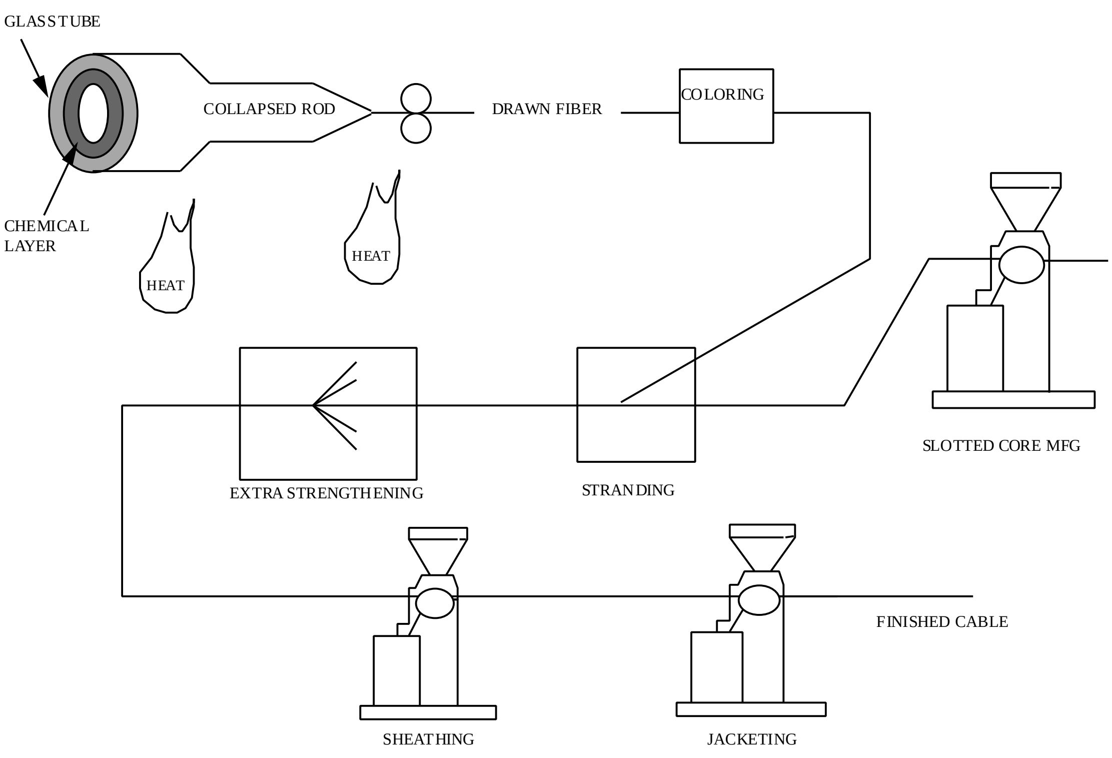


FIGURE 2. MANUFACTURING PROCESS FOR OPTICAL FIBER PRODUCTS

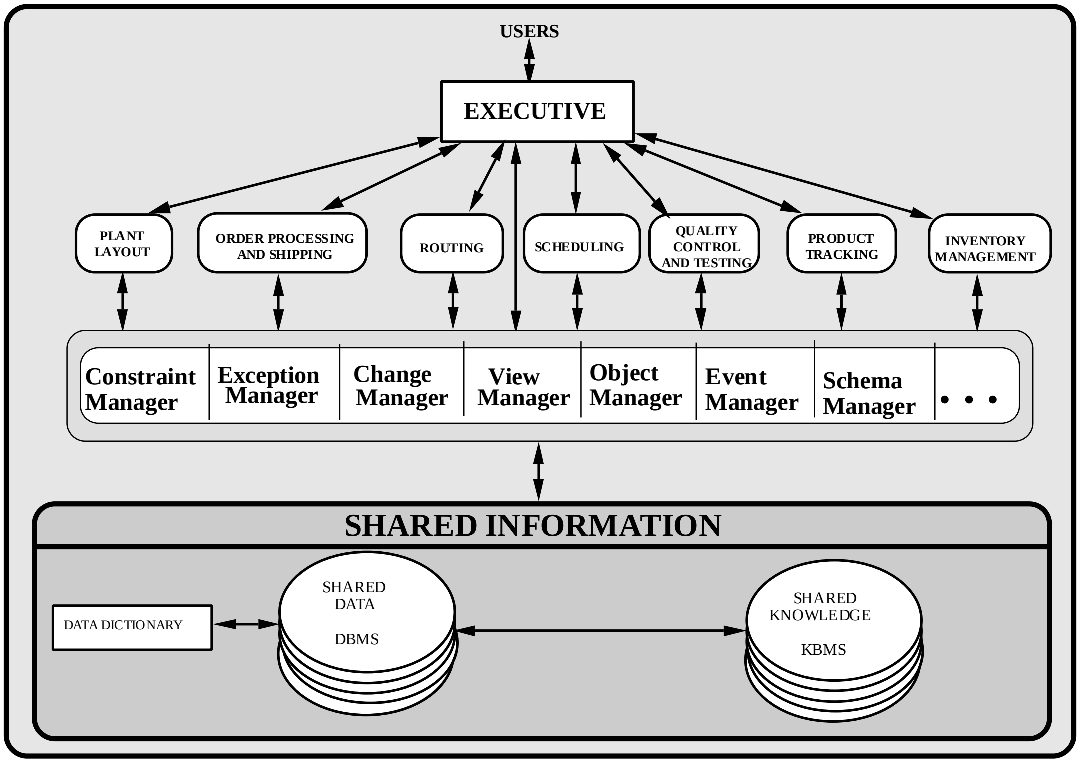


FIGURE 3. PROPOSED FRAMEWORK FOR OPTICAL FIBER MANUFACTURING

