A. Kusiak, *Engineering Design: Products, Processes, and Systems*, Academic Press, San Diego, CA, 1999.

CHAPTER 1

FUNDAMENTALS OF PROCESS MODELING

1. INTRODUCTION

A model may represent a system (e.g., a manufacturing system), an object (e.g., designed artifact) or a problem (e.g., designing a shaft) and is typically constructed for the purpose of analysis. Models are used to describe existing systems, as well as to evaluate the feasibility and anticipated performance of proposed systems. Although models must capture enough details to facilitate reliable experimentation, the purpose of modeling must not be violated. Such a violation may be caused by including unnecessary information at a cost exceeding the cost of building and/or experimenting with the actual system. The motivation and potential drawbacks of modeling efforts vary considerably between applications and methods. A thorough understanding of functions, data, resources, and the organizational structure is essential in modeling processes. A model can provide a sufficient understanding of the system being modeled without disturbing the actual environment. For example, in manufacturing, models can be used to analyze the manufacturing system ability to respond to the market changes. This enables rapid and accurate reconfiguration when new products are demanded. Ultimately, an executable version of the model can simulate and even control the actual process. Once a process model has been developed, one should be able to perform various analyses, e.g., time (temporal), quality of the underlying processes by presenting a user with different perspectives (see Figure 1). The time perspective optimizes the process duration and distribution of cycles among activities and it determines critical activities. It also enables the validation of process models. The other perspectives consider quality, reliability, risks, costs. A negotiation perspective (module) evaluates the conflicts from decisions made by the individual perspectives.

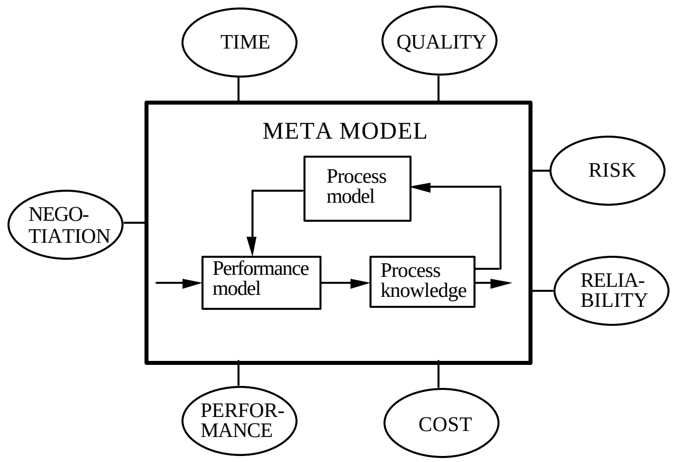

Figure 1. Process analysis tool

The core of the process analysis tool is a meta model including three major components:

- generalized process model, e.g., in the form of an extended object-oriented IDEF3 model
- performance evaluation model for generation of performance metrics and evaluation of the process performance
- knowledge about the process

2. EXISTING MODELING METHODOLOGIES

The development of tools for modeling and analysis of processes has been motivated by the need to increase productivity and improve communications and structure of design and manufacturing systems. Constructing a process model is only one component of a comprehensive process modeling effort. Representing models as process graphs, performing observational and structure model analyses are important issues.

This section briefly describes several of the existing process modeling methodologies. They vary in scope, appearance, and theoretical foundations.

CIM-OSA

Computer Integrated Manufacturing - Open Systems Architecture (CIM-OSA) is under development by the ESPRIT Consortium AMICE (European Committee for Standardization 1994). The methodology facilitates total enterprise modeling through a model construction process that includes enterprise requirement definitions, enterprise design specifications, and an enterprise implementation description. Four enterprise views (perspectives) are considered: function, information, resource, and organization. Within each view, generic building blocks describe the functions, information, and resources in the system. Relations between building blocks define the total enterprise (Beekman, 1989).

CIM-OSA recognizes the functional, information, resource, and organizational perspectives often considered in modeling processes. Furthermore, abstraction concepts such as encapsulation, classification, and inheritance are supported.

EXPRESS

In 1980, the technical committee TC184/SC4/WG5 of the International Standard Organization (ISO) initiated work on EXPRESS and Version 1.0 was approved in 1991. The

PDES consortium uses EXPRESS systematically in its work on STEP. A graphical representation of the language is available as EXPRESS-G. The methodology provides a syntax for defining classes of entities (which may be information, resources, material, products, etc.) that support abstraction. However, dynamic behavior can not be modeled.

Although not as explicit as CIM-OSA, EXPRESS can capture the functional, information, resource, and organizational perspectives of a process. The abstraction concepts of EXPRESS are enhanced by the concepts of Subtype and Supertype (European Committee for Standardization, 1994).

GRAI Method

The GRAI method was developed in the early 1980s at the GRAI Laboratory, University of Bordeaux, France (Doumeingts et al., 1987). It is built around a conceptual reference model that is based on the theory of complex system, hierarchical systems, system organization, and discrete event theory. The manufacturing system is structured in three subsystems: a physical system, an information system, and a decision system. The GRAI formalism focuses on the decision subsystem and relies on other methods, such as IDEF0 (discussed later in this Section) and Entity Relationship Attribute, to model the physical and information systems. The GRAI formalism is supported by two graphical representations: the GRAI grid, and the GRAI net.

The GRAI method explicitly focuses on decomposition from the organizational perspective. However, the method does not cover the functional, information, and resource perspectives. Through decomposition, the method supports encapsulation but classification and inheritance are not supported.

IEM

Integrated Enterprise Modeling (IEM) is a public domain methodology developed by IPK Berlin (European Committee for Standardization, 1994). Unlike the previous methods, IEM is designed around the object-oriented paradigm. Objects are categorized as products, orders, and resources. A generic activity model is defined for operating on objects.

The object-oriented paradigm allows for the simultaneous modeling of the functional and information perspectives through a single construct class. Although not explicitly considered in IEM, the organizational perspective can be added and integrated using the class concept. This methodology demonstrates the robust and generic modeling capabilities provided by the object-oriented paradigm that are considered essential in process modeling.

PSL/PSA

Problem Statement Language/Problem Statement Analyzer (PSL/PSA) was commercially developed by META Systems. The PSL component is a language that can be used to describe information systems in terms of objects, properties, and relationships. PSL/PSA is based on the concepts of relational database theory. Formal and graphical representations are provided and reports can be generated from the commercially available software.

SSADM

Structured Systems Analysis and Design Method (SSADM) is a method of systems analysis and design. The Central Computer and Telecommunications Agency, United Kingdom, developed it in the early 1980s. The method focuses on analysis of business requirements, design, and specification of application databases and software. A project is broken down into modules that contain activities that must be completed to deliver the product. Each step has a list of tasks, inputs, and outputs. SSADM includes modules for feasibility study, requirements analysis, requirements specification, logical system specification, and physical design (Ashworth, 1988).

The clear focus of SSADM is on the information perspective. Although many information modeling concepts have been incorporated (i.e., data flow modeling, entity event modeling, and relational data analysis), the method does not employ the object-oriented paradigm.

OOMIS

The Object-Oriented modeling methodology for Manufacturing Information Systems (OOMIS) consists of two phases, an analysis phase, and a design phase (Kim et al., 1993). The first task of the analysis phase is to decompose the manufacturing functions into component functions using an approach similar to IDEF0, which is discussed later in this chapter. After a functional model has been constructed, function tables, data tables, and operation tables are generated. In the design phase, the object-oriented paradigm is used to translate the function tables, data tables, and operation tables into an integrated information model. Classes consisting of an identifier, attributes, and methods are defined for components of the manufacturing system. Two specific class types are used, function class and entity class. Relationship diagrams facilitate semantic design.

Unlike other methods that treat the functional and information perspectives independently (e.g., IDEF – discussed later in this Section), the object-oriented paradigm is employed to form an integrated model. In fact, the information perspective is derived directly from the functional model.

MOSYS

Development of MOSYS (Modeling SYStem) dates to 1993 (Mertins et al., 1993). MOSYS is a software tool for modeling of the functional structure, topology, and control rules of enterprises. The functions of system can be described with five building blocks: manufacture, transport, store, assemble, and test. These blocks are parametric and they can be customized to a specific application. MOSYS includes modules for cost evaluation, system animation, as well as a Petri net module for material flow.

