Chapter 5: The Process Design and Information Modeling

The way existing design tools represent information and communicate with each other is often not consistent. The design tools used in the stages of part design, tool design and process design, do not communicate well with each other, since different representations are used by these tools. IPPPIS is developed using information models in this work integrates product design and manufacturing cycles in a systematic way to facilitate the swift, cost-effective progression of new products from concept to final product. Information models are data structures that represent information content. A large amount of information in manufacturing planning needs to be computerized so that IPPPIS can use them. All this information is identified and represented by information models.

In this chapter, task clarification of the IPPPIS is presented in the first part. The process planning approach is detailed with functional structure used in the system. Information content in the IPPPIS is discussed, with different approaches and modeling methods to represent this information. This chapter concludes with details on the systematic information modeling hierarchy and UML class diagram to represent this hierarchy of information.

5.1 Introduction

Traditionally, when designing a new product, a team of designers usually participates in what is known as a product development cycle. In general terms, the first stage in designing a new product is that of concept design, in which overall needs and aims are addressed. Next is the initial design stage. The initial design stage comprises of steps for designing the part, then choosing the materials and determining the process to make the part, and finally designing the tool to make the part. For example, if the part is to be a
computer keyboard, first the size and shape of the keyboard (i.e., the part) is determined by a part engineer. Next, when the part has been designed, a second engineer determines the design of the tool that can be used to make the keyboard. Separately, a process engineer determines the materials and process to make the part, selects the process parameters and the rate of production of the process. The next stage, after a prototype of the part has been made; revision of the design is done. The above-mentioned steps of part, tool and process design are repeated until a satisfactory part is produced, satisfying both design and cost of production targets. Traditionally, each of the above steps is carried out sequentially, usually by different people. One person may design the part, another tool, and a third the process. Collaboration between these designers is usually minimal. Where many revisions have to be made, numerous iterations are needed and a long period passes until a satisfactory part is produced. Using present design approaches and tools, there is incomplete knowledge of the required manufacturing steps to produce the part, and inadequate consideration of the variety of other downstream influences that shape time to market, marketplace acceptance, and product longevity. Often neglected, but of importance in part design are the constraints added by environmental concerns relative to the product and its process of fabrication. These flaws lead to a multitude of costly and time-consuming design reworks, or difficult process modifications, as unanticipated problems occur that must be rectified.

5.2 Task Clarification of the IPPPIS

A model has have been developed using object oriented technology, after analysing the fundamental elements necessary for modeling manufacturing and process planning framework used in collaborative design and manufacturing in a selected machine tool manufacturing company. The performance of this model is shown by using real world case. The
manufacturing information based design tool integrated with an intelligence
design system developed, can be used for collaborative design and
manufacturing. This will support machine tool designers’ effort to achieve
cost effective and timely design by performing various tasks as described
below:

1. Establishing a library of default process steps and product design
data, including a parts list library of preferred components associated
with model configurations and sequence of events configurations for
the part.
2. Initiating a new part design for the part based on the library and
periodically checking new part design against a predetermined set of
new part requirements.
3. Establishing a complete sequence of events for new part design for
the part.
4. Identifying a default process step in said library of default process
steps incompatible with production of the new part design within the
part requirements.
5. Replacing the default process step with a revised step compatible
with production of the new part design within the part requirements.

To enable correct part design, the present model provides the part designer
with all relevant information effecting the part design (such as, for example,
information about the processes and materials used to make the part) while
the part is being designed. Further, the designer and the process designer are
also provided with all relevant information affecting their designs. The
information supplied to the part designer, the tool designer and the process
designer is the same "model" of relevant information. This model of
information can be shared concurrently by each designer. Design decisions
made by each designer can be included as a factor in the design decisions by
other designers. As such, the functions of part designer, tool designer and
process designer often merge and overlap when the present invention is
utilized. The method for development of the proposed CAPP in IPPPIIS is
given in Figure 5.1.
5.3 Process Planning Approach in the IPPPIS

The two types of approach used for CAPP systems are referred to as variant and generative. The first approach consists of using generic process planning for a part family that is modified to adapt to the characteristic features of each one. Because of its nature, this approach does not fully address the problem of generating a plan. However, it does enable the integration of process planning into other company activities. The generative approach consists of generating a process plan each time; automating this stage and enabling its integration into other activities. There is need for setting up models that enable the development of CAPP systems with an optimum degree of generality and flexibility. Some of the aspects that, largely, determine the operation and application of a CAPP system are as follows.