USED AT:	AUTHOR: G. B. JAZBUTIS PROJECT: AT&T NOTES: 1 2 3 4 5 6 7 8 9 10	DATE: 2/8/91 REV:	x WORKING DRAFT RECOMMENDED PUBLICATION	READER DATE	CONTEXT:

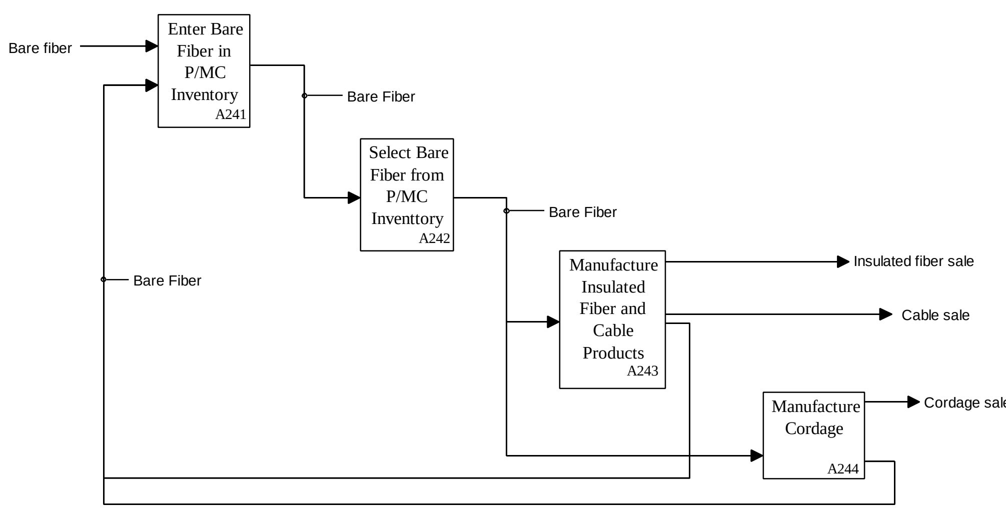
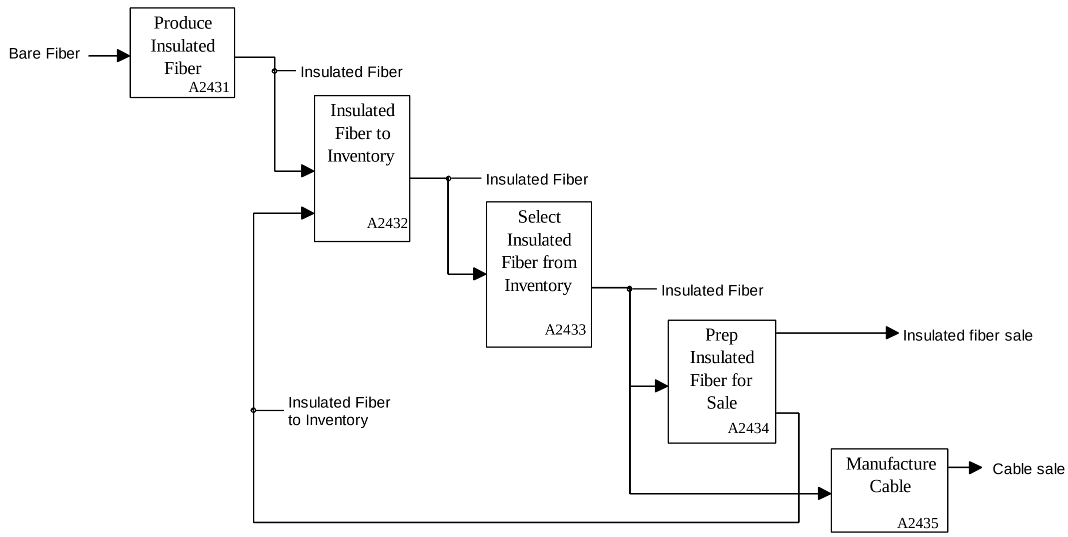