Petri Nets

A Petri net (PN) is a directed graph that is defined as a five-tuple (P, T, IN, OUT, M_0), where $P = \{p_1, p_2,..., p_m\}$ is a finite set of places, $T = \{t_1, t_2,..., t_n\}$ is a finite set of transitions, the set of places and the set of transitions are disjoint. The mapping from places to transitions is defined by IN, whereas OUT defines the mapping from transitions to places (Peterson 1981). In a graphical representation, circles, transitions, represent places by bars or boxes, and there are arcs either from a place to a transition or from a transition to a place (see Figure 2). If there exists an arc from place p to transition t, the place p is called the input place of the transition t. Similarly, the output place of the transition is symbolized by an arc from the transition t to the place p. The input (output) place often denotes the precondition (postcondition) of an event, whereas the occurring event is associated with a transition. A marking in a Petri net is a vector M that specifies the assignment of tokens to the places, i.e., $M : P \rightarrow N$, $N = 0$. An initial state of a Petri net is called the initial marking, M_0. The execution of a PN is controlled by the number and distribution of tokens in the net. A PN executes by firing transitions. A transition is enabled if each of its input places contains at least as many tokens as arcs from a place to a transition. A transition may fire if it is enabled. When a transition fires, tokens are removed from its input places and deposited to its output places. Firing a transition will in general change the marking M of the PN to a new marking, M'.

Consider a process that consists of only one activity. To execute the activity, two resources have to be used. The net in Figure 2(a) models this process, where $P = \{p_1, p_2, p_3\}$, $T = \{t_1, t_2\}$, $M_0 = (1\ 1\ 0)$. Transition t_1 in Figure 2(b) is enabled because each of its input places, p_1 and p_2, contains a token. When a transition is enabled, it can be fired. After firing the transition t_1, the output place of t_1, p_3, gets a token, the new marking M' is (0 0 1).

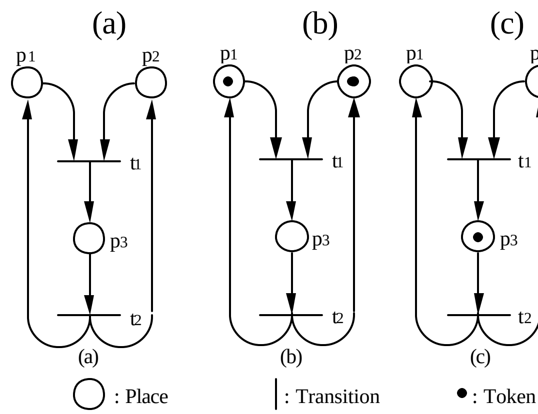

Figure 2. An activity that uses two resources: (a) PN model, (b) initial marking, (c) marking after t_1 has been fired

IDEF Methods

Development of the Integrated DEFinition (IDEF) methods began with the Air Force Program for Integrated Computer-Aided Manufacturing (ICAM). Through this work, the need for a family of mutually supportive methods for enterprise integration was realized and development was continued in the Air Force Information Integration for Concurrent Engineering (IICE) program. As the IDEF methods have become widely used in concurrent engineering (CE) efforts, total quality management (TQM), and business process reengineering (BPR) initiatives. The IDEF acronym represents a family of Integrated DEFinition methods.

IDEF0 was developed for modeling a wide variety of systems, which use hardware, software, and people to perform activities. An IDEF0 model consists of three components, diagrams, text, and a glossary, all cross-referenced to each other. The box and arrow diagrams are the major components of the model. In a diagram, a box represents a function and an arrow represents an interface. A box is assigned an active verb phrase to represent the function. An interface may be input, an output, a control, or a mechanism, and is assigned a descriptive noun phrase. At a given level, there are three and six function boxes. Each box on the diagram may be decomposed into a lower level boxes. The resulting diagram represents information stored in a hierarchical form.

The IDEF3 process flow description is made up of activities (units of behavior), links, and junction boxes. Relationships between activities are modeled with three types of links, precedence links, relational links, and object flow links.

The IDEF1x model is used to semantically model the relationships between various pieces of data. The basic constructs of an IDEF1x model include entities, attributes, and relationships. The IDEF4 methodology provides syntax and semantics for capturing the thought process that is required to develop modular, maintainable, and reusable applications programmed in object-oriented languages such as C++, Object Pascal, Common Lisp Object System (CLOS), and Smalltalk. The dynamic behavior of a system can be captured using IDEF2. Methods for modeling domain ontologies (IDEF-5) and defining the motives that drive the decision making process (IDEF6) have also been developed. As a whole, the IDEF family of methods facilitates model construction for a variety of purposes.

3. INTEGRATED DEGINITION METHODOLOGY

The origin of structured approaches to modeling and analysis of information systems dates to the late 1960s and early 1970s. A graphical notation for representing the processes, which transform data, was needed to assist in the development of design architecture. DeMarco (1979) introduced the term *structured analysis*, as well as a set of symbols and a methodology for creating information flow models. The *data flow diagram* (DFD) models information flow and transformation on various levels of detail. At level 0, the DFD represents the entire system. Additional information may be incorporated on level 1, level 2, and so on, as subfunctions of the overall system.

As deficiencies in the DFD approach became apparent, Page-Jones (1980) proposed variations and extensions of structured analysis, Gane and Sarson (1982), Ward and Mellor (1985), Hatley and Pirbhai (1987), and many others. The need for modeling real-time systems by capturing control flow and control processing information became apparent. The research effort of Ross and Schoman (1977) has resulted in the Structured Analysis and Design Technique (SADT). SADT is a graphic language which "provides a limited set of primitive constructs from which analysts and designers can compose orderly structures of any required size" (Ross and Schoman, 1977). The notation consists of boxes and arrows, which represent system components and interfaces, respectively. An SADT model captures multiple levels of detail in a hierarchical manner, with the top level being a general representation of the system. For a complete description of the SADT methodology, see Ross (1977).

Ross (1985) presented numerous applications, extensions and enhancements of SADT. During the 1970s, ITT Europe used SADT to design telecommunication switching systems and train personnel. Other commercial applications of the methodology include financial and transaction models, budget construction and tracking cycles, security systems, and curriculum development in education (Ross, 1985).

In 1978, the power of SADT as a communication and analysis tool was recognized by the United States Air Force and selected as the language to support the Integrated Computer Aided Manufacturing (ICAM) program. The ICAM program, thus resulting in the development of the ICAM DEFinition Methodology (IDEF0), later renamed as Integrated DEFinition 0 methodology. Ross (1985) states that "thousands of people from hundreds of organizations working on more than one hundred major projects" proceeded to use the methodology for system definition and design, as well as project management.

IDEF0 was developed for modeling a wide variety of systems that use hardware, software, and people to perform activities (U. S. Air Force 1981). An IDEF0 model consists of three components, diagrams, text, and a glossary, all cross-referenced to each other. The box and arrow diagrams are the major components of the model. In a diagram, a box represents a function and an arrow represents an interface. A box is assigned an active verb phrase to represent the function. An interface may be input, an output, a control, or a mechanism, and is assigned a descriptive noun phrase. Inputs (I) enter the box from the left, are transformed by the function, and exit the box to the right as an output (O). A control (C) enters the top of the box and influences or determines the function performed. A mechanism (M) is a tool or resource that performs the function. The interfaces are generally referred to as the ICOMs (see Figure 3). Each block may have many ICOMs.

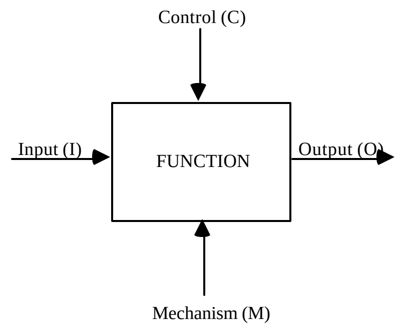

Figure 3. IDEF0 function box and interface arrows

Replacing function of the IDEF0 block with an activity and deleting a mechanism results in an IDEF3 block. The experience with various applications indicates that retaining the mechanism in IDEF3 useful. In the subsequent chapters the extended IDEF3 methodology will be used with mechanisms, time, reliability, quality, and other relevant attributes included, as necessary.