* Planning Methodology: All the steps needed for process planning, i.e., it sets out the planning method. The methodology must enable work to be carried out on any geometry of parts, while at the same time being
dependent from manufacturing technology and resources. Consequently, the CAPP system will not only be able to operate on any parts families, but will be able to develop in line with the company, the technology implemented and the resources available. An important aspect of the planning methodology used is the generation of alternative process plans that provides several route sheets for the parts. This optimizes the planning and control of production and the result is a high degree of productivity in the manufacturing system.

The requirements demanded of a methodology for developing general and flexible CAPP systems for machining parts are as follows.

- Independence from the geometry or shape of the part.
- Take into account not only the machining processes, but also other processes closely linked to it in the final stages of part manufacture such as heat treatment, coating, cleaning and deburring operations.
- Independence from production resources, operating on various types and configurations of these resources.
- Assess process plans with regard to the quality obtained in the product and the manufacturing costs, by performing an in-depth study of the resources used.
- Enable the generation of alternative process plans at all levels: processes, machines, fixtures, etc., assessing the production cost in each case.

* Architecture of the System: * Structure of the various elements making it up (functions, databases, etc.) and, more importantly, their interrelations. The architecture of a CAPP system must be able to support the planning methodology, while controlling its integration with other production system tasks.

* Information System: * specifies the data stored and their structure. Here the data used for representing parts, production resources and the description of processes available is particularly noteworthy. The information system must
be sufficiently general to represent any type of part and production resources. In this way, it allows general CAPP systems that are not dependent on parts, resources and technology.

The overall functions a CAPP system must have to make a process plan is extremely wide and varied. It must decide on the processes to be applied, their sequence and all of the resources to be used (machines, fixtures, tools, etc.). It must also ensure that the plan is valid at all times. As a result, each of its functions—although they operate together—focuses on specific, individual aspects. Among these functions, it is worth pointing out those that play a role in determining alternative plans. Because of their impact, they subsequently provide production planning control with greater options for obtaining high performance and flexibility in the manufacturing system.

5.4 Process Planning in the IPPPIS

Using the methodology process planning was done as shown in Figure 5.2. As mentioned above, the methodology starts from part information using machining features and its quality specifications. Its purpose is to establish all the viable alternative process plans, organizing them as sequences of phases, set-ups and operations.

* **Raw Material:** The first stage is to determine the raw material for machining the part. In a part model based on machining features, this raw material may be specified in the part model. Otherwise, this stage is responsible for specifying raw material taking into account other forming processes to obtain the preform (moulding, plastic deformation, cutting, welding, etc.). Therefore, this function includes process planning for processes that are different to machining. As a result, it is placed in the context of a generic planning or a macro-plan before machining.
Machining Feature: Using the part’s input model, this step consists of determining the model that the system will use to work with. This model is based on the definition of the part with certain machining features, complying with two conditions that distinguish it from the input format. The first condition is that all the machining features will be simple. For this purpose, compound features will be broken down into simple features. The second condition is that the properties for each of these simple machining features include both their shape and quality specifications for manufacture. It also considers the part as a whole, taking into account factors such as, for example, its accessibility. This means moving on to a model of machining features where the definition of each feature contains individual information as well as information about the rest of the part.

