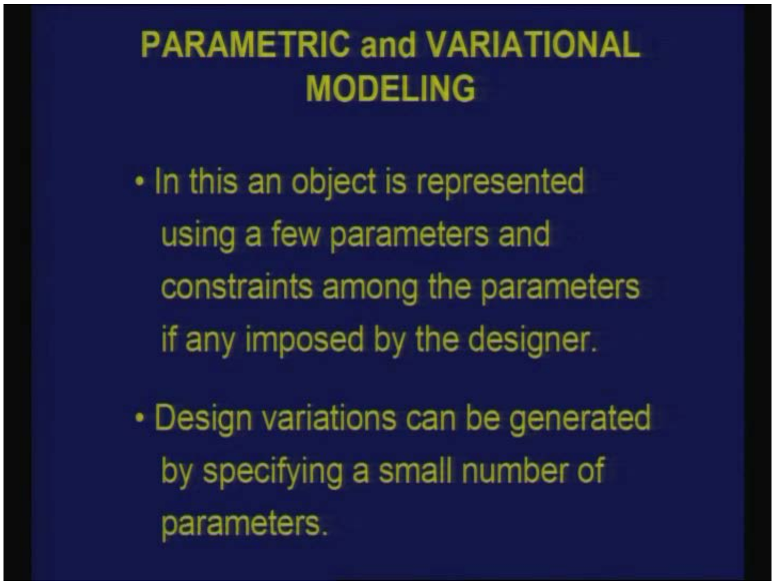
CAD / CAM
Prof. Dr. P. V. Madhusudhan Rao
Department of Mechanical Engineering
Indian Institute of Technology, Delhi
Lecture No. # 11
Geometric & Product Data Exchange

Let’s get started. So far we have been discussing the subject of curve, surfaces and solids. I think we spent many classes discussing about these subjects. Now the next phase which we are going to discuss, the next few topics basically are related to how these modeling techniques are used for various manufacturing applications like geometric modeling and then applications of geometric modeling to manufacturing consists of a major portion. Now before we do that, before we take up a let’s say applications like assembly or inspection or manufacturing and other applications, it would be appropriate to know like what we mean by geometric and product data exchange because representation of geometry is one aspect and how it is exchanged among various CAD systems is another aspect because more often one has to work with more one particular software. So in those situations there are many issues which are related to geometric representation and data exchange which one should know before we take up the applications. So today’s lecture will basically look at this aspect. But before I take up geometric and product data exchange, I just wanted to basically first cover what we call as parametric and variational modeling.

(Refer Slide Time: 00:03:06 min)



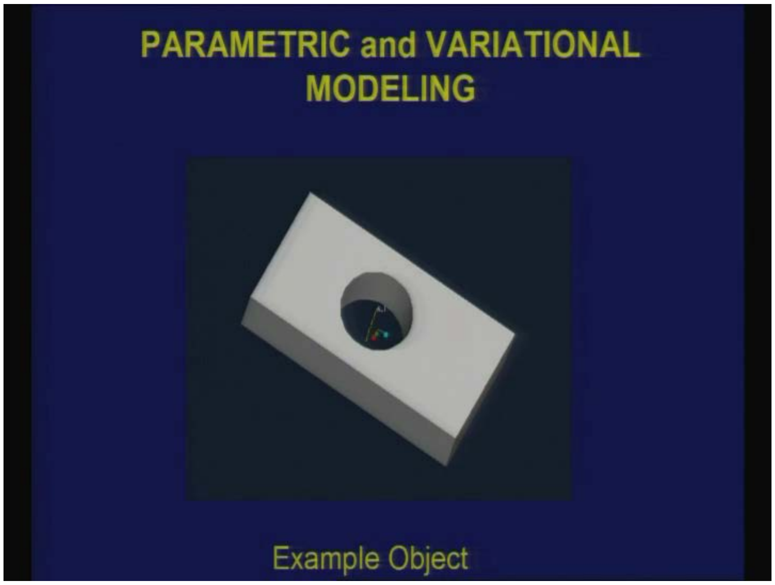
PARAMETRIC and VARIATIONAL MODELING

- In this an object is represented using a few parameters and constraints among the parameters if any imposed by the designer.
- Design variations can be generated by specifying a small number of parameters.

Parametric modeling and variational modeling are the two terms which are very often discussed and used with reference to CAD CAM applications. So it’s appropriate to know what parametric and variational modeling is and then we will know, how like why this aspect is important when we do the data exchange and other aspects. So what is done in both parametric and variational modeling is an object is basically represented using a few parameters and constraints among the parameters if any imposed by the designer. If you recall when we were discussing the subject of solid modeling, we looked at one aspect of solid modeling called primitive instancing. Primitive instancing basically refers to, basically representing a class of objects using a certain fixed parameters. So parametric and variational modeling, you can say is an extension of primitive instancing but the only thing which you have here is that if there are n parameters which are we used to represent a object uniquely then you can also, you are also allowed to impose certain constraints among these parameters. Sometimes this is also called as a constraint based modeling. Actually this is more close, parametric and variational modeling is more closer to what is usually is the designer's requirement.

So I will come back to that and second is once I represent the parameters as well as the constraints among the parameters, I can generate the design variations by specifying only a small set of parameters like imagine a case of a primitive instancing like an object can be uniquely defined if I provide let's say 6 or 7 values to represent the solid. Now if you have lot of constraints and relations among the parameters then you need not input all the 6 or 7 parameters. May be you may have to do, you may have to input parameters which are much less than the total number of parameters that the values which you have to provide. So we look at this aspect, so if I basically take this approach, it's usually called as a parametric and variational. Now there is difference between parametric and variational, the two approaches. We will also look at what is the difference between that.

(Refer Slide Time: 00:05:50 min)

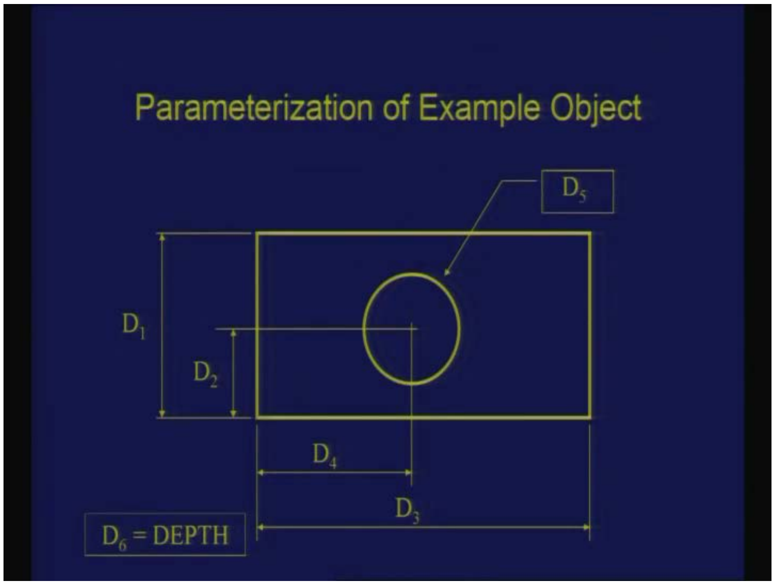


PARAMETRIC and VARIATIONAL MODELING

Example Object

Now suppose if I have an object like this then this is one of the objects which we earlier also dealt with reference to primitive instancing. Here is an example object and we also know that if I want to represent let's say this topology what are the number of parameters which are required. So how many values are required in order to uniquely represent this object, if you can recall our previous lecture. Six. So we discuss that the number of parameters are 6, 3 refers to the dimensions of the cuboid I can call this as basically the length then this is the width and this is the depth portion. So length, width and depth are the three parameters and then it also has a through hole so through hole has certain diameter, diameter becomes the fourth value and position of this particular hole with reference to the part like where exactly it is located on a given face requires inputting another two values. So, 6 values can be used to represent not this object, a class of objects which is cuboidal in shape but also has a through hole in it on one of its faces.