Each box in IDEF0 or IDEF3 has a specific node number and is connected by all relevant interfaces. The functions (IDEF0) or activities (IDEF3) may be decomposed into lower functions or activities. The resulting diagrams form a hierarchy of information that is summarized in a node tree. Figure 4 illustrates the decomposition principle of an IDEF3 model.

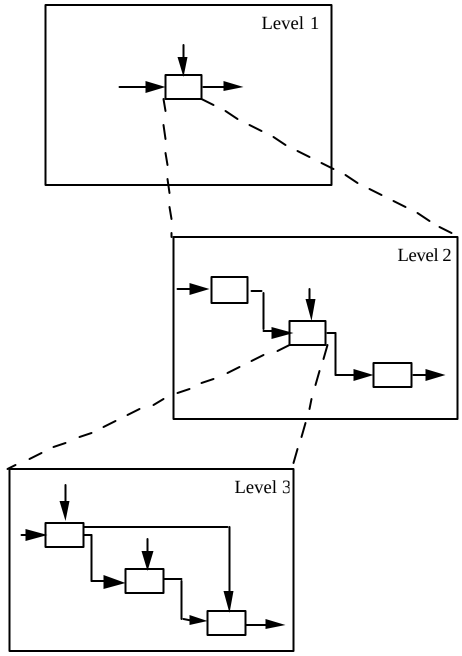

Figure 4. Decomposition in IDEF3 methodology

While IDEF0 provides a structured representation of the functions, information, and objects that are interrelated in a system, IDEF3 was created specifically to model processes. An IDEF3 model enables an expert to communicate the process flow of a system by defining a sequence of activities and the relationships between them. There are two basic components of the IDEF3 process description language, the process flow description and the object state transition network description. The two components are cross-referenced to build IDEF3 diagrams (Mayer et al. 1992).

The IDEF3 process flow description is made up of activities (units of behavior), links, and junction boxes. Examples of activities include *assemble parts*, *perform inspection*, or *evaluate proposal*. Relationships between activities are modeled with three types of links, precedence links, relational links, and object flow links. Precedence links express simple temporal precedence between activities. Relational links highlight the existence of a relationship between two or more activities, however, no temporal constraint is implied. Object flow links provide a mechanism for capturing object related constraints between activities and carry the same temporal semantics as a precedence link.

Relationships between activities are modeled with precedence links and logical connectors. The logical connectors (junctions) used for activities are classified as *and* (&), *or* (O), and *exclusive or* (X) as shown in Figure 5. Inputs or outputs that happen at the same time are linked with synchronous logical connectors, while asynchronous connectors represent inputs and outputs occurring at different time.

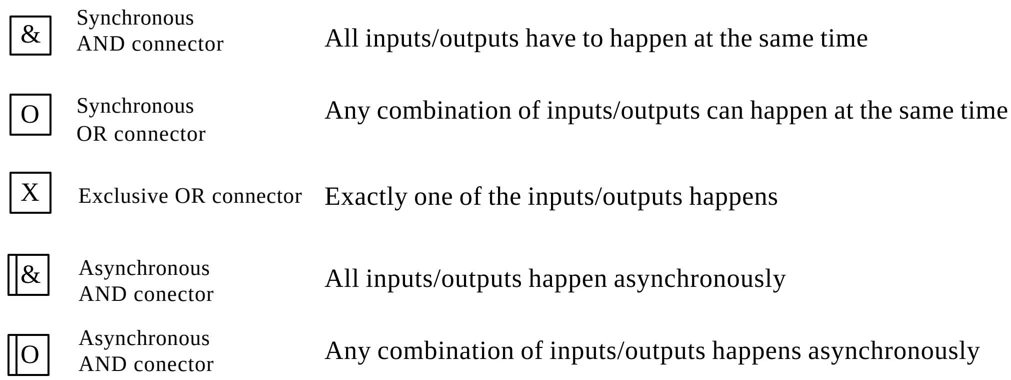

Figure 5. Logical connectors in IDEF3 process models

An example of IDEF3 process flow model with OR (O) and exclusive Or (X) junctions is shown in Figure 6.

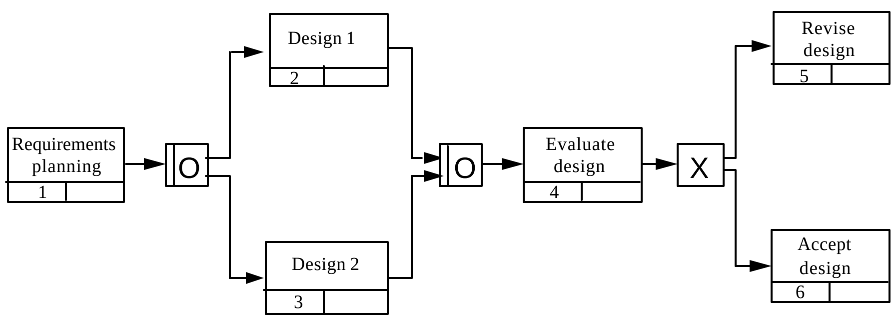

Figure 6. IDEF3 process model

4. COLLECTING INFORMATION

Much of the effort required constructing a process model is spent on collecting information. Some process modelers have identified excessive resource requirements (labor, financial, etc.) as a drawback to process modeling. This section discusses two approaches to information collection in the context of a process modeling effort. Following a brief description of the information acquisition method, a comparison is made, including advantages and disadvantages of each approach.

4.1. Group Meeting Method

Collecting information by group meeting requires the presence of all domain experts in a modeling team. The team meets regularly to perform tasks related to the process modeling effort. In this manner, the domain experts, rather than process-modeling experts, produce the model. Since this approach to information collection requires that each domain expert be knowledgeable in the area of process modeling, each modeling project begins with instruction on process modeling and, in particular, the modeling language being used (e.g., IDEF3). As the team continues to meet, information collection and model construction are performed simultaneously. This approach facilitates continual verification and validation of the process model by domain experts. Upon completion of the model, the modeling team feels a sense of ownership of the model, which encourages its use and maintenance.

4.2. Interview Method

Information collection by interview requires considerable effort by the modeling expert to identify domain experts, prepare an interview strategy, conduct complete interviews, and interpret data collected. It is also necessary for the modeling expert to develop a strategy for model verification and validation once models are constructed. This approach requires less effort of the domain experts. Knowledge about the process is presented to the modeling expert in a format chosen by the domain expert, rather than a format mandated by the process-modeling task. Convenient way of collecting information is to first sketch out a high-level process model and then collect information using a template shown in Figure 7.

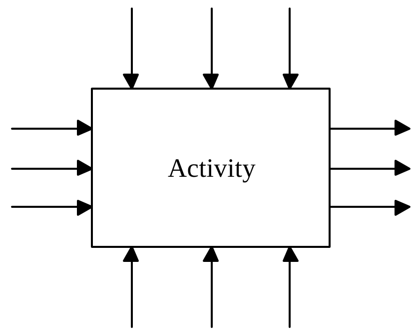

Figure 7. IDEF3 template

Information is typically collected on forms that are designed to capture the necessary information for the modeling language used. The information on the forms may then be arranged in a database for storage until the model is constructed. The domain expert is not required to participate in model construction. However, understanding the conventions of the process modeling language is necessary for model verification and validation. Once completed, the model needs to be analyzed by the domain experts, modeling expert or even a third party analyst.

4.3. Comparison of Information Collection Methods

The comparison of the information collection methods can be made based on two criteria: (1) cost of information, and (2) reliability of information. Figure 8 illustrates the relationship between these two measures relative to the number of domain experts participating in information collection. Although it is much less costly to collect information through interviews, there is less exposure to domain experts and, therefore, the information might be less reliable. This is largely due to the fact that several different people are likely to view the same process in several different perspectives. The group meeting approach collects observations from many different perspectives and uses group discussion as a way for generalizing information. Thus, the resulting information is more reliable, however, costly to collect.