Assignment of Processes and Operations: Using the part model obtained in the previous stage and based on simple machining features, it is possible to determine the processes that are technologically able to achieve them and the operations necessary for these processes. The strategy proposed is to assign to each feature all the processes and operations that may be
applicable to that type of feature and that are wholly or partly capable of satisfying any of the quality requirements (Figure 5.3). This bears in mind that a process may be applied to the type of machining feature, as well as the ability of its operations to satisfy the feature’s quality specifications. For this purpose, process modeling establishes the capacity margin for each operation, including the machines, fixtures and tools that provide it with support. This definition of capacity enables processes and operations to be initially assigned without considering specific resources (machines and fixtures) [90].

Figure 5.3 Assignment of alternative processes and operations

*Assignment of Machine:* All possible machines are determined for each of the alternative processes and operations proposed in each machining feature of the part. This offers a wide range of solutions to the problem at phase or machine level. It enables selecting those machines that are most suitable and that satisfy the constraints imposed on the sequences of operations. The potential use of a machine in an operation depends on its assessed capacity. This not only takes into account the capacity values of the machine itself, but also the productive resources it uses, such as fixtures and tools. As the number of alternative machines increases, sequence phase options quickly increase and in the interests of obtaining time-efficient procedures, a
reduction in alternatives is applied. Equivalent machine groups are set up for machining the part. This is done by grouping different machine options according to similarity.

*Machine Sequencing:* At this point, where all possibilities have been determined for operations, processes and machines for each machining feature; the possible phases and their sequence are determined. First, the precedence's these operation sequences must comply with are established. These affect all levels, e.g., the finishing process of a machining feature follows the rough operation. Once these precedence's are established, a general algorithm for the formation and sequencing of phases is applied to develop all the options. Before sequencing, machine group alternatives that might lead to non-optimal sequences are rejected. The criteria used by the sequence formation algorithm correspond to the minimum number of phases. Lastly, sequences furthest from the optimum were rejected. For this purpose, the operational cost of each sequence was estimated and the least favorable sequences were deleted. This leaves the solution open for the following stages in which the plan will be specified.

* Assignment of Fixture: * For each of the phases obtained in the previous point, a possible fixture or alternative set-ups are established. At this level, it is worth pointing out that the number of alternative fixtures for each machine stage should not be excessively high. Furthermore, some of the phase sequences proposed above is rejected because of set-up problems. For each of the set-ups, it is essential to conduct the capacity study to obtain part specifications and bear in mind the specific capacities of each fixture. Each of the set-ups sequences proposed is then assessed by using a cost estimate.

* Assignment of Tools: * The problem of assigning tools to operations and sequencing is resolved within each set-up in order to reduce the number of tools and settings. As a result, the tools assignment depends on the sequencing of higher levels, such as phase and set-up levels. The reason for
this is that tools represent the least critical and most flexible resource.

One of the most outstanding features of the proposed methodology is the generation of alternatives with a hierarchical construction of plans, i.e., first determining the required phases or machine sequence, then the set-ups and finally the tools. As a first step in this procedure, all the possibilities are determined for processes and operations applicable to the part’s machining features. In this way, a process of grouping these processes was initiated while gradually opening up to other possible alternative solutions. As the plan is specified, many of these are rejected as impractical and are arranged according to their estimated cost.

The methodology proposed is supported by several models of generic information that enable the: definition of a part on the basis of machining features; the definition of production resources and their capacities; and the definition of processes to be considered. Together with the methodology itself, these models become a key factor in applying the system, to various production environments with different part forms and geometry. Information content in the developed system is shown with the help of a flow diagram for better understanding in Figure 5.4.

5.5 IPPPIS Functional Structure

The implementation of the methodology has been hierarchical structured. At the initial level (Figure 5.5) five functions have been taken into account.

* **Determine Raw Material.** Function responsible for deciding raw material for machining the part. It establishes high-level process planning in order to determine the preform or raw material.