FIGURE 4. IDEF0 DIAGRAM

FIXED

NODE: A24	TITLE: Manufacture Premise/Mil Cable Products	NUMBER:

USED AT:	AUTHOR: G. B. JAZBUTIS PROJECT: AT&T NOTES: 1 2 3 4 5 6 7 8 9 10	DATE: 2/8/91 REV:	x	WORKING	READER DATE	CONTEXT:
				DRAFT		
				RECOMMENDED		
				PUBLICATION		

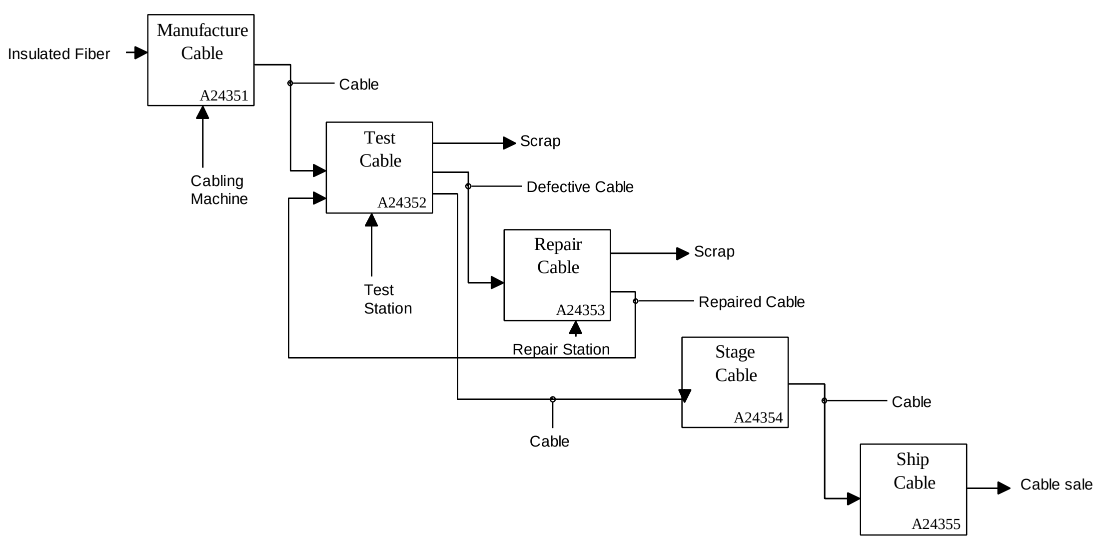


FIXED

FIGURE 5. IDEF0 DIAGRAM

NODE: A243	TITLE: Manufacture Insulated Fiber and Cable Products	NUMBER:

USED AT:	AUTHOR: G. B. JAZBUTIS PROJECT: AT&T NOTES: 1 2 3 4 5 6 7 8 9 10	DATE: 2/8/91 REV:	x	WORKING	READER DATE	CONTEXT:
				DRAFT		
				RECOMMENDED		
				PUBLICATION		



FIXED

FIGURE 6. IDEF0 DIAGRAM

NODE: A2435	TITLE: Manufacture Cable	NUMBER:

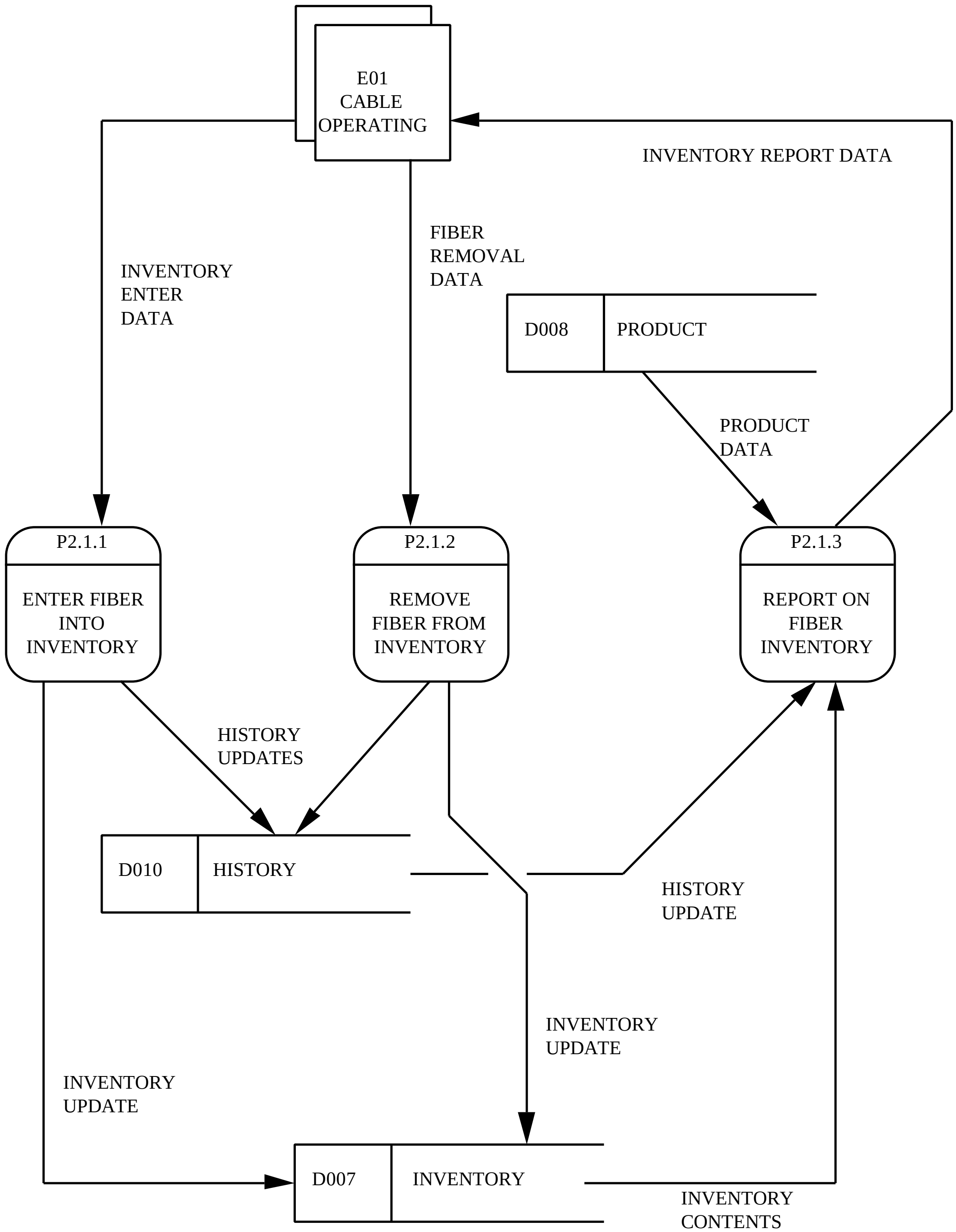


FIGURE 7. FIBER INVENTORY DATA FLOW

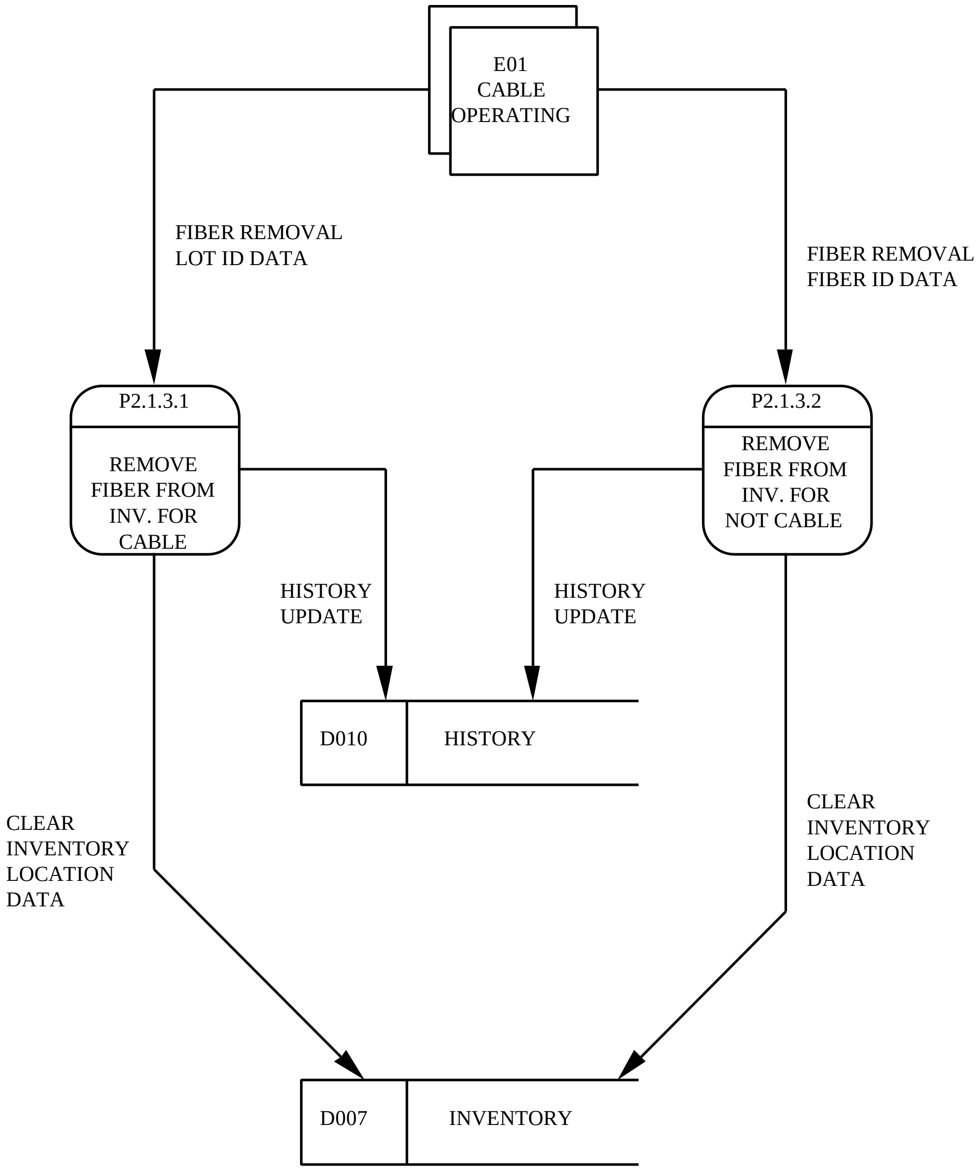


FIGURE 8. REMOVE FIBER FROM INVENTORY

NAME	ALTERNATE NAME	DESCRIPTION
PROD_ID	PRODUCT_IDENTIFIER	AN IDENTIFIER TO UNIQUELY IDENTIFY AN OPTICAL FIBER PRODUCT SPOOL
OPER_ID	OPERATOR_IDENTIFIER	AN IDENTIFIER TO UNIQUELY IDENTIFY AN OPERATOR
MACH_ID	MACHINE_IDENTIFIER	AN IDENTIFIER TO UNIQUELY IDENTIFY A MACHINE
DEF_CODE	DEFECT_CODE	IDENTIFIES THE TYPE OF DEFECT IN A FIBER
ATT_PRF	ATTENUATION_ PERFORMANCE	ATTENUATION PERFORMANCE OF FIBER AT 1300 mn IN dB/km.
CRUSH_STR	CRUSH_STRENGTH	CRUSH STRENGTH OF FIBER IN kN.
ORD_NUM	ORDER_NUMBER	ORDER NUMBER ASSIGNED TO EACH NEW ORDER

FIGURE 9. SAMPLE OF DATA DICTIONARY

* **OPTICAL FIBER**

  FIBER
  INSULATED FIBER
  REINFORCED FIBER
  CABLE
  DOUBLE FIBER
  ETC.

* **ORDER**
* **ITEM**
* **INVENTORY**
* **MACHINE**
* **OPERATOR**
* **DEFECT**
* **SCRAP**
* **TRANSACTION HISTORY**

FIGURE 10. IDENTIFIED OBJECTS

OpticalFiber

barCodeNo
productType
productStatus
length
parent
child
productDefect

show_barCodeNo
show_productType
show_productStatus
show_length
show_parent
show_child
show_productDefect
change_length
create_scrap
split_fiber
add_productDefect