(Refer Slide Time: 00:07:11 min)

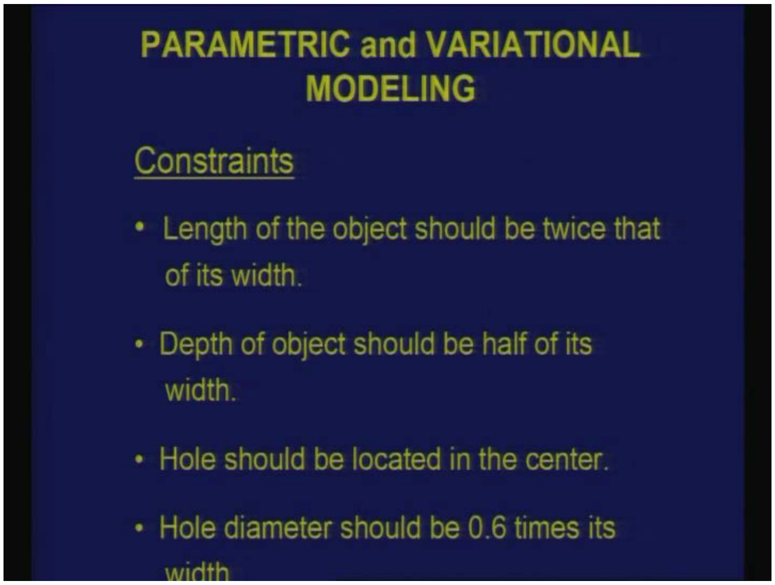


Parameterization of Example Object

D_6 = DEPTH

Now if I look at this, the representation, this if you recall the parameterization of this example object which we did is the length is represented here as D_3, width is represented as D_1 then depth is represented as D_6 because you seeing only one view of this. Then diameter is represented as D_5 and I have the position of this particular through hole on the part is represented by two other parameters which are like D_2 and D_4. Now if I give a set of these values D_1 to D_6, I am able to generate this object. By varying the D_1 to D_6 I am getting the class of objects.

(Refer Slide Time: 00:08:06 min)



PARAMETRIC and VARIATIONAL MODELING

Constraints

- Length of the object should be twice that of its width.
- Depth of object should be half of its width.
- Hole should be located in the center.
- Hole diameter should be 0.6 times its width

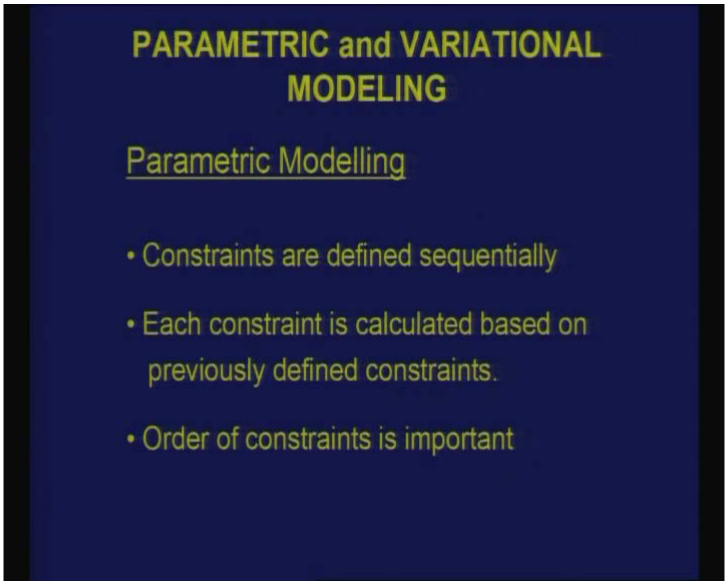
Now let's look at like, let's extend this to what we mean by parametric and variational modeling. Now in parametric and variational modeling what is done is I have those 6 parameters which are used to define the object uniquely. Designer may impose certain constraints like if you, suppose if you are in a design situation and if I have to arrive at a final geometry, usually you don't give all the final dimensions of a geometry. During a process of design you say that one is like you can say that for example a constraint like the length of object should be twice that of widths. This has nothing to do with what is the exact value which you are going to arrive but this is like one of the constraints which designer imposes. In fact taking of this aspect of design is usually called as a design intent. Designer has certain intentions in imposing these constraints. So you are not directly taking the intentions but you are able to put them in the form of geometric relations. So this is basically if I do that, these are the constraints which can be imposed in the form of a parameter. So designer for some reason decides that this object which is to be modeled should have a length which is twice that of a width, so this may be one of the intentions.

There may be another intention that depth of the object should be half of its width, this is a second intention which comes because of certain functional aspects of the component. Then hole should be located in the center then you say hole diameter should be 0.6 times its width. This is basically to take care of that the hole is not close to let's say a width of this, in that case you have a problem you have a wall thickness which are critically or which are critical or which are very small in nature. So there is certain intention behind imposing these constraints. Now take care of those 6 parameters which are used to define the object together with designer's intention, the number of parameters which effectively reduces to a smaller number. You don't have to provide all the 6 values in order to define the object because these constraints will basically reduce the number of inputs which need to be supplied. So, if I take care of these constraints and when the designer comes up with these constraints, it is usually based on certain functional aspects.

Many a times the constraints could be conflicting too like you have a three situations whenever you are in a design process. For example I may impose these constraints which are more than required like I may have a conflicting constraint that means there are two constraints which are conflicting and both cannot be satisfied. So you say that this is an over constraint object, so you have to release one of the constraints in order to define the object or you may have a under constraint. In a under constraint, you do not have enough constraint to define the object uniquely. There can still be a multiple operations or I can have an object which is rightly constraint where these constraints and the input values are given such that there can be only a single shape and size of the object which can be defined with the given input.

So you have all the three situations and this aspect you have also dealt when you are working with software like you have an option like regeneration. What is regeneration? Regeneration is nothing but you are trying see whether the constraints which you have imposed or rightly so that you can generate a unique object. Sometimes when you impose more constraints, it doesn't regenerate it says you relax of the constraints or when you do not impose enough, you do not give enough input values then the software request you to give more input so that it is rightly constrained. So the regeneration which is usually done has direct relation with what we call as parametric and variational modeling. Now these are the designer's intentions then I have the 6 parameters. Now how do I go about solving this?

(Refer Slide Time: 00:12:35 min)



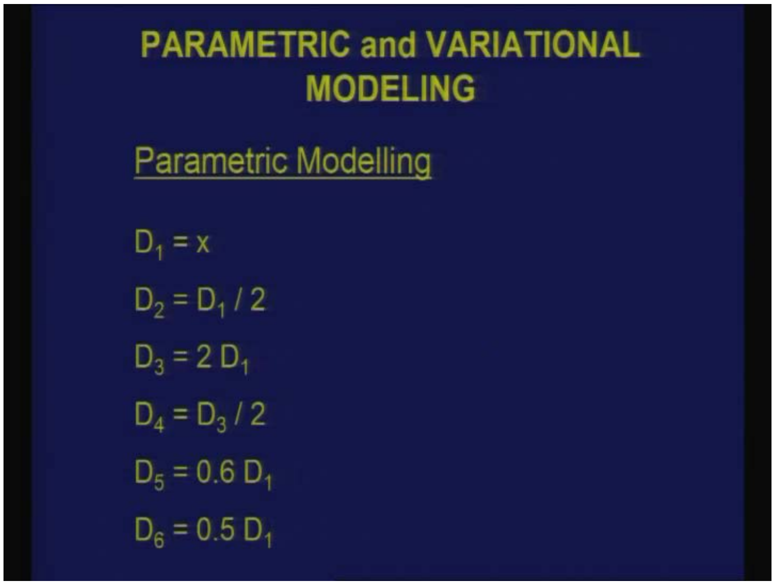
PARAMETRIC and VARIATIONAL MODELING

Parametric Modelling

- Constraints are defined sequentially
- Each constraint is calculated based on previously defined constraints.
- Order of constraints is important

First is let's take up what we call as a parametric modeling. In parametric modeling what are done is whatever are the designer's intention which can be written in the form of constraints are sequentially defined. You not only define all the constraints like what we have seen earlier, you say that these are the 4, these can be mathematically be represented in terms of parametric values. You not only do that but you also prioritize. The idea is if I have to relax let's say one of the constraints which one I am going to relax depends on what is your priority. So, the one which has the highest priority, one which has the highest priority has to be respected. One which has a relatively low priority has to be basically something which can be relaxed if necessary, so that aspect can be taken. So coming back to this constraints are defined sequentially then each constraint is calculated based on the previously defined constraints, I will come back to that and order of constraints is important so that this helps you in relax.