* **Determine Options for Operations, Processes and Machines.** Function for selecting production resources needed to establish the process plan at phase level. Considering alternatives for these resources represents a key factor in establishing alternative process plans at all levels.
Figure 5.4 Information Content in the System Developed

*Select and Sequence Machines and Fixtures.* The aim of this function is to establish alternative phase and set-up sequences for manufacturing the part. It uses the alternative production resources proposed for the previous function.

*Determine Tools and Work Conditions.* Function for specifying micro planning details relating to operations level. This function determines the tools, work conditions and sequence of operations for each set-up in order to reduce the number of tool changes.
*Write Results.* Generator functions for the output of results in suitable formats, such as: route sheet, raw material requirements, cost data, etc.

Figure 5.5 First Functional Level of CAPP System.

The function determines options for operations, processes and machines’ have been structured in five functions. Figure 5.6 shows the functions and information exchanges, including the following:

*Analyse and Describe Machining Features.* The aim of this function is to establish the description of the part based on the information model proposed for machining features. The advantage of this characterization is to provide a complete, individual description of the machining features. This enables the independent assignment of processes, operations and machines.

*Select Possible Processes and Operations.* This function assigns processes and operations to each machining feature. In the assignment, alternative processes and operations were deemed to those that are able to
satisfy feature specifications, as well as those that only manage to come close. These are structured in a general way following the same pattern for all the processes and operations. This knowledge is independent from process and operation capacities. A general information model was used for their description. The developed rules store only qualitative information about process and the operations, such as: type of machine; comparison of requirements and capacities; assessment of optimization for the process and operations, etc..

![Diagram of Functional Model Proposed for Function Information Exchange](image)

*Making Operation Alternatives Compatible.* Once all the alternative processes and operations have been proposed, they must be made compatible as described in the methodology. This allows the alternatives to be processed in a general way, enabling them to be combined independently

149
with each other. Compatibility requires equivalence or exchangeability between processes and this is achieved by homogenizing their ability to produce the machining feature. A general functional procedure has been developed for this.

*Select Capable Machines. The aim of this function is to assign alternative machines to each of the alternative processes. However, this is via the machine groups described in the methodology. The machine groups are made up of equivalent or exchangeable machines that are specifically determined for each part. Three important aspects are taken into account when making up the groups, such as: the type, quality and size of the machines. In connection with the aspects of quality and size, the proposed operation requirements for the part are taken into account. This enables machine groups to be set up that suit the parts requirements most accurately.

*Cost Estimation. The aim of this function is to estimate the cost of operations, specifying it for each of its alternative processes and optional machines. For this purpose, it uses a model of costs that takes into account the cost factors attributable to the machine and the individual factors of the process and operation. These costs will be necessary during the phase or machine sequencing to establish the estimated cost of each one of the sequences.

5.6 Type of Information used in the IPPPIS

There are basically four categories of information in the IPPPIS:

**Design Information:** Design information is the input of IPPPIS. Generally, part information, including part geometry information, tolerance information, functional information, and production information (production volume, material), are analyzed and stored in an IPPPIS.

**Manufacturing Resource Information:** Manufacturing resources may include cutting tools, machine tools, fixtures, and inspection tools. Some of
them are standard tools and are readily available. Others were designed specifically for particular processes used in manufacturing plans.

**Manufacturing Knowledge:** Manufacturing knowledge is used to help engineers make the right decisions. It is composed of general manufacturing rules and best practice knowledge that is summarised by manufacturing industries.

**Information Generated by IPPPIS:** Information models to describe the result generated by IPPPIS systems. This consists of process information, including the utilization of manufacturing resources and process parameters, setup information, and manufacturing planning information. Several information models have been provided for representing and storing the above information. Group technology and coding systems have been applied to represent part design information by fixed-length codes or flexible-length codes. These codes have been used to group parts into part families that link them with standard process plans. Decision tables and decision trees have been used to computerize the decision-making procedures that incorporate manufacturing knowledge. Relational databases have been employed to store part design and manufacturing resource information.