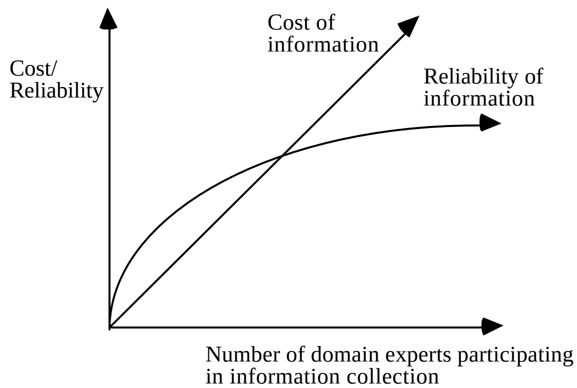

Figure 8. Reliability of information and cost of information
vs. number of domain experts

Adding domain experts to the modeling team increases the cost, however, their contribution may be minimal. Although here not quantified, the reliability of information and cost of information are likely to behave as illustrated in Figure 8. This provides justification for limiting the number of domain experts participating in the modeling effort. A well-conducted interview, targeting several domain experts, provides a balance between the cost and reliability of information. This assertion was reinforced by numerous observations made based on industrial process modeling projects.

A second factor discouraging the use of group meetings to collect information is the reluctance of management to allocate the personnel necessary for the group meeting format. As comprehensive modeling of industrial processes is a relatively new approach, it is often difficult to quantify the benefits of a process modeling effort. Busby and Williams (1993) cited neglecting economy as a limitation of process modeling. As a result of the lack of quantitative information (i.e., time, lot size, failure rate) in models, quantitative process changes are seldom recommended.

An advantage of using the group meeting approach is model ownership. Domain experts participating in a group modeling effort are more likely to use the process models developed. Upon model completion, domain experts may have a greater interest in the success of the project, which depends on how the model will be used. Also, by participating in model construction, domain experts are familiar with process modeling and the conventions of the modeling language. The primary user of models generated by the interview approach is often the modeling expert (or interviewer). A thorough analysis of the model may take place, however, maintaining the model for future use is less likely.

The current state of process modeling for analysis of manufacturing and design systems favors the use of the interview approach for collecting information. As the tool evolves, investing in group meetings for information collection may be easier to justify. However, an interview strategy employing a list of domain experts, information collection forms, a well-constructed database, and a strategy for model verification and validation is an effective alternative.

5. MODEL BUILDING CASE STUDY

In this section, a case study illustrating the process of inspection of plastic labels is considered. The "as-is" process is identified as level 1, six-activity model in Figure 9. These activities include: premount, inspect premount, set-up press, run press, inspection by operator, and inspection by press clerk.

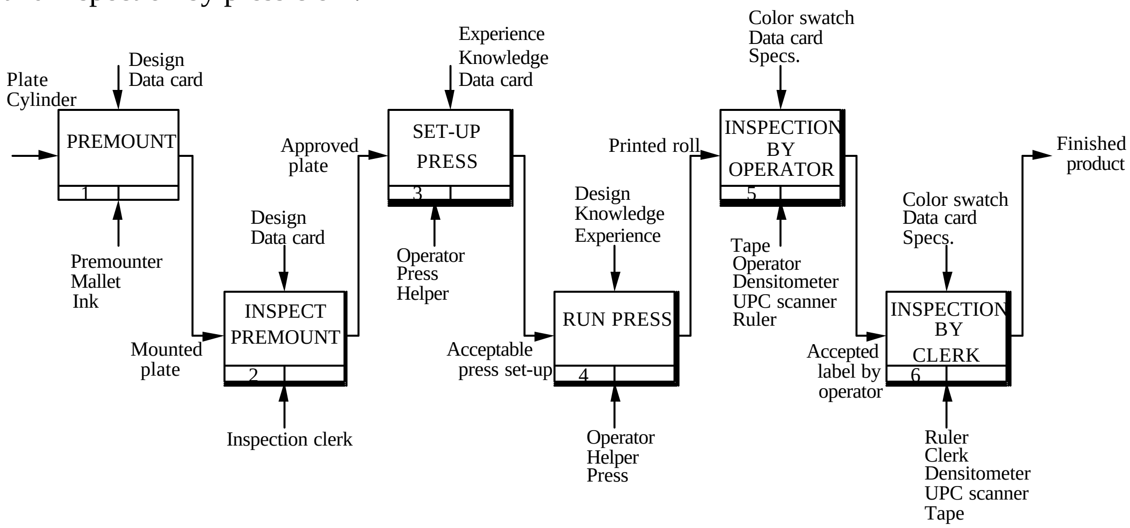

Figure 9. IDEF3 process model at level 1

The process model determines the sequence of activities and allows to identify improvements of the process. In order to successfully build an IDEF3 model, a systematic procedure has to be followed. Using such a procedure is especially important in building large-scale process models. Next, a model building procedure widely used by Rockwell International Corporation is presented.

5.1. Model Building Procedure

The label inspection process model in Figure 9 is constructed with the model building procedure involving eight steps:

Step 1. Define scenario;
Step 2. Identify and define appropriate activities;
Step 3. Arrange activities in phased sequence;
Step 4. Identify and define input and output objects;
Step 5. Determine object life cycle states;
Step 6. Determine decision points and flow junctions;
Step 7. Identify and define activity controls and mechanisms; and
Step 8. Define notifications and messages.

In Step 1 (Define scenario), the label inspection process is described in detail necessary to define the flow and change of state of the objects. The process involves various activities contributing to the finished product of (printed label). A customer places an order to have a label printed on a plastic roll that is later cut to make packaging bags. The order includes the customer's specifications such as: dimension of the bag, quantity, number of colors of print, design of the label, and other labeling requests. A plate that contains the impression of the label design is used in printing the plastic roll. The plastic roll changes state from not being printed to being printed, inspected, and accepted or rejected.

Step 2 (Identify and define appropriate activities), involves interviews with the employees who participate in each of the activities considered in the model. This includes the premounter (the person who mounts the design plate onto the printing press cylinder), the press operator, helper, and the inspection/press clerk. Each person describes their tasks and identifies the major activities within the labeling process. The activities are identified according to these descriptions and are defined as verbs in the activity nodes. The top level, six activities are identified as the following: premount, inspect premount, set-up press, run press, inspection by operator, and inspection by press clerk. The activity definitions are recorded in a glossary of terms for each level of the model.

The following activities at level 1 are defined:

Premount - A process where the premounter applies a rubber or photo polymer design (premount) to the cylinder so that it can be used on the press to make impressions onto the plastic rolls.

Inspect Premount - The inspection of the premount copy performed by the clerk in accordance to customer specifications.

Setup Press - A process where the premounted cylinder is installed in preparation for production.

Run Press - Running the plastic roll through the press where impressions are applied onto the plastic roll.

Operator Inspection - Inspection of a sample printed label performed by the operator.

Clerk Inspection - Inspection of a sample printed label performed by the clerk.

In Step 3 (Arrange activities in phased sequence), the sequential and parallel order of activities is established. The activities are ordered in time, e.g., the premount activity must be placed prior to the inspect premount activity (see Figure 9) .

In Step 4 (Identify and define input and output objects), the activity boxes a linked together. Input is always transformed while the output is produced. For example for the first activity in Figure 9, premount, the design plate and cylinder are inputs. These inputs are transformed to produce a mounted plate. The plate is mounted to the cylinder that fits into the printing press. The mounted plate is the output from the premount activity and also serves as the input to the inspect premount activity.

The glossary of inputs and outputs for level 1 is provided next.

Plate - Piece attached to the cylinder that allows for impressions onto the plastic.

Cylinder - Metal cylinder to which plates are attached.

Mounted Plate - A plate that is fully attached and centered onto the cylinder.

Approved Plate - Plate that was shown as satisfactory by the clerk.

Acceptable Press Setup - Press that is setup correctly for printing labels.

Printed Roll - A roll that has been run into the press.

Accepted Label by Operator - Label that the operator believes meets the quality requirements.

Finished Product - Labels classified as acceptable that meet all requirements and are ready for shipment.

During Step 5 (Determine object life cycle states), the input and output life cycles of objects are identified. For example in the level 1 model, the life cycle of the output from activity node 1, mounted plate, spans between the premount activity and the inspect premount activity. The life cycle states on the model are documented. In the level 1 activity node 2, inspect premount, the input and output plates are the same plates. Only the state of the plate has changed from being a mounted plate to an approved plate.

In Step 6 (Determine decision points and logical junctions), the areas where logical relationships exist are identified. In the model considered here no decision points have been identified at level 1.