(Refer Slide Time: 00:13:42 min)



PARAMETRIC and VARIATIONAL MODELING

Parametric Modelling

$D_1 = x$

$D_2 = D_1 / 2$

$D_3 = 2\ D_1$

$D_4 = D_3 / 2$

$D_5 = 0.6\ D_1$

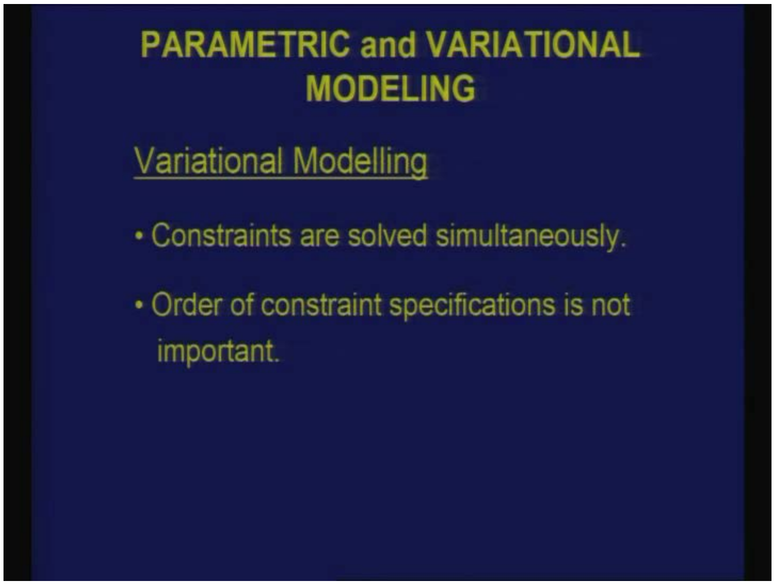
$D_6 = 0.5\ D_1$

Now if I put the designer's intention in terms of the 6 parameters which we looked say that D_1 is equal to x, x is input which somebody has to provide, this is nothing but the width of the object. And you are saying that D_2 that is one of the dimensions which define the center of a hole should be half of the width because you are saying that center should be located at this, so you say that D_2 is D_1 by 2. then I have a D_3 which is the length of the object is twice that of a width which is another constraints, so I can say that D_3 is nothing but twice that of D_1.

And I can say D_4 which is the second aspect that is a second parameter which defines the center of the hole is half of its length, so you say D_4 is equal to D_3 by 2. Then we also define that what should be like the depth of this particular thing which should be the last constraints defines that the depth of the object is half of the width and another constraints is hole diameter should be 0.6 times that of the value which is defined. What is the value, which is D_1 which is the width of this point. So these are the four constraints which are written in the form of an equation

Now I can solve for these equations and I know that if I provide one value just one input that is x, I can solve for all other values. So what has happened is together with designer intentions or constraints, a parameter which were 6 has now reduced to just one. So given a particular value of x which is nothing but the width of the object, the object is fixed. It's both shape and size has been fixed. If I take care of these constraints also as a part of my definition, it's usually called as a parametric definition. And now solution which is done is sequentially like I will start from 1 2 3, they are also prioritized and I do that.

(Refer Slide Time: 00:15:58 min)



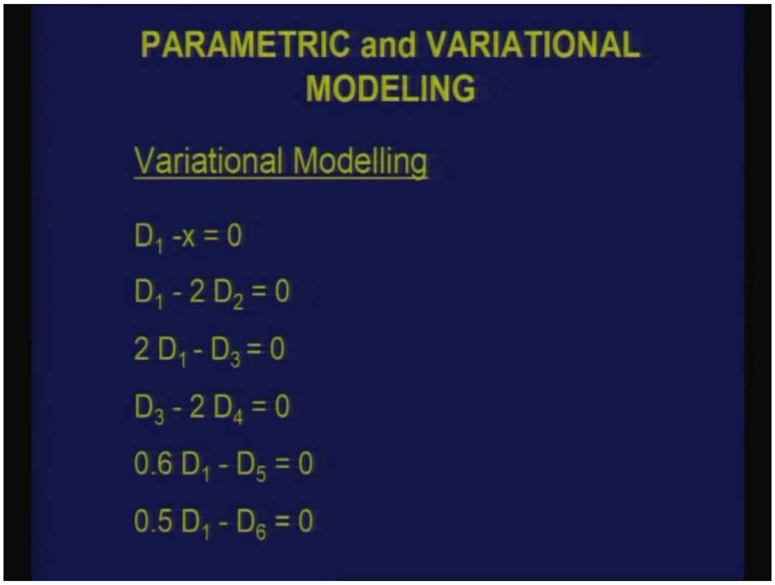
PARAMETRIC and VARIATIONAL MODELING

Variational Modelling

- Constraints are solved simultaneously.
- Order of constraint specifications is not important.

In variational modeling it is slightly different. What is done is you do not prioritize the constraints. You put down all the constraints in terms of the equation and solve all the equations simultaneously either solution may exit or it may not exit. If it is rightly constraint, I will get a solution, if it is not rightly constraint if it is over or under constraint then I may not be able to solve it. And here the order is not important like before designer’s intentions need not be prioritized they can be given in any form. So I can again write down the equations.

(Refer Slide Time: 00:16 36 min)



PARAMETRIC and VARIATIONAL MODELING

Variational Modelling

$D_1 - x = 0$

$D_1 - 2 D_2 = 0$

$2 D_1 - D_3 = 0$

$D_3 - 2 D_4 = 0$

$0.6 D_1 - D_5 = 0$

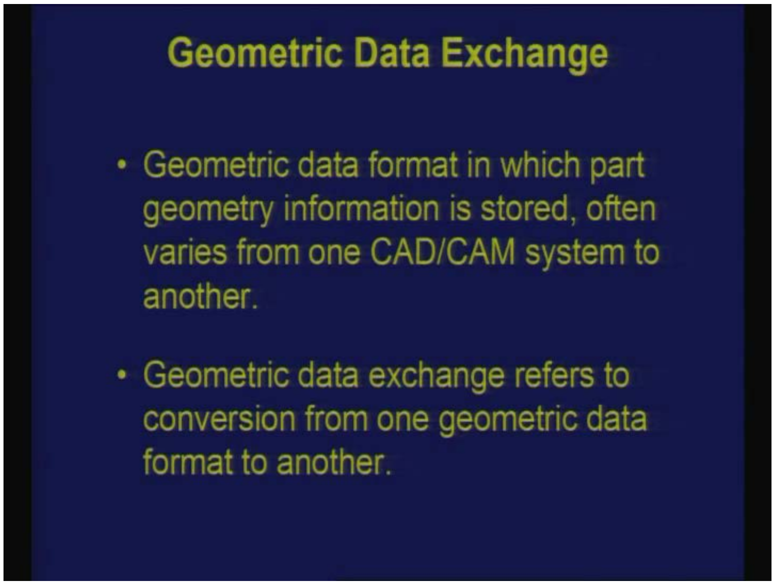
$0.5 D_1 - D_6 = 0$

Same thing you are writing as a equations of this and I can solve for these equation in terms of x and I have again given a value of x, I can uniquely solve this. Now the example which we have taken here is very very simple both the parametric and variational model of this particular object in terms of equations is this. Usually you may have these equations which may have certain trigonometric relations, you may have square roots, you may have polynomials. So a solution sometimes is not possible to, that means it’s not possible to solve these equations analytically. You may even go and solve it numerically whereas in the case of parametric modeling, since you are solving sequentially as like earlier relation leading to a new relation, so usually the solution is much simpler that means solution can be very easily obtained. But in the case of variational modeling which has an advantage that you don’t have to prioritize the constraints but you may have to go for equations like numerical methods, etc.