5.7 **Design Retrieval and Cost Reduction Through Automated Coding**

In designing new products, the ability to retrieve drawings of existing components is important if costs are to be controlled by preventing unnecessary duplication of parts. Variety reduction is a popular technique for controlling the costs associated with creating a new part number including administration, stockholding, obsolescence, service and reduction in the economies of scale. Actual figures for creating a new part are difficult to derive reliably but are often quoted as thousands of pounds per part per annum [91]. Many companies use standards and preferred item catalogues to control the proliferation of common parts such as fasteners, but this
approach is ineffective for non-standard items. What the designer requires is an easy means of retrieving a suitable existing part that is significantly quicker than the alternative of drawing a new one.

Part numbering systems exist in most companies and they can be used to retrieve information about parts. However, since these systems exist primarily to provide unique identity, they offer very limited search facilities. Product data management (PDM) systems do allow searches on other fields, properties or special keywords and most CAD systems allow text searching on the drawing description. The effectiveness of all these techniques depends on rigorous application of naming conventions and none allows the search to be based on the part geometry.

Many coding and classification systems have been developed for use in manufacturing industries, either for variety control or cell family formation [3]. Once a part's shape information has been encoded into a code 'number', similar parts can be found since they will have identical or similar code numbers.

5.8 Information Content in the Model

A systematic information modeling ontology have been used to describe the information relationships and associativities, in which Object-oriented Systems Analysis (OSA) approach is employed to establish the information models. The concept of an object is derived from software engineering and is considered as the computerized representation of entities in the real world. The OSA uses three kinds of models: an Object-Relationship Model (ORM) describes the static characteristics such as information composition of objects; an Object-Behavior Model (OBM) defines the dynamic characteristics of objects; an Object-Interaction Model (OIM) pictures information association and interactions between objects.

By using the OSA approach, a system model has been used for the high-
level view, as shown in Figure 5.7. It is divided into four object packages: part design and manufacturing planning packages; manufacturing activity knowledge; manufacturing resource; and manufacturing cost and time packages. The arrows in Figure 5.7 indicate the relationship and interaction between these object packages. The part information is the input, which is composed of features and the relationships between features and feature’s manufacturing strategies are linked with features. Manufacturing planning package includes both part level and machine-level decision-making strategies. The manufacturing activity knowledge package provides the knowledge constraint to control the manufacturing planning behaviors. The manufacturing resource package provides the description of the manufacturing resources, such as machine tools, cutters and fixtures in specific manufacturing companies. Each object packages in the system model will be broken down into low-level objects.

Figure 5.7 Information Modeling
5.8.1 Conceptual Design Feature Information Content

In the conceptual design stage, the combinations of geometric patterns as well as interactions among them realize product functions are looked at. Conceptual design feature geometry only includes those geometric entities, which are indispensable for realizing the function. A conceptual design feature definition is given Table 5.1.

Table 5.1 Conceptual Design Feature

<table>
<thead>
<tr>
<th>Attributes</th>
<th>semantics; functions; behaviors (input and output); dimensions; tolerances; and material specifications</th>
</tr>
</thead>
<tbody>
<tr>
<td>Constraints</td>
<td>spatial constraints; functional constraints</td>
</tr>
<tr>
<td>Geometries</td>
<td>critical geometrical-entities</td>
</tr>
<tr>
<td>Methods</td>
<td>create; edit; check validity; query information</td>
</tr>
</tbody>
</table>

5.8.2 Detailed Design Feature Information Content

In the detailed design stage, the conceptual design, i.e., critical geometric entities and interactions among them, are further refined into complete product geometries and specifications. A primitive detailed design feature is defined as a set of related geometric entities and has is given as Table 5.2.