FIGURE 11. SAMPLE OBJECT: OPTICAL FIBER

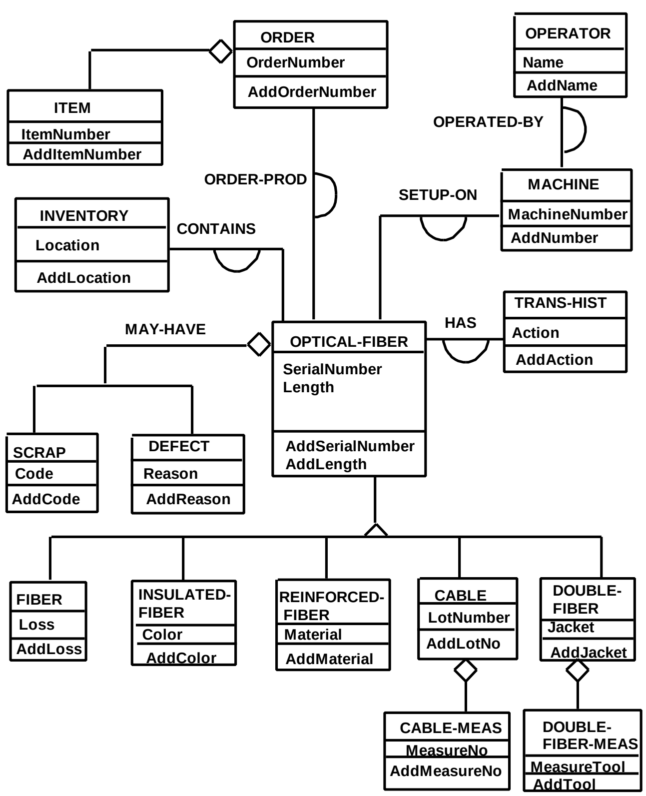


FIGURE 12. OBJECT MODEL FOR OPTICAL FIBER MANUFACTURING

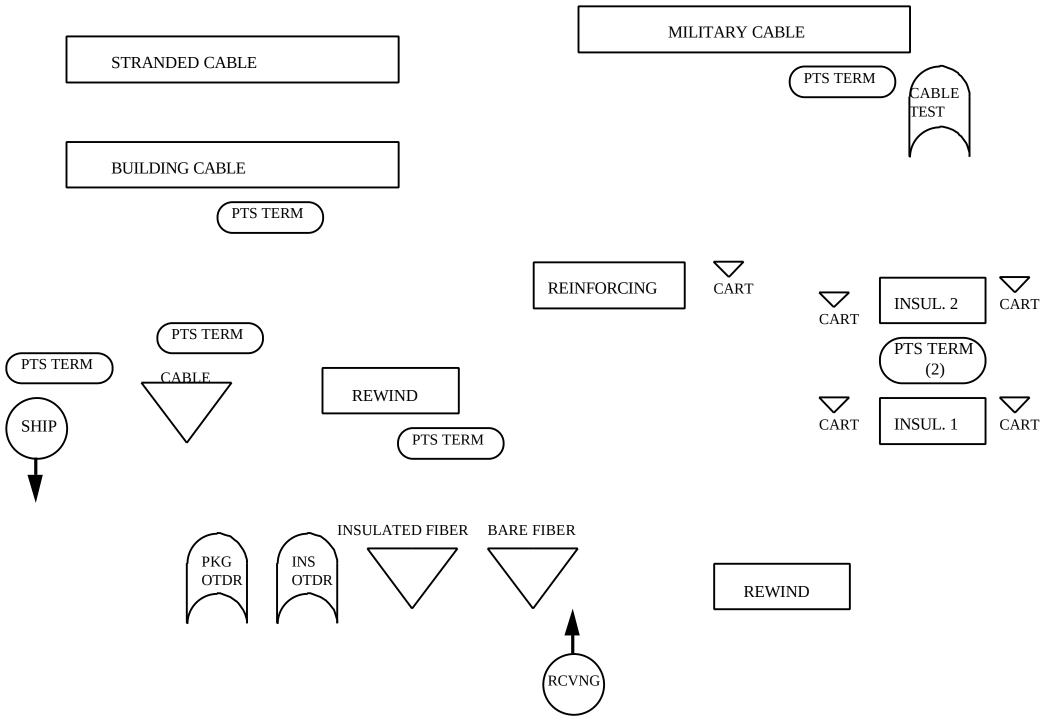