Step 7 (Identify and define activity controls and mechanisms), involves determining controls and mechanisms for the activities identified in the model in Figure 9. Controls guide and influence the transformation, while mechanisms are the resources (e.g., people, systems, or devices) associated with performing the activity. For example in the model in Figure 8, the controls for activity 1, design and data card, are the guidelines that the mechanism, the premounter, uses in transforming the input, plate and cylinder, into the desired output, the mounted plate.

Level 1 definitions of controls and mechanisms are provided next.

Design - The object created on the plate for making impressions.
Data Card - Displays relevant material including the order number, tubing width, length, cylinder size, and diamond span for the labels.
Premounter - Employee mounting the plates to the cylinder aligning it correctly for the press.
Mallet - Used to apply the plate to the cylinder.
Ink - Used to get a print onto the paper for inspection of the mounted plate.
Inspection Clerk - Employee whose job is to help obtain a quality product through constant inspections of the plates used and the labels themselves.
Experience - Gained through repetitive attempts of running the process
Knowledge - Gained by employees through training and practice enabling him/her to perform the job efficiently and develop a better judgment.
Operator - An employee that runs the plastic through the press to print labels and checks the print quality. The operator is in charge of running and maintaining the press.
Press - Machine used to print labels.
Helper - Employee who aids the operator in running the press and checking the labels.
Color Swatch - A book used by the operator and the clerk in determining whether the right color schema is being used for printing.
Specs - Specifications or guidelines to be followed by all employees.
Tape - Used to perform the adhesion test.
Densitometer - Machine used to determine the whiteness of the print on the label.
UPC Scanner - Machine used to test that the correct UPC code was printed onto the bag.

Ruler - Used to measure the dimensions of the label to assure that it meets all specifications.

Step 8 (Define notifications and messages), involves identifying activities generating messages. In this case study, there were no messages identified on the model for any activity. A message, "Labels are printed" could be issued at activity 4, run press, of level 1 model in Figure 9.

The eight-step procedure produces the IDEF3 model in Figure 9. The same procedure is applied to all but the first activity at level 1 to generate a more detailed process models that are later integrated in an overall process model.

5.2. Process Models

The process of building detail level process models is provided next.

First, the design of the bag must be imprinted onto a permanent design plate. This plate is made of rubber or photopolymer and it may contain more than one impression. The plate is mounted to a cylinder that fits onto a printing press. The premount activity is the only one that is not decomposed into level 2 activities. It involves a premounter manually applying the design plate to the cylinder by means of a mallet and adhesive tape. Once the premounter mounts the plate, he/she runs a test label using ink and white paper.

The press clerk at the second activity inspects this sample, inspect premount. Activity 2 from the model in Figure 9 is decomposed into next level activities (see Figure 10).

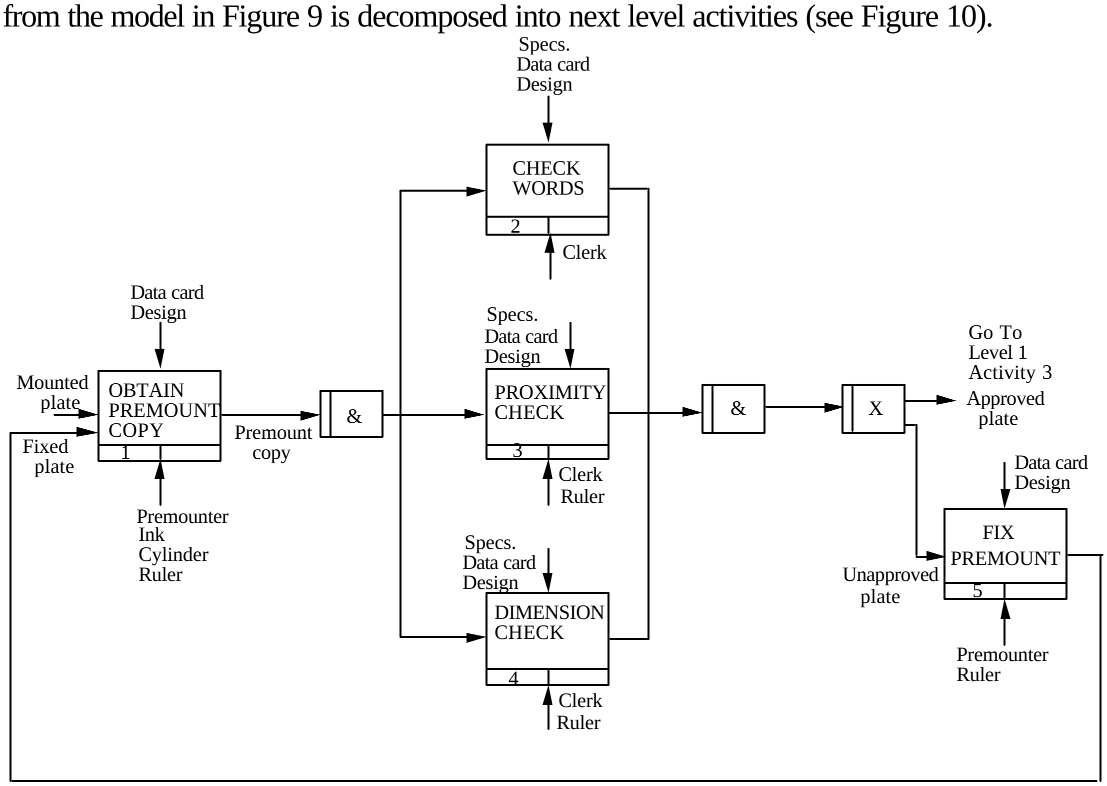

Figure 10. Premount process model (level 2)

Three glossaries for the premount process model at level 2 are provided next.

Glossary of activities for the premount process (level 2):

Obtain Premount Copy - Once the premounter has mounted the premount onto the cylinder, he/she makes an impression onto paper that is passed on to the clerk for inspection.
Check Words - Clerk checks the correctness of the text printed on a label.
Proximity Check - Clerk checks design, looking at the centerline for equal distances according to specifications.
Dimension Check - Clerk checks for appropriate location of the design on a bag.
Fix Premount - Adjustments made to the premount by the premounter when the clerk does not approve the copy.

Glossary of inputs and outputs for the premount process (level 2):
Mounted Plate - A plate that is fully attached and centered onto the cylinder.
Premount Copy - Copy made by spreading ink on the mounted plate.
Unapproved Plate - Plate shown as unsatisfactory by the clerk.
The clerk did not approve fixed Plate - Plate that has been adjusted after it.
Approved Plate - Plate shown as satisfactory by the clerk.

Glossary of controls and mechanisms for the premount process (level 2):
Data Card - Displays relevant information including order number, tubing width, length, cylinder size, and diamond span for the labels.
Design - The design that is created on the plate used for making impressions.
Premounter - Employee who mounts the plates to the cylinder and align it correctly for the press.
Ink - Used to get a print onto the paper for inspection of the mounted plate.
Cylinder - A large cylinder which used to mount plates onto to print labels.
Ruler - Used to measure the dimensions of the label to assure that it meets the specifications.
Clerk - Employee whose job is to help obtain a quality product by inspecting the plates and the labels.
Specs - These are specifications or guidelines that all employees must follow.

In the process in Figure 11, the clerk obtains a premount copy and inspects it for accuracy of the text printed, appropriate dimensions, and the location of the words on the bag. These three activities do not have to be performed in any particular order, however, the fix premount activity cannot start until all prior activities have been performed. This is denoted by the junction boxes classified according to logical semantics as *and* (&) and *exclusive or* (X). If an error is found in the printed sample, the clerk notifies the premounter that the plate is not acceptable. Once the premounter has fixed the label, he/she again submits it to the press clerk for inspection. This process continues until the clerk approves the plate.

Activity 3 from the level 1 process model in Figure 9 involves setting up the press for a production run of the label. The clerk and helper load the plate, establish the appropriate color as specified by the data card, and setup the plastic roll. The three activities can be performed in any order but must be completed before the adjust machine activity may begin. The former is denoted by the *and* (&) box.
The necessary machine parameters such as: pressure, tension, and temperature are adjusted depending on the type of order being processed. The orders vary depending on quantity, the

number of colors on the label, the number of impressions per plate, and printing on one or both sides of the plastic. The operator continually adjusts the machine until the press is acceptable for a production run. The process model corresponding to this activity is shown in Figure 11.