Table 5.2 Detailed Design Features

<table>
<thead>
<tr>
<th>Attributes</th>
<th>Semantics; patterns; parameters (e.g. diameter); dimensions; tolerances; positions; orientations; material; roughness</th>
</tr>
</thead>
<tbody>
<tr>
<td>Constraints</td>
<td>geometric constraints; algebraic constraints</td>
</tr>
<tr>
<td>Geometries</td>
<td>parts; assembly; components; features; geometric and topological entities; references; derived entities</td>
</tr>
<tr>
<td>Methods</td>
<td>create; edit; check validity; query information</td>
</tr>
</tbody>
</table>

5.8.3 Machining Feature Information Content

Feature-based process planning covers two processes, operation planning and machining passes. Machining operations can be defined according to setup or cutter changes. Machining parameters are determined in the scope of each pass. A primitive machining feature is defined as a set of related
geometric entities that represents the volumes removed or faces generated during a machining cut. The primitive machining feature definition is given below in Table 5.3.

Table 5.3 Machining Features

<table>
<thead>
<tr>
<th>Attributes</th>
<th>semantics; machine information; tools; machining parameters; operational and locating datum; dimensions; tolerances and roughness of the machined faces</th>
</tr>
</thead>
<tbody>
<tr>
<td>Constraints</td>
<td>Machining constraints (power, workspace, etc.); tool constraints (cutter radius, flute length, etc.); geometric constraints</td>
</tr>
<tr>
<td>Geometries</td>
<td>features; geometric and topological entities describing the workpiece before and after the operation or cutting pass</td>
</tr>
<tr>
<td>Methods</td>
<td>create; edit; check validity; query information</td>
</tr>
</tbody>
</table>

5.9 Object-Oriented Systems Analysis (OSA) Approach

O-O modeling is recognised as a powerful tool to model real-world systems. An object is an encapsulation of data and procedures (or methods) that operate on the data. An object can be defined as an existing entity in the real world such as a part, a manufacturing plan, and a machine tool. The real world can be considered as a group of interacting objects. The interaction is described according to the way that human beings think. Therefore, O-O modeling can create information models that exhibit close resemblance to real world systems, and the main task of O-O modeling for a system is to identify objects and analyze their interaction within the system. Here are some basic concepts used in O-O modelling:

Object: - An object is a bundle of variables and related methods. A variable is an item of data named by an identifier. An object implements its behaviour with methods. A method is a function associated with an object.

Class: - A class is a set of objects that have shared properties. A class is represented by a rectangle in the diagram.

Encapsulation: - Packaging an object's variables within the protective custody of its methods is called encapsulation.
**Relationship:** - A relationship establishes a logical connection among objects.

**Inheritance:** - Inheritance is a kind of relationship between objects. O-O modeling allows classes to be defined in terms of other classes. For example, a rotational part is a kind of a part. Therefore, the part is a superclass and the rotational part is a subclass.

In the research, the OSA approach is used to analyze the information in the IPPPIS. The ORM is used to represent the static relationships between objects. The OBM describes the behavior of individual objects and how objects respond to dynamically occurring events and conditions. The OIM expresses the information associations between objects.

### 5.9.1 Object-Relationship Model (ORM)

An ORM is created to represent the static relationships between objects. ORM’s are usually described by ORM diagrams. Users can define their own relationships with the specific relationship name attached to ORM diagrams. There are two basic relationships used frequently, and specific symbols are assigned to represent them in ORM diagrams.

1. **Generalization-Specification Relationship**

   In an ORM diagram: a rectangle represents an object; an ellipse represents a variable of an object; and a transparent triangle represents the generalization-specification relationship (Figure 5.8). The relationship in Figure 5.8 (a) is read as: “special class is a kind of general class.” The special class inherits the variables and methods of the general class, which is implied by the Generalization-Specification relationship.
2. Hole-Part Relationship

Another type of relationship that appears often is the Whole-Part relationship. The relationship declares that an object, called a super object, is composed of other objects called sub-objects. Figure 5.9 shows an example of a hole-part relationship. Figure 5.9b is read as “The block is composed of a flat surface, a hole feature and a slot feature.” A solid-filled triangle is used to represent the whole-part relationship.