FIGURE 13. IDEF2 DYNAMIC MODEL FACILITY LAYOUT

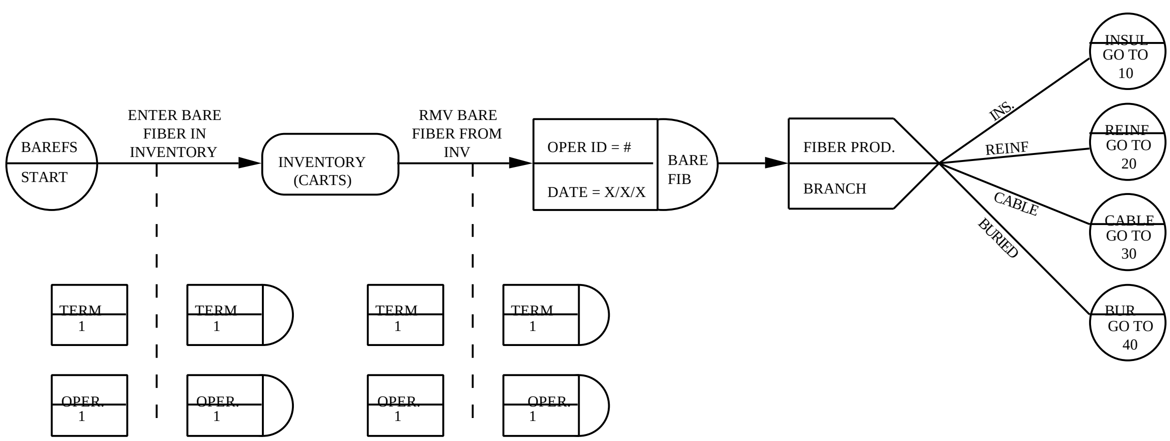


FIGURE 14. IDEF2 DYNAMIC MODEL ENTITY FLOW SUBMODEL

(FIBER TO/FROM INVENTORY)