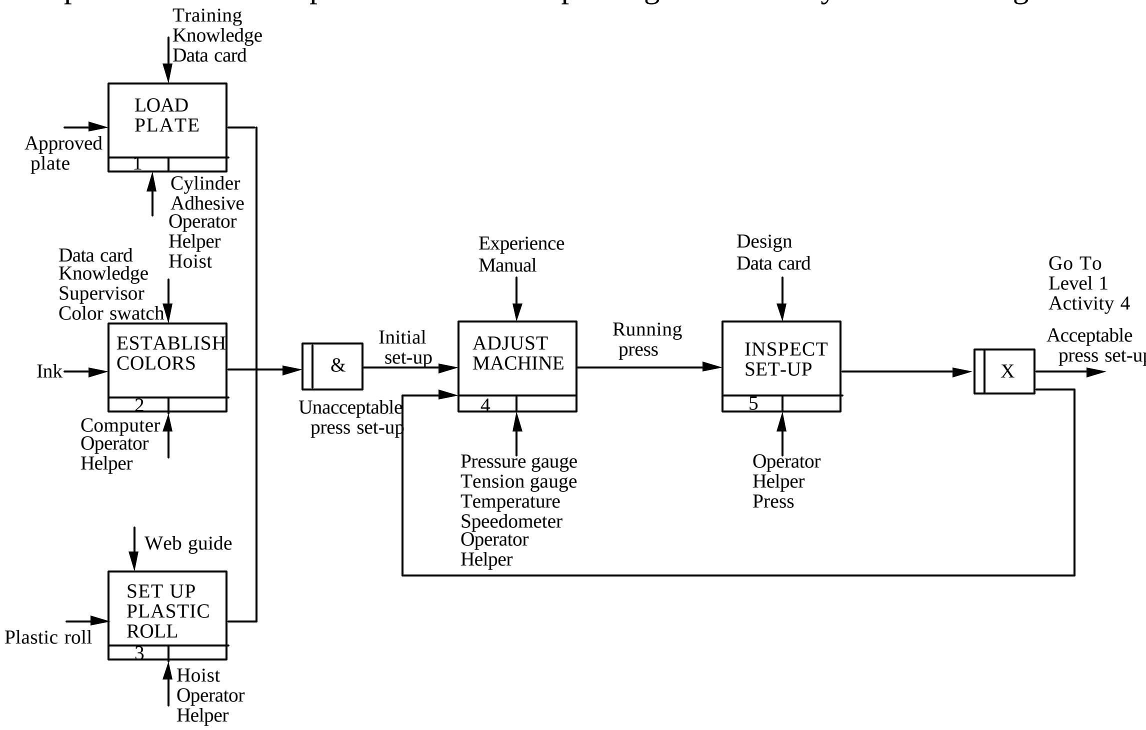

Figure 11. Setup press activity process model (level 2)

During activity 4 (Figure 9) the operator runs the press and continues to make the necessary adjustments to the machine. The plastic roll is run through the press for making impressions. Once an entire roll has been printed on, the operator adds a splice to the end of the printed roll and the beginning of an unprinted roll. This is performed so that the plastic roll may be changed over without stopping the press. A splice is just a piece of tape indicating where the two ends meet. The resultant model is shown in Figure 12.

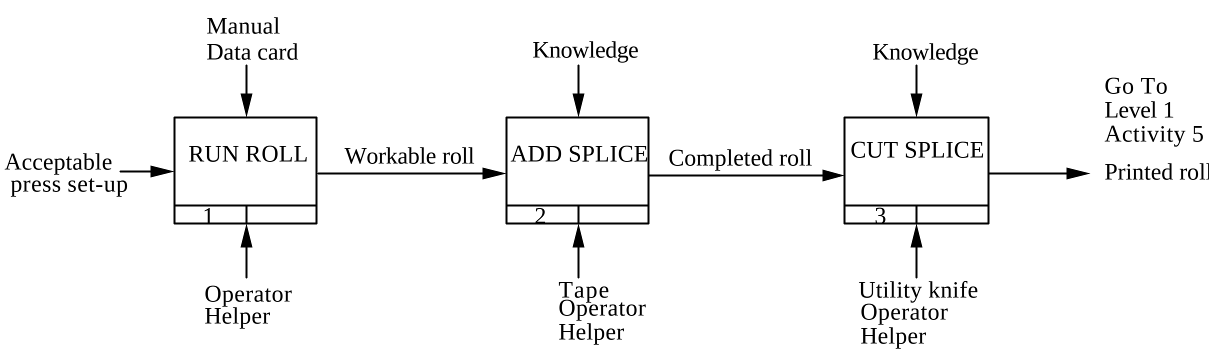

Figure 12. Run press process model (level 2)

Activity 5 (Figure 9) involves the operator running the roll until an appropriate sample label is obtained. The operator checks the label for adhesions, color, print accuracy, and legibility. These four activities can be done in any order but must be completed before the operator can determine that the sample label is acceptable. This is denoted by the logical junctions *&* and *X*. When the operator determines that the label meets the specifications, a sample is submitted to the press clerk for inspection. The level 2 model corresponding to this activity is shown in Figure 13.

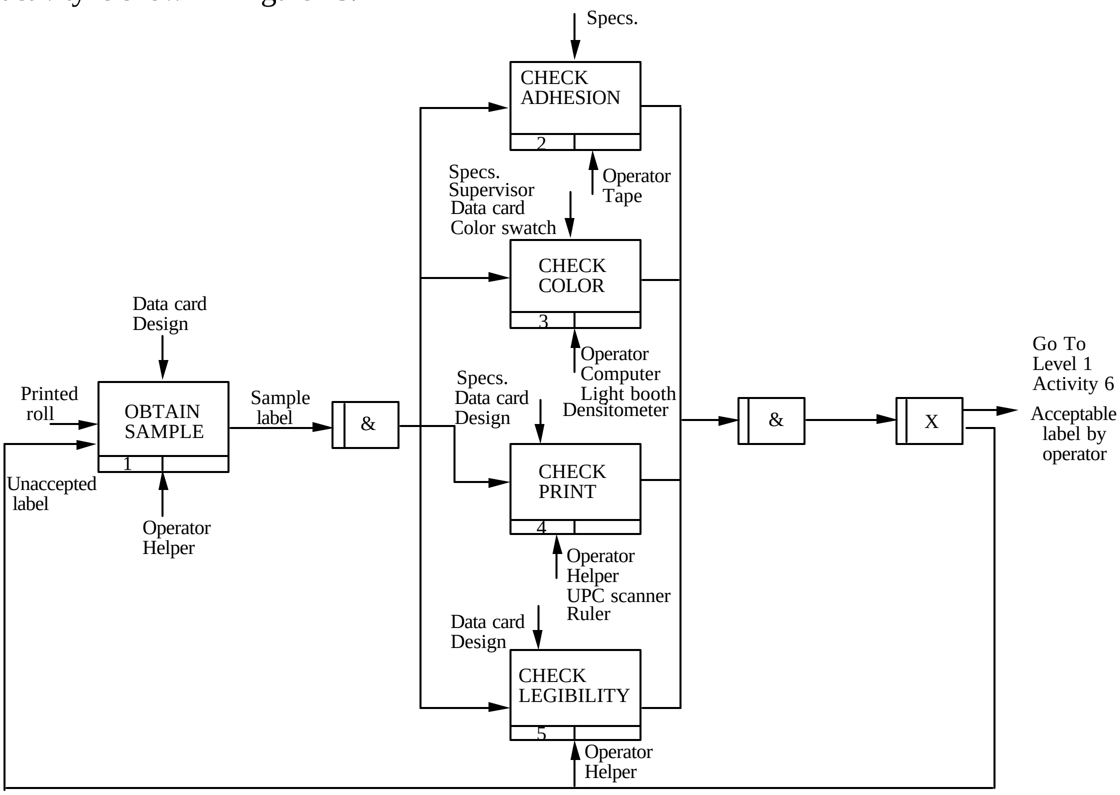

Figure 13. Operator process model (level 2)

Activity 6 from the model in Figure 9, inspection by press clerk, involves reinspection of the label that the operator finds to be correct for production run (see Figure 14). The press clerk obtains the sample label from the operator and performs, in any order, the following activities: check dimensions, check adhesion, check color, check print accuracy, and check legibility. Once these activities are performed, the clerk notifies the operator that the sample label is acceptable or unacceptable. If an error was found, the operator is required to make the necessary process parameter adjustments. The press clerk does not reinspect the label once and error has been corrected (see Figure 13).