3. User-defined Relationship

In the CAMP, the user-defined relationships reflect the pre-defined information relationships, which may come from the BOP or general manufacturing knowledge. For example, a hole feature has 5 alternative
manufacturing processes. The feature is defined as an object, and a process is defined as another object, as shown in Figure 5.10.

![Figure 5.10 User-Defined Relationships](image)

By using of generalization–specification, whole-part and user-defined relationship, the system’s ORM can be setup as shown in Figure 5.11, so that information can be classified into objects. For example, in order to describe the part shown in Figure 5.9, a part object is created to represent the design information of the part which is composed of a flat surface object, a hole object and a slot object. Each feature is associated with specific processes. At the same time, this part is a non-rotational part. Thus, a non-rotational part object is associated with the part object so that the part object can have all the characteristics of non-rotational part.

![Figure 5.11 An ORM diagram for the Block and its Features](image)

### 5.9.2 Object-Behaviour Model (OBM)

The objective of a behavior model is to describe the way that each object in a system interacts, functions, responds or performs. A behavior model for an object is similar to a job description for an object. In this research, state nets are used to represent OBM. A basic concept of behavior modeling is the set
of states that an object exhibits in a system. In OSA, a state represents an object’s status, phase, situation or activity. Figure 5.12 shows some states of process objects. The procedure of changing the state of an object is called

![State Net for a Process Object](image)

Figure 5.12 A State Net for a Process Object

transition. The events and conditions that activate state transitions are called triggers. The activity that an object performs is called action. A state net is a configuration of symbols representing states and state transitions for an object. In state net, rounded rectangles represent states. Rectangles that are divided into two sections represent transitions. The top section contains a trigger description. The bottom section contains the actions. For example, Figure 5.12(a) shows the components that construct a state net. Figure 5.12(b) shows an example of the activities to define a process object. There are three states of a process object: process undefined, process underdefined and process defined. The first step of a process object is to select a cutter that is used in this process. The process is incomplete when it only has cutter information and the state is under defined. The second step is the definition of toolpath. After this step, the definition of a process is finished.
5.9.3 Object-Interaction Model (OIM)

The ORMs describe the static relationships among objects. The OBM describes the behavior of an object, but in isolation from other objects. An OIM model is used to describe the interaction, such as information associations among objects. One object interacts with another in many different ways. For example: an object may send information to another object; an object may request information from another object; an object may alter another object; and an object may cause another object to do some actions. Figure 5.13 shows an example of the interaction between a part object and its process object, which were defined in Figure 5.10 and Figure 5.12.

![Diagram](image)

**Figure 5.13 OIM Between a Part Object and One of the Process Objects**

The part object is composed of a flat surface, a hole feature and a slot feature. Each feature has its own parameters. When choosing the process to machine the flat surface, the cutter and the toolpath in the process are determined by the feature’s parameters. The two zigzag arrows in Figure 5.13 show the parameter-driven interaction between the flat surface and the
process. When the dimensions of the flat surface are changed, the process to machine the flat surface may change accordingly. The zigzag arrow indicates that programming work is needed to implement this interaction activity.

5.10 Systematic Information Modeling Used

When using OSA approach to model a complex system in the research, high-level abstraction of objects is applied to reduce complexity and make the information models easy to create, maintain, and display. A high-level object groups relative objects and the relationship among the objects into a single object [40]. The top-down approach is used to expand a high-level object into low-level objects and relationships. Figure 5.14 shows the hierarchic structure of system information models.

![Systematic Information Modeling Hierarchies](image)

Figure 5.14 Systematic Information Modeling Hierarchies

The building of the information models is split into three levels:
1. The definition of a system model, which contains domains that are subdivided into subsystems. The system model may be derived from analysis of the system’s high-level object interaction models.