Sometimes numerical methods are used in parametric too but more often they are used in variant variational modeling. and this aspect which is now manifest in many of the CAD packages like you may have seen like what you basically do is a variational modeling like when you practiced building of the parts is you went through a variational modeling. Constrains can be directly imposed in terms of the values also and the constraints which we looked at need not be purely in terms of dimensions, you can also have geometric constraints. I can say that here is an object, one face of this particular object should always be perpendiculars to another object or the angle between two faces should always be 60 degrees. So you can also impose geometric constraints like I can say that two faces should be parallel, perpendicular, angular relation so some of those relations can also be imposed.

So you have two types of relations. One you call as a dimensional constraints and second is geometric constraints together with that because more often the designer is more comfortable dealing with their intentions rather than directly arriving at the dimensions solving themselves. So designer need not solve for final geometry, solution to final geometry is given by the software. Designer only gives what are the intentions which should be satisfied as a part of a design. So this is you can say a powerful aspect of recent CAD CAM systems where it supports both parametric and variational modeling. I just brought this because when we are discussing about data exchange, we may refer to parametric and variational models at later stage.

(Refer Slide Time: 00:19:58 min)

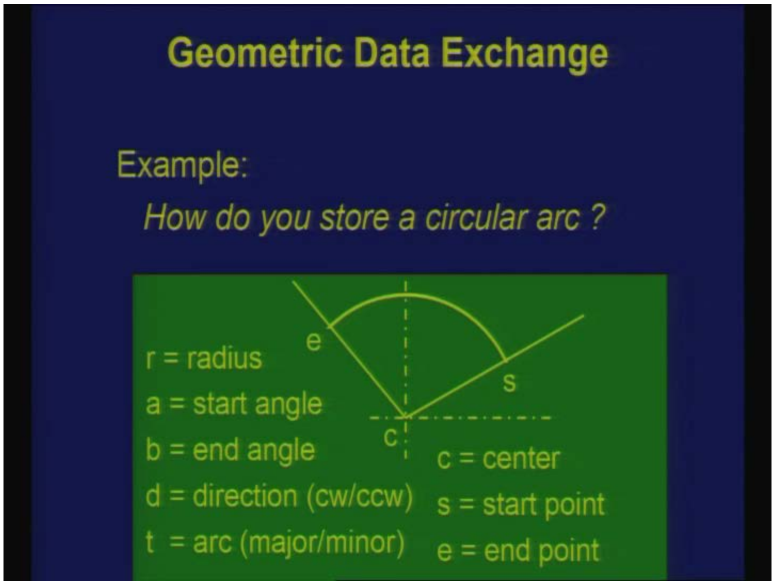


Geometric Data Exchange

- Geometric data format in which part geometry information is stored, often varies from one CAD/CAM system to another.
- Geometric data exchange refers to conversion from one geometric data format to another.

So after this let's come back to the actual subject of a today's lecture. We are basically interested to discuss about geometric data exchange which refers to how the geometry is basically exchanged between two CAD systems or let's say any of the two software packages where the geometry has to be transferred from one system to another system. And this is necessary because the format in which a geometry can be stored need not be a unique one. Since you have a multiple options or multiple ways in which a given geometry can be stored, so at some stage or the other there is a need for converting it from one format to another or converting from one system to another is required. So hence the subject becomes important.

(Refer Slide Time: 00:20:52 min)



Geometric Data Exchange

Example:

How do you store a circular arc ?

r = radius
a = start angle
b = end angle
d = direction (cw/ccw)
t = arc (major/minor)

c = center
s = start point
e = end point

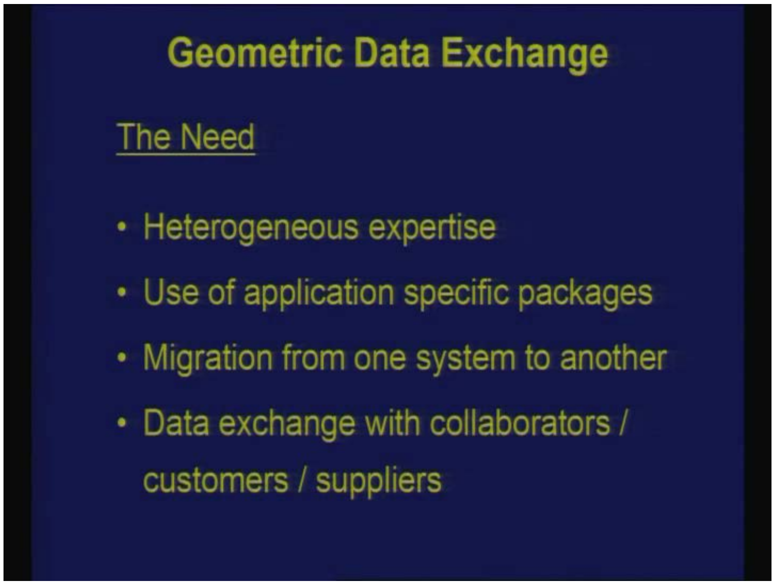
If I just take an example like think I referred to these example earlier also is suppose if I want to store a circular arc. Here is a segment of an arc, s indicates starting point of the arc, e indicates the end point of the arc, c is the center point of the arc. Now when I store this particular arc let's say in my native format or in my package or as a part of a software, what are the values which should be stored like you are trying to come back. Given a particular object what parameters isn't it, what parameter should selected in order to represent an object is the case. So somebody may say these are the possible parameters. there is only a one set, radius of the arc is important, a is a start angle like if this is my x axis, the angle which the starting point basically makes with x axis is of important, I can call it as the start angle. then the angle which is subtended with the end point is a end angle, direction whether the arc is a clockwise or counter clockwise is important or whether the arc is a major arc or a minor arc because with the same starting point and same end point and same center, another arc is possible which is a major arc then center point of this arc, start point and end point.

Is it necessary to store all these data? Not necessary, because there is a lot of redundancy in terms of information storage. If I am able to store a few parameters, other parameters can be calculated. So when somebody chooses a format to let's say represent an arc, it need not be true that everybody selects the same set of parameters. Somebody may be more comfortable with selecting start angle, end angle. Others may prefer to choose let's say start point or end point. Somebody may select center, others may select radius. There can be a many ways in which the parameters can be selected from a given set in order to represent the object uniquely.

If this is the case and if there are two packages which are storing in two different formats, when I actually the data is for example translated from one format to another format, there has to be a conversion which has to be done because in one case you are actually storing a start angle and an end angle. There is another package where the start point and end point is stored. So if I just dump this data there, it cannot interpret because it doesn't make any sense in the second package. so you need a translator which reads the information the way it is stored in package one and converts into a format which can be interpreted by the other package. And then saves information in the other packages format.

Second important thing is even if suppose there are two systems, CAD systems they select the same set of parameters. You may actually represent these values in a different order. Somebody may first store the center point and next the radius, there may be other system which may store first the radius and next the center point. So if there is order in which the parameters also vary, even they have to reordered as a part of this translation process, when you do a data exchange. So all these things are part of you can say geometric data exchange, when I have to convert a part which has been built in one of the packages has to be read or has to be displaced in another package, I have to do this conversion which is necessary. Now this is a, example which is taken here is a very very simple. And once you go for more complex objects, the number of parameters could be a few tens or even few hundreds. If I take a very complex part, in order to represents its geometry uniquely, I may have to have many parameters which are possible. So in those situations the effort becomes much more in terms of conversion or necessity for conversion becomes more evident.