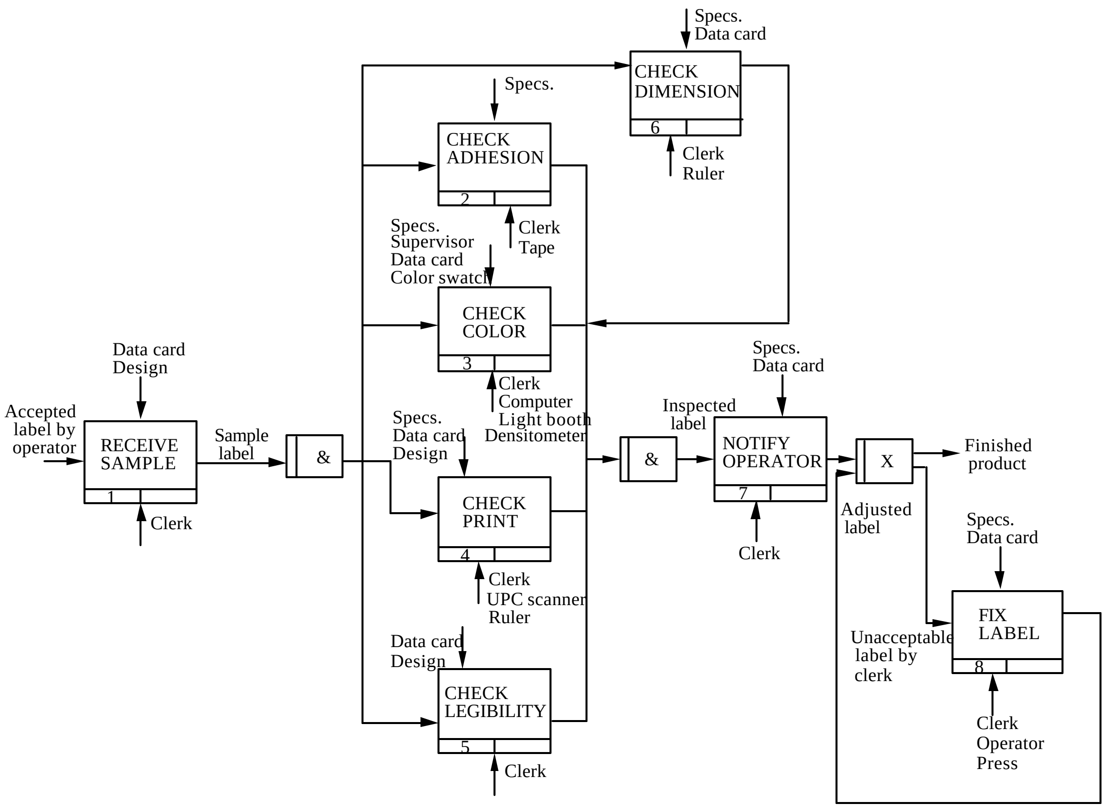

Figure 14. Press clerk process model (level 2)

The inspection performed by the press clerk is basically identical to the inspection performed by the operator. This step was established as a double check of the operator's performance. However, it is questionable whether this activity is necessary to the process.
In the course of statistical analysis it has been determined that for most labels this activity can be eliminated, thus resulting in meaningful savings.

VI. SUMMARY

In this chapter, various techniques for modeling processes were overviewed. Although differences between the techniques exists, the experience shows that understanding how to use a technique is often more important than the technique itself. The selection of a modeling methodology or tool is more a matter of a personal preference than the modeling environment requirements. An important attribute of a modeling technique is it extendibility, as it is not known yet how to design a universal modeling technique. Of all methodologies discussed in this chapter, IDEF0 and IDEF3 are perhaps the simplest to use and the easiest to extend. As the tool simplicity is often a warrant of its proper understanding and use, the IDEF methodology is frequently used in the remaining chapters. An industrial case study illustrating the application of IDEF3 methodology and a procedure for building models were presented. The most frequently recognized shortcoming of process modeling may be the lack of use and/or incomplete analysis of models. Due to the qualitative nature of models, mathematical techniques are difficult to apply. In the next chapters, various approaches for analysis of IDEF models are discussed. One way to analyze an IDEF0 or IDEF3 model is to represent it as a matrix or a graph.

Elements of the matrix may then be manipulated to identify the underlying structure of the IDEF model (see Chapter 2 for details).

REFERENCES

1. Ashworth, C. M. (1988), Structured systems analysis and design method, *Information and Software Technology*, **30**, 153 - 163.
2. Beekman, D. (1989), CIMOSA: Computer integrated manufacturing - open system architecture, *International Journal of Computer-Integrated Manufacturing*, **2**, 94 - 105.
3. Busby, J. S., and Williams G. M., (1993), The value and limitation of using process models to describe the manufacturing organization, *International Journal of Production Research*, **31**, 2179 - 2194.
4. DeMarco, T. (1979), *Structured Analysis and System Specification*, Prentice-Hall, Englewood Cliffs, N.J.
5. Doumeingts, G., Vallespir, B., Darricar, D., and Roboam, M., (1987), Design methodology for advanced manufacturing systems," *Computers in Industry*, **9**, pp. 271 - 296.
6. European Committee for Standardization (ECN) TC310 WG1, (1994), An evaluation of CIM modeling constructs: evaluation report of constructs for views according to ENV 40 003, *Computers in Industry*, **24**, 159 - 236.
7. Gane, T., and Sarson, C. (1982), *Structured Systems Analysis*, McDonnell Douglas Corporation, St Louis, MO.
8. Hatley, D. J., and Pirbhai, I. A. (1987), *Strategies for Real-Time System Specification*, Dorset House.
9. Kim, C., Kim, K., and Choi, I. (1993), An object-oriented information modeling methodology for manufacturing information systems, *Computers and Industrial Engineering*, **24**, 337 - 353.
10. Kusiak, A., Larson, T. N., and Wang, J. (1994), Reengineering of Design and Manufacturing Processes," *Computers and Industrial Engineering*, **26**, 521 - 536.
11. Mayer, R. J., Cullinane, T. P., deWitte, P. S., Knappenberger, W. B., Perakath B., and Wells M.S. (1992) *Information Integration for Concurrent Engineering (IICE)*, IDEF3 Process Description Capture Method Report, Armstrong Laboratory, Wright-Patterson AFB, Ohio 45433, AL-TR-1992-0057.
12. Mertins, K., Rabe, M., and Stiegenroth, H. (1993), Analyzing production systems using Petri net based functional techniques, *Proceedings of the International Conference on Industrial Engineering and Production Management*, FUCAM, Mons, Belgium, **2**, 961 - 970.
13. Page-Jones, M. (1980), *The Practical Guide to Structured Systems Design*, Yourdon Press, New York.
14. Peterson, J. L. (1981), *Petri Net Theory and the Modeling of Systems*, Prentice Hall, Englewood Cliffs, N.J.
15. Ross, D. T. (1977), Structured Analysis (SA): A language for communicating ideas, *IEEE Transactions on Software Engineering*, **SE-3**, 16 - 24.

16. Ross, D. T. (1985), Applications and extensions of SADT, *Computer*, April, 25 - 34.
17. Ross, D. T., and Schoman, K. E. (1977), Structured analysis for requirements definition, *IEEE Transactions on Software Engineering*, **SE-3**, 6 - 15.
18. Ward, P. T., and Mellor, S. J. (1985), *Structured Development for Real-Time Systems*, Yourdon Press, New York.
19. Mayer, R. J., Keen, A., and Wells, M. S. (1992), *Information Integration for Concurrent Engineering (IICE)*, IDEF4 Object-Oriented Design Method Report, Armstrong Laboratory, Wright-Patterson AFB, OH, AL-TR-1992-0056.
20. U. S. Air Force (1981), *Integrated Computer Aided Manufacturing (ICAM) Architecture Part II*, Volume IV-Functional Modeling Manual (IDEF0), Air Force Materials Laboratory, Wright-Patterson AFB, Ohio 45433, AFWAL-tr-81-4023.