2. The definition of an information model, which contains objects that are subdivided into states.

3. The definition of the state model, which describes the behaviour of objects.

5.11 UML Class Diagram for the Information Modeling

Among competing object-oriented analysis and design tools, the authors have selected the Unified Modeling Language (UML). The UML is the standard universal language is approved by the object management group for representing (i.e., specifying, building, and documenting) every kind of software system. The UML defines a metamodel-based graphical notation for O-O analysis and design. This is embracing all the features of the O-O paradigm such as reusability, representational versatility, inheritance property and rapid prototyping [92]. In other words, the UML is a formal semantic metamodel defining basic modeling concepts (object, class, association, etc.) and it also is a graphical notation for system representations including eight different diagram types. It embraces all the features of the O-O paradigm through a synergistic combination of various proven techniques including encapsulation, inheritance, polymorphism, and generalization [58].

The UML is not a programming language but a design language. It is a modeling language that specifies semantics and notation but it is not a methodology and does not define a software design procedure. In building the model, the UML may be used to:

- display the requirements of a system and its major functions using use-case diagrams and actors;
- illustrate use-case realizations with interaction diagrams (sequence
and collaboration diagrams);

- represent a static structure of a system using class diagrams;
- model the behavior of objects with state transition and activity diagrams;
- reveal the physical implementation architecture with component and deployment diagrams.

UML with its use-case view diagrams, logical view diagrams and component view diagrams develops three orthogonal models: functional; object; and dynamic models [93]. The functional model represents the transformational aspects of the system and it captures what the system does without specifying how it works. The object model represents the static structural aspects of the system, in which objects, their entities, their attributes, their operations and their relationships with other objects are described in detail. The dynamic model represents the temporal behavioral aspects of the system, in which any change is described by activities and events, sequences of events, and states [69].

UML have been used as a tool for developing manufacturing system control software. By means of few diagrams, proposal is made on a systematic design methodology to make the task of developing the control software system easier.

The class is an information entity, which has one or more attributes, zero or more methods, constraints and relationships. Entity is something in the real world that you wish to describe or track. Attributes define what the classes attribute consists of and have identification name, data and definition. Methods define what the functions class performs. Constraints limit the behaviour of class. Relationships relate the class with others in the model. A relationship includes inheritance, composition, aggregation, and recursion. An object is one instance of a class with specific data. A class diagram is a visual model of the classes and associations in an organisation [63].
Multiplicity is shown as a number for the minimum value, ellipses and the maximum value. An asterisk (*) represents an unknown quantity of many.

Many times a relationship requires both of the entities to exist. For example, what happens if you have a manufacturing activity, which incurs a cost and time, but there is no data on the file for that? There is a referential relationship between the manufacturing activity and the cost and time. Business rules require cost and time data, which must be available to make use of it in manufacturing activity. This relationship can be denoted by specifying the minimum value of the relationship (0 if it is optional, 1 if it is required).

5.12 Conclusion

Task clarification of the IPPPIS has been done in this chapter, with the details on the method used for development of the proposed CAPP in the IPPPIS. Process planning approach in the model was given with explanation of planning methodology, architecture of the system and information system. How the process planning was carried out is outlined with the different steps involved in the selection, such as raw material, machining features, assignment of processes and operations, assignment of machines, machine sequencing, assignment of fixtures and assignment of tools. Functional structure of integrated system developed has been explained in detail, with various components its structure. (i.e., determine raw material, determine options for operations, processes and machines, select and sequence machines and fixtures, determine tools and work conditions, write results, analyse and describe machining features, select possible processes and operations, compatibilize operation alternatives, select capable machines and cost estimation.)

Subsequent sections in this chapter explained the information modeling methods and approaches with the information content on the IPPPIS developed in this research. Different approaches and models such as OSA,
ORM, OBM and OIM were explained in detail, along with the application of UML class diagrams for the information modeling. Next chapter gives details on the how modeling of domain specific knowledge in a machine tool factory was done using the modeling methods and tools explained in this chapter.