(Refer Slide Time: 00:25:22 min)



Geometric Data Exchange

The Need

- Heterogeneous expertise
- Use of application specific packages
- Migration from one system to another
- Data exchange with collaborators / customers / suppliers

Now what is the need for let's say a geometric data exchange like why this subject is important, why its study. There are a few important points which are put here. First is you have a heterogeneous expertise in the industry like industry may recruit let's say people who are working in CAD CAM applications from different places and they have different specializations. Somebody is probably familiar with having worked with a package one then there is another this thing. Another person who has previous experience of working in another package, so the expertise itself is heterogeneous. And if the industry is using all these packages, then this heterogeneity remains and at some stage when I have to transfer data from one system to another, I have to go for a conversion which is there.

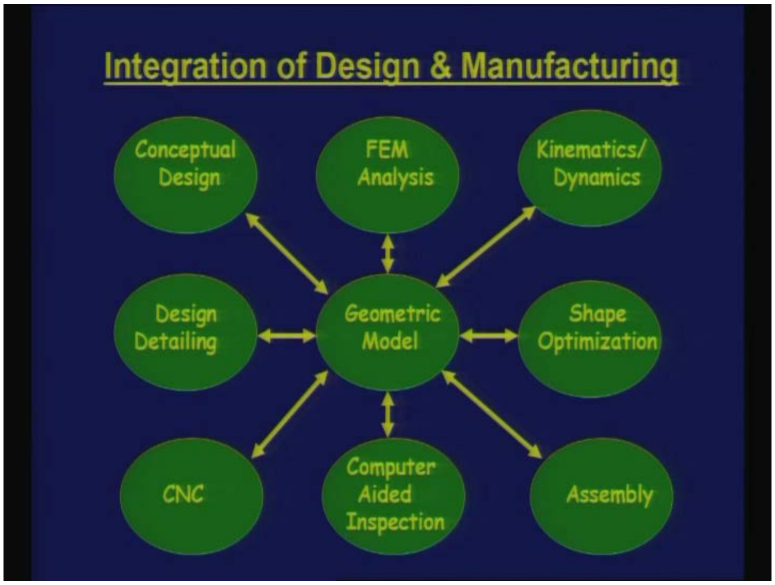
And use of application specific packages. Now in order to avoid let's say a geometric data exchange aspect, one of the things which is done in CAD CAM software is there is a single piece of software or you can say a major CAD CAM system, you see all the applications which are needed for design and manufacturing. You call as a CAD CAM applications, whether it's a manufacturing, analysis, modeling or assembly everything is done under one package. So you have many high end systems like pro-engineer or KTR unigraphics, so you can call them as a high end systems. In these situations you probably may not really require the data exchange because it is developed by the same group of people. So whatever is the storage geometry, the parameters and the order in which I am storing a geometry in one module of the package, same thing will be followed in another module. So there is no necessity for conversion from one format to another. In spite of this like it is seen in many big and medium industries is that a single package will not satisfy all the requirement, though I may standardize all my design and manufacturing applications based on a single CAD CAM system, single CAD CAM software but still for certain reasons you may have to go for a very specific engineering software like the package which you are chosen is weak in certain applications or it cannot do certain things which are the industry requirements.

So in those situations you may use apart from standardizing your efforts in one of the high end CAD CAM systems, I may still use applications specific package like I may use a specified finite element package or I may use a specific package for let's say manufacturing or CAM applications for generation of NC code etc. So in those situations till the data transfer are taking a model which has been built in one of the high end CAD CAM systems to the application software becomes necessity. So in those situations you need the geometric data exchange. So hence we are learning this aspect. And third important aspect is migration from one system to another like an industry which has been doing many of its design and manufacturing task using a specific package realizes that over a period of time, there are better systems, better CAD CAM systems in the market which can offer a better features and more facilities than the one which I am using. So industry decides to migrate to a new system, so all the old design and manufacturing data or the geometric data which is there has to be now transferred to a new system or the migration sometimes becomes necessity when you have like within the CAD CAM packages, mergers and acquisitions because one of the company is closed or it has been bought by another company, so you do not have any support for the package which you are using over a long period of time, so you prefer to migrate to a new system.

All your geometric data which was there in the previous system has to be now translated into a new system. So in those situations geometric data exchange becomes important. And this is very important, the final point is when you are actually working with the external world like you have collaborators who use a different package like there is a technology transfer which is happening from one company to another company. The people who are the beneficiaries of the technology and the people who are the developers, they use different CAD CAM systems. So this needs to be converted or nowadays you have like what you call as an extended enterprises. In extended enterprises, you try to involve both customers as well as suppliers. So when you have your enterprise which is not restricted your own organization but goes on the customer and supplier side, you have lot of information which needs to be exchange particularly in environment like product data management etc.

So in those situations you require the geometric data exchange like a very typical example is like an air craft manufacturer like you have two commercial air craft manufacturers and you have three major manufacturers for air craft engines. So an air craft manufacturer may have option of buying engine from any of the three engine manufacturers. These three engine manufacturers use different kind of data and data formats and solid modeling package but when I have to actually communicate or collaborate or let's say I purchase then the data exchange becomes extremely important. So this is a very typical and practical case of what is happening in the case of this. This is true with many major other industries whether it's automobile or a consumer goods industry. So the necessity of geometric data exchange is very well taken. So let's look at what are the options which are available when somebody has to do this kind of task.

(Refer Slide Time: 00:32:00 min)



Integration of Design & Manufacturing

Conceptual Design

FEM Analysis

Kinematics/ Dynamics

Design Detailing

Geometric Model

Shape Optimization

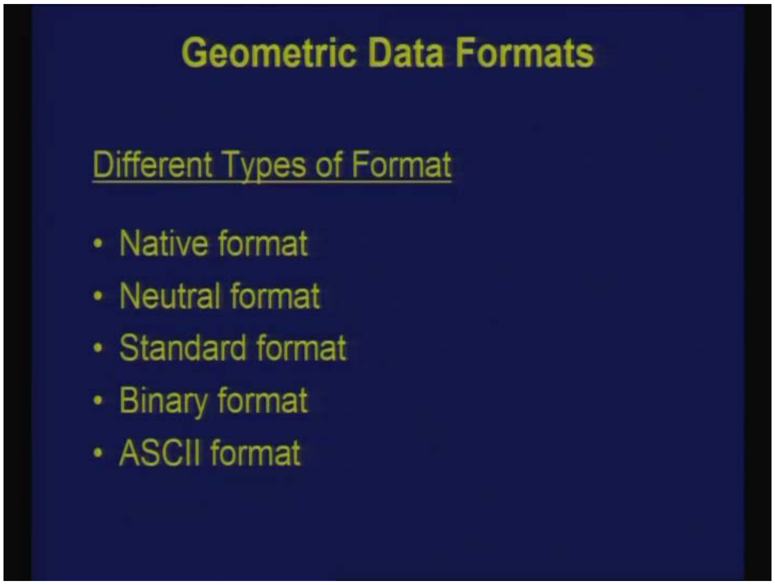
CNC

Computer Aided Inspection

Assembly

This is one of the slides which we have seen is when you have applications specific package or when you have a integrated environment, when you have to move from one package to another, the geometric model become central and the data exchange becomes a necessity.

(Refer Slide Time: 00:32:17 min)



Geometric Data Formats

Different Types of Format

- Native format
- Neutral format
- Standard format
- Binary format
- ASCII format

Now whenever you say geometric data, there are certain terms which are always you come across, these are called as a geometric data formats. One is which you call as a native format. Native format is a format, suppose if I am using let's say a specific software, CAD CAM software there is a format which has its own format which you call as a custom format or a native format in which the parts which are modeled are represented like whether I use this specific software, when I create a model and store it as a file, it gives an extension and it stores the data in certain format. So that is a native to that; that is not a standard format. That is purely native and this file can be read only by that package, it cannot be read in other software except when we have a translation facility exchange. Second is a neutral format. What is done in a neutral format is the packages give you an option like to put the geometric information in the form of a file and which can be very easily interpreted by outside world. They also give like the complete format details and how the information is stored, what is the order in this information is stored. So if I have a manual for this particular neutral data format, I can always interpret what this is and I can use it. Neutral formats are extensively used to build applications over an exciting package like for example let's say I have a CAD CAM system which gives which has a native format and the neutral format. Now I am let's say trying to build a new application which is a CAD CAM like suppose you take up let's say a new application for which you are building like somebody may be working on inspection others may be working on nesting.