QUESTIONS

1. What is a process?
2. What is a structured process model? Give two examples.
3. What is a unstructured process model? Give two examples.
4. What are the main methodologies for modeling process models?
5. What is more important the modeling tool itself or the user familiarity with the tool?
6. What is the main difference between IDEF0 and IDEF3 methodologies?
7. What are the main characteristics of the group meeting method of building process models?
8. What are the main characteristics of the interview method of building process models?
9. When the group meeting method and when the interview method should be applied?
10. How many activities may be included in a product development model?

PROBLEMS

1. Figure A1 illustrates four IDEF3 logical junctions.

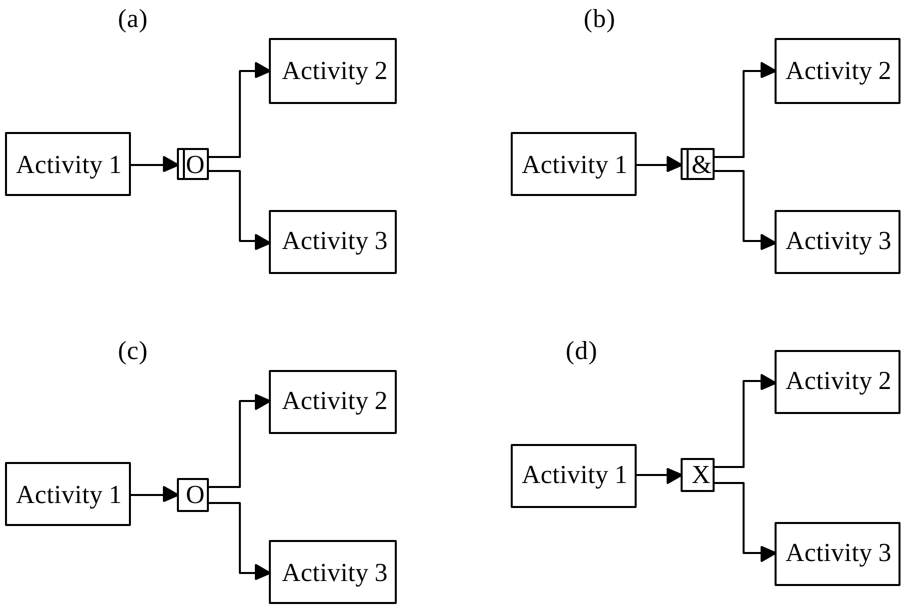

Figure A1. IDEF3 logical junctions

(a) Name each junction.
(b) Is it possible that the outcome of (a) and (b) to be equivalent? If yes, when?
(c) Is it possible that the outcome of (c) and (d) to be equivalent? If yes, when?
(d) Is it possible that the outcome of (a) and (d) to be equivalent? If yes, when?
(e) Is it possible that the outcome of (b) and (c) to be equivalent? If yes, when?

2. The IDEF3 process model in Figure A2 includes a few errors.
Draw a correct process model. Name all inputs, controls, outputs, and mechanisms (ICOMs) in the corrected model.

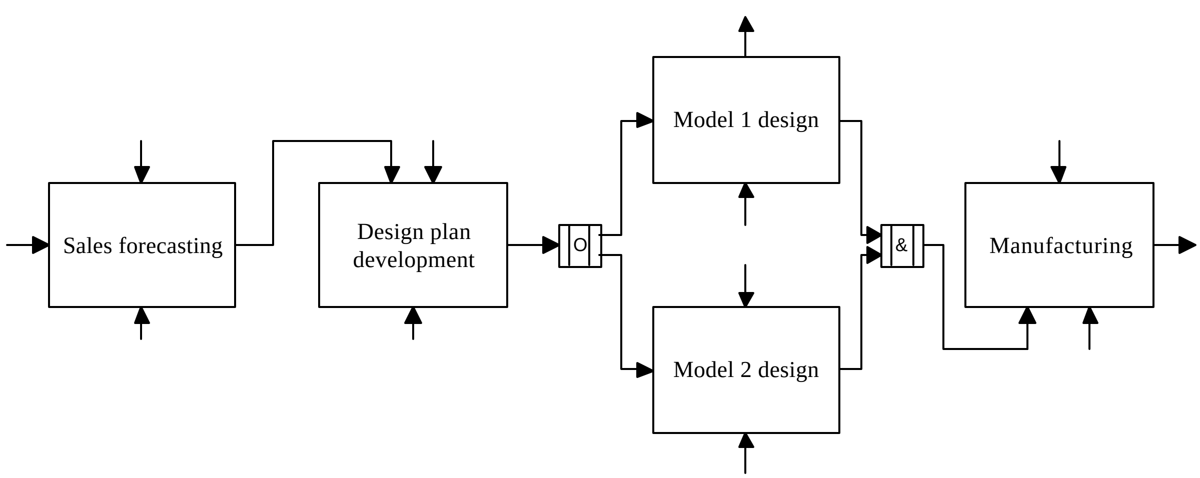

Figure A2. A generic IDEF3 process model

3. Develop and draw an IDEF3 model of the process of buying a home computer and software.
Scope: The process should include getting money, visiting stores and web sites, getting a discount (when applicable), and installing the software.
For the IDEF3 model developed, define:

- activities
- inputs
- outputs
- mechanisms
- controls

4. Develop (draw) an IDEF3 model of the process of buying a new car. The process should include trading in an old car, arranging a loan, and negotiating the price.
For the IDEF3 model developed, define:

- activities
- inputs
- outputs
- mechanisms
- controls

5. For the IDEF3 model in Figure A3:
(a) Define all inputs, outputs, and controls,
(b) Give an example of input, output, and control defined in (a)

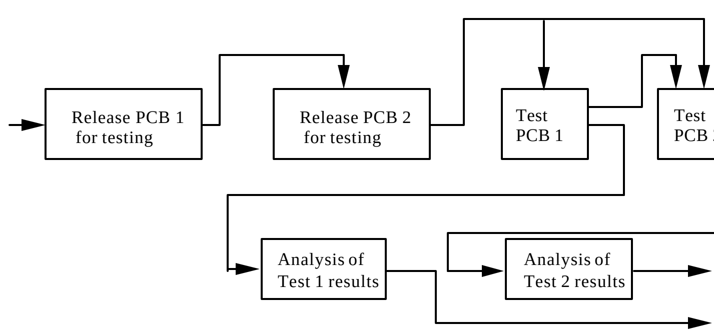

Figure A3. Incomplete IDEF3 model

6. The manufacturing system in Figure A4 includes two machining centers MC1 and MC2 of the same type (each having the capability to perform operations O1 through O4); two serially arranged assembly stations (each dedicated to one family of parts) and a packaging station. Reengineer the process so that the machining and assembly setup for the two product families is reduced. A setup is defined here as the effort to prepare a machine (assembly station) for processing the next part family. Draw an IDEF3 model of the reengineered process.

A. Kusiak, *Engineering Design: Products, Processes, and Systems*, Academic Press, San Diego, CA, 1999.

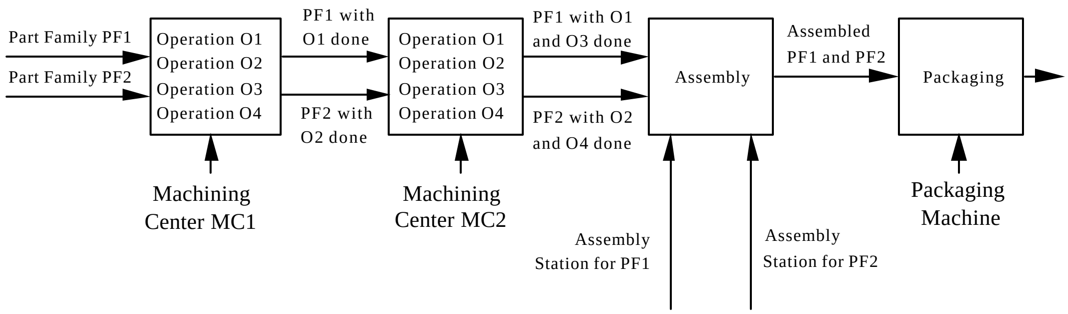

Figure A4. The IDEF3 model of a manufacturing system