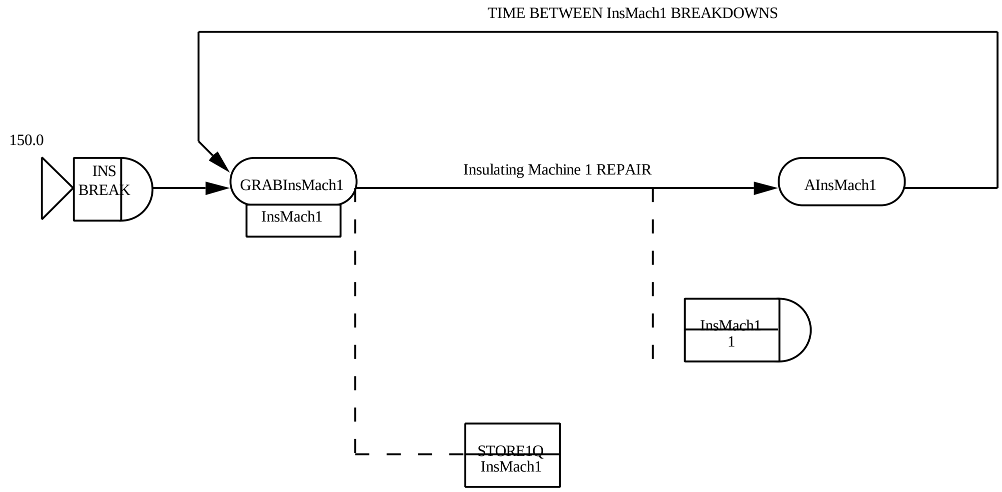


FIGURE 15. IDEF2 DYNAMIC MODEL CONTROL SUBMODEL

(INSULATING MACHINE BREAKDOWNS)

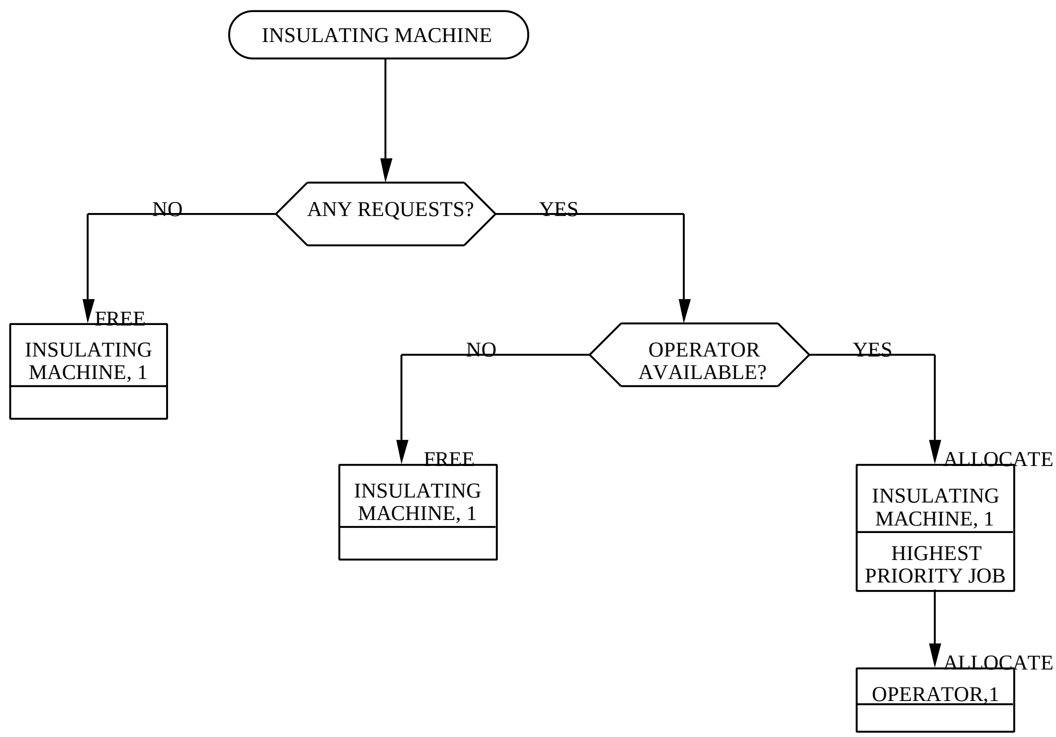


FIGURE 16. IDEF2 DYNAMIC MODEL RESOURCE DISPOSITION SUBMODEL