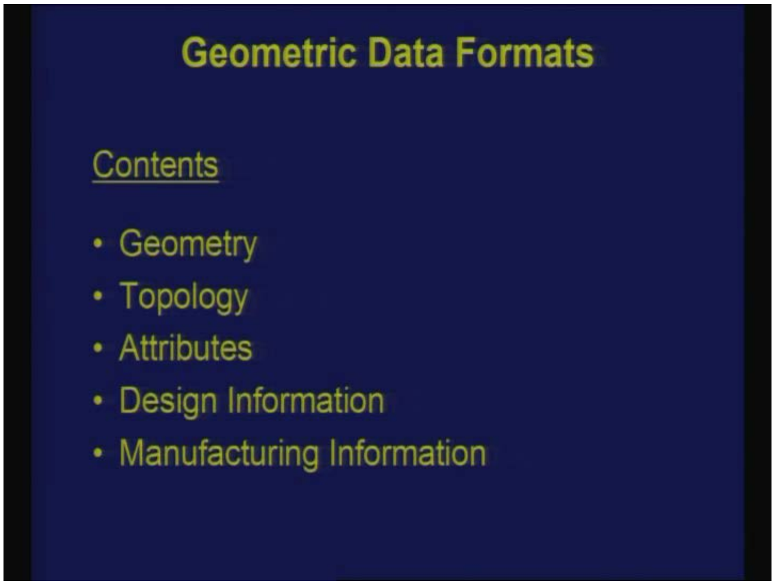
Now whenever I develop let's say a new algorithm or let's say new implementation, I would like to make use of the existing system as a front end, so that it can be displayed in the package which I am trying to use. So what you do is if you know what its neutral format is, I can always right output of my development in a format which can read by the package and I can display the same like I can use the display options and all the user interaction options which are available in the package for my output because I know what its neutral format is. One may not be able to dump it in the form of a native format because usually it is not open and it's often binary.

So you cannot really interpret what somebody's native format is but neutral format is something which is given, so that it's thrown open to the world so that one can make use of this for application development or even for data exchange. Then you can also have a standard format where everybody agrees that let's there be some international or national standard and if this exits most of the companies or most of the organizations which are working in a nation or let's say in a global world, if they follow all of them this standard format there is no problem like if I say that here is a geometry which is represented in an international standard format, whatever may be the software or nation or wherever it is, so it can always has an interrupter to read the standard format and I can interrupt what the geometry is.

So standard format is another this thing then the data can be represented as binary or ASCII. There are advantages for both. Binary formats are preferred as native formats the native, the custom formats which packages you use often they prefer binary format because in a binary this thing input and output option becomes much faster like when I write the geometric data to a file or when I read from a file and display it, so the input output option becomes much convenient, it becomes much faster.

If it is like a neutral format then you have to do a lot of interpretation within the package in order to display that whereas ASCII formats are the other where you can read a file, you can interrupt what are the characters and the phrases and others in order to know what the geometry is. And often the neutral and standard formats are ASCII formats that's a more common option. It is not necessarily true but standard formats can also be binary but more often neutral and standard formats give you an ASCII option for the data exchange. So when we go along some these formats you will see that both these formats are being used. Now what do, let's say when you say that a geometry is stored in the form of a file, what would be its contents.

(Refer Slide Time: 00:37:41 min)



Geometric Data Formats

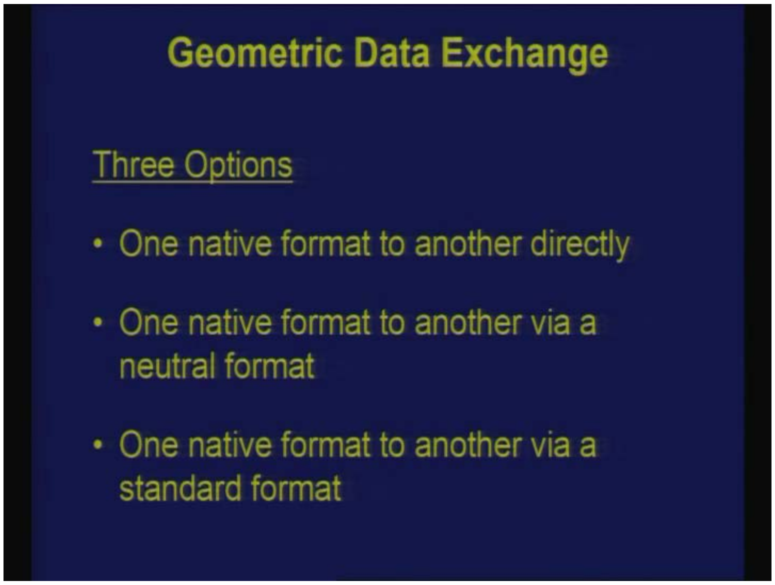
Contents

- Geometry
- Topology
- Attributes
- Design Information
- Manufacturing Information

It will have a geometry, it will also store topology like we have discussed that geometry and topology are different in our earlier lectures. Then it may store attributes like somebody may have made an assembly where you are trying to represent parts in different colors and I would also like to store the color of that particular part which has nothing to do with geometry, it has nothing to do with topology but it is an attribute which is attached to that or I would like to store let's say this particular object is made of certain material, a specific material.

So material becomes an attribute, so attributes can be stored. Then other geometric information sometimes which are related to design and manufacturing can also be stored like in some cases you can also store tolerance information. There are situations where you can store surface roughness information, you are not only storing an object, you say that this object has many surfaces and the surface roughness which is may be desired at end of a manufacturing process can also be associated with data. So depending on which system you are using, you may have one or more of these things as a content of your file which is being stored.

(Refer Slide Time: 00:39:11 min)



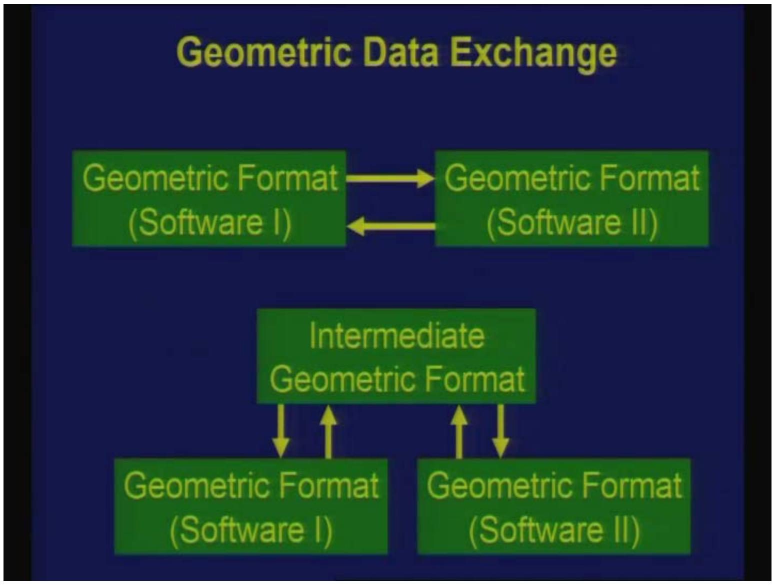
Geometric Data Exchange

Three Options

- One native format to another directly
- One native format to another via a neutral format
- One native format to another via a standard format

Then in order to do an exchange you have three options like how do I take let’s say a data or a geometry which is created in one system to another. One is one native format to another directly like there are two CAD packages, they use their own native format but I buy a translator or I develop a translator which reads files from one system and it dumps output as another file which can be read in a second package. These are usually quite fast, this kind of a system and sometimes they are more reliable but they are sometimes more expensive too like either to develop or to buy it. Other option which we have is to go from one native format to another via a neutral format like there is, part is modeled in one of the packages. It also has a neutral format, so you can store it as a neutral format.

(Refer Slide Time: 00:40:29 min)



Geometric Data Exchange

Geometric Format (Software I)

Geometric Format (Software II)

Intermediate Geometric Format

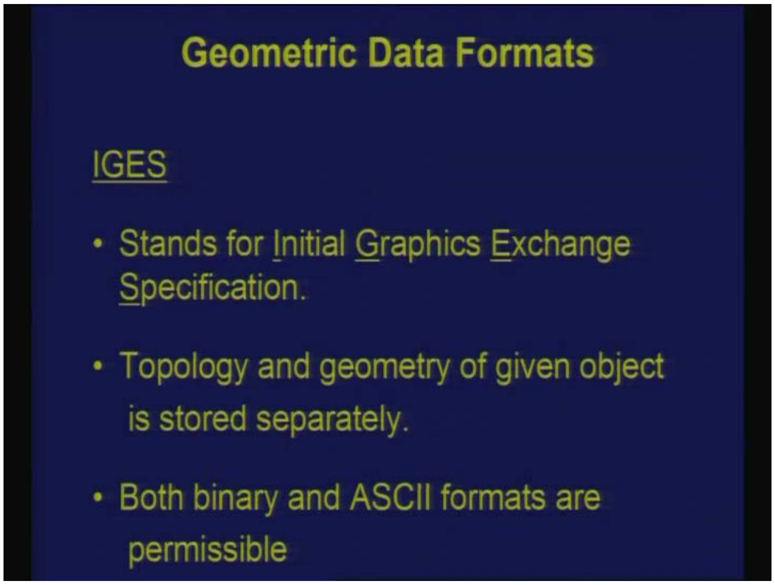
Geometric Format (Software I)

Geometric Format (Software II)

And you write an interpreter which reads this neutral format and writes an output as a native or another neutral format. So here is a second option. Third is the intermediate format which is used is an international standard or a national standard which is used to do a data exchange. Let me show this using a figure like the three options which we have discussed. I think we have seen this also slide earlier like this is a custom translation from software one to software two, I have two translator. There is no intermediate format which is coming into picture whereas here I do not have a direct translation from one format like software one to software two, I have to go through an intermediate format. This intermediate format can be an international standard or national standard or it can be a neutral format which is given by the package. So these are you can say three options, one which is represented by the first, two options which are represented by the second figure which are available for geometric data exchange.

Now this also we have seen that when the number of software increases, the number of translator here explodes whereas the number is number of translator which you need in this case would be restricted if I am using an international standard this thing. So that is an advantage of development and usage of a standard for geometric data exchange.

(Refer Slide Time: 00:41:42 min)



Geometric Data Formats

IGES

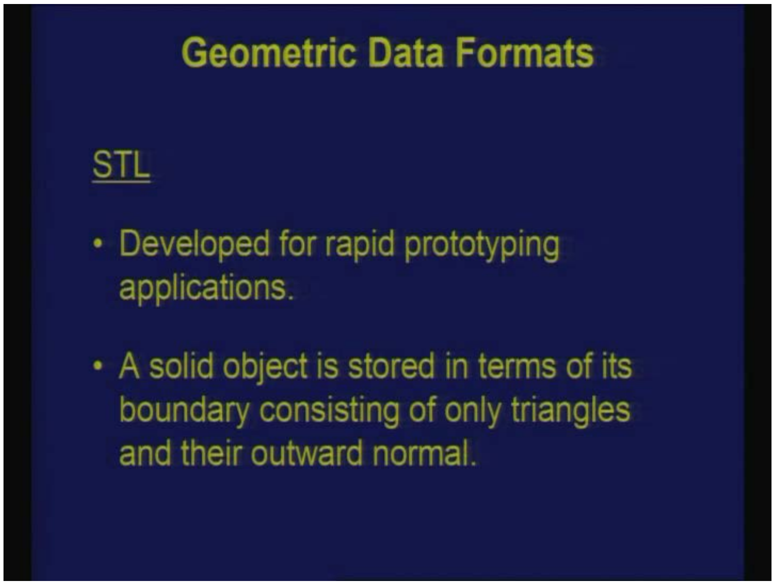
- Stands for Initial Graphics Exchange Specification.
- Topology and geometry of given object is stored separately.
- Both binary and ASCII formats are permissible

One of the very probably the most common data exchange format which is used is IGES. It is, this is an ISO standard that means it’s an international standard and it stands for Initial Graphics Exchange Specification. When the first time it was developed, it was given a name and basically the IGES is a short form of Initial Graphics Exchange Specification. So this is an effort where many nations came together and they said let’s have a format but primarily it is a effort of National Bureau of Standards of united states and this format evolved about 25 years back. That’s about typically in 80’s, the format became very popular and subsequent developments happened in 90’s and years more futures were added etc.

Now what IGES does is it basically separates the topology and geometry separately and stores them in two different sections. In one sections it stores the topology as we do more like a boundary representation and the geometry stored in another section. So these are two major sections of an IGES. In fact it has more sections, it has a section which says that which is called as a start section which tells who has generated this IGES file or with software this thing. then it also has an end section which says key, this is an end of a file but there are also a section which is in the beginning which says key like how the data is stored like suppose if I am storing let's say coordinates of a point in a space xyz, x coordinate y coordinate and z coordinate in a single line then again there may be a variation. You may have a blank between this data or I may use a delimiter as comma. So, all these things are initially defined prior to defining topology and geometry.

So you have the start section then you have section where initial attributes are defined then you have a topology, geometry and end section which constitutes complete IGES specification. so a file usually is a quite lengthy because it has to have all the options which are there and it offers both binary and ASCII like an IGES file can be created in a binary as well as ASCII option for usage which is there and IGES one option which is given by almost every CAD CAM vendor. So one need not actually buy this as this thing. so if I buy let's say a CAD CAM software, it's most likely that the package also has an option where the geometry which is modeled in the package can be dumped as an IGES file which can read by other package but sometimes IGES also has a versions. So one should be careful like IGES file which is generated in one version may not work another version, so that problem has to be dealt.

(Refer Slide Time: 00:45:11 min)



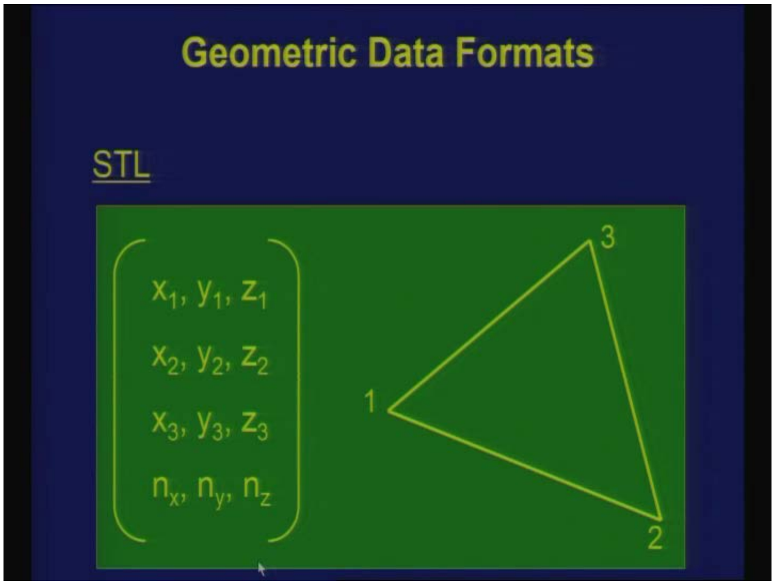
Geometric Data Formats

STL

- Developed for rapid prototyping applications.
- A solid object is stored in terms of its boundary consisting of only triangles and their outward normal.

STL is another format which is used by rapid prototyping applications. This is particularly developed by one of the first rapid prototyping machine manufacturer, one of the first rapid prototyping technologies which has come out is, which was commercially successful was stereo lithography. So this format was called as a stereolithography format and which has been shortened as STL. STL actually is a short form of stereo lithography. Here an object is stored in the form of a, solid object is stored purely in the form of its boundary so it's a B-rep type of format but the boundary is also consists of only triangles. The entire object can be completely represented as triangles. Suppose if I take let’s say parallelepiped, if I have a box like object, it has 6 faces each face is a rectangle but I can divide each rectangle into two triangles. So I can store the object as 12 triangles or if I have an object like curved object let’s say a sphere, I can take enough points on this particular sphere and have an approximate polyhedron version of the sphere in terms of triangles. So it basically approximates the object in terms of triangles and the object is stored purely. And apart from the triangle information, so when you open an STL file what you see is that here is an object which consists of let’s say about 3000 triangles. I have xyz coordinates of all the triangles. It not only stores the three vertices xyz values, it also stores the normal information. Whenever I have a triangle, it has two normal, one which you call as outward normal which represents the like if I have this thing, I have xyz which basically represent but I also have nx ny nz. One is basically represent the outward normal, the direction which says there is no material.

(Refer Slide Time: 00:47:12 min)



Geometric Data Formats

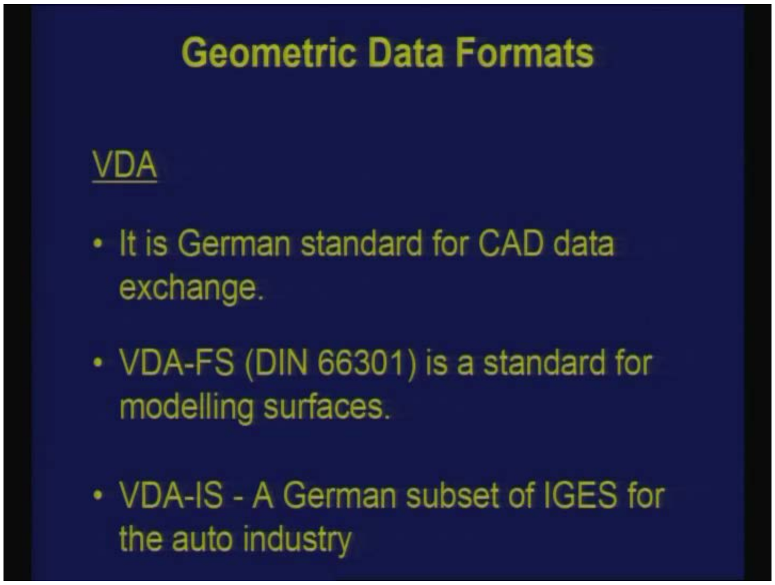
STL

$$\begin{pmatrix} x_1, y_1, z_1 \\ x_2, y_2, z_2 \\ x_3, y_3, z_3 \\ n_x, n_y, n_z \end{pmatrix}$$

Then three is an inward normal which tells that this is the direction like every triangle which is represented as a solid boundary will have a material on one side, there is no material on the other side. So in order to distinguish these two, the nx ny nz which also gives you an out word normal. if I have this I have a complete information for a solid and I can use that like this is a very convenient representation for many applications like suppose if I want to know the surface area of a given solid, very easy you read all the triangular information, find out the area of these triangles, sum it up and area of a triangle is a very simple expression.

And I have an approximate surface area of this particular thing. How good is your approximation depends on, how many triangles are used to approximate a given object. In some cases a fixed number of triangles are enough to represent the object exactly like we have, earlier we took an example of a box like object, a rectangular parallelepiped. 12 triangles represent the boundary completely without any loss of information. If I take let’s an a object like doubly curved object like a sphere, irrespective of how many triangles I will take it will always be an approximate value. I won’t be able to get exact value. And when you generate, again STL is option which is given by most of the CAD CAM software. I can model an object in one of the CAD systems and say that use a simple command to generate an STL file for that. And when I looked at the STL file, I will have all the triangles which this particular object consists of. So it's very easy to generate and easy to use as far as this. It's not only use for rapid prototyping applications, many times it's used for other applications too where the simple data formats are desired.

(Refer Slide Time: 00:49:35 min)



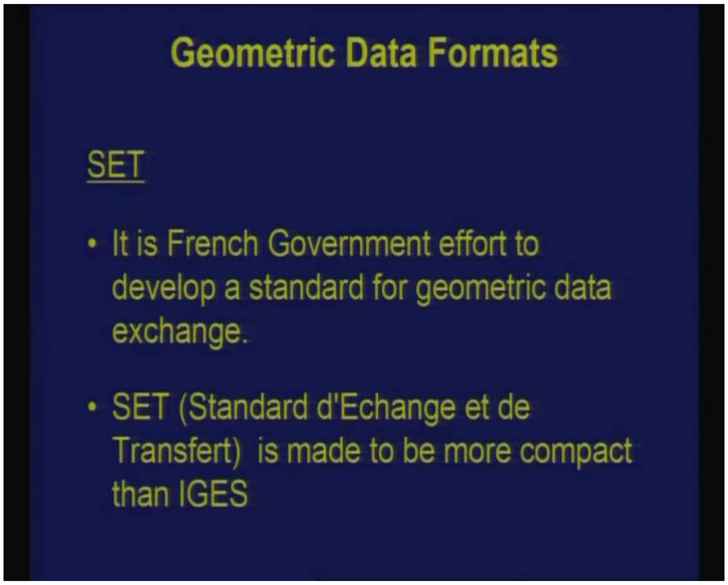
Geometric Data Formats

VDA

- It is German standard for CAD data exchange.
- VDA-FS (DIN 66301) is a standard for modelling surfaces.
- VDA-IS - A German subset of IGES for the auto industry

VDA is another geometric data exchange standard. This is actually a national standard; it's a German standard for CAD data exchange. There are two options one is basically used for the surfaces and we know that most of the German standards have this number DIN which is there. So this VDA-FS has DIN number 66301 which is used for surfaces then there is a VDA-IS which is for solids. It is some kind of a subset of IGES particularly followed in auto industry, automobile industry. So these two are you can say a popular formats which are used. And many software packages give an option of VDA to, but many of times you may have to buy it, it may not be readily available. If I know that I have to deal with those firms where VDA is one of preferred standards, so I may.

(Refer Slide Time: 00:50:36 min)



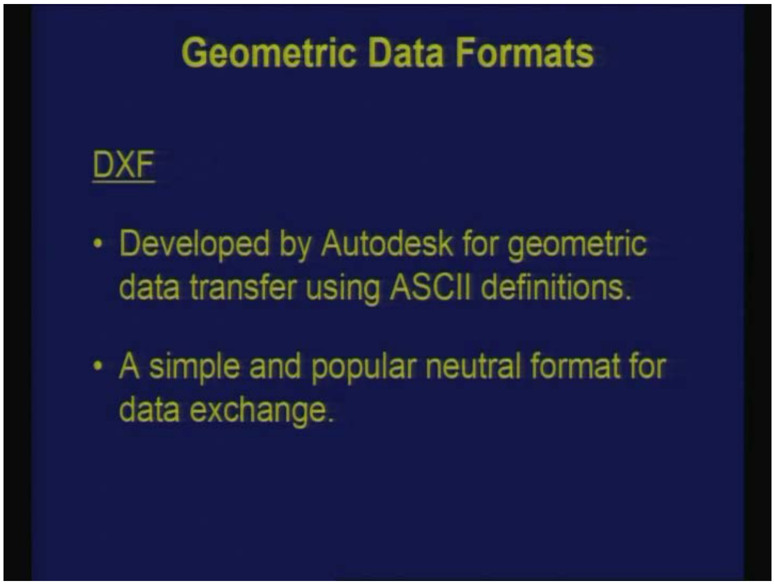
Geometric Data Formats

SET

- It is French Government effort to develop a standard for geometric data exchange.
- SET (Standard d'Echange et de Transfert) is made to be more compact than IGES

Similarly there is a French standard this is a set I can model a geometry and dump it as a set file but one of the reasons for developing this SET standard by French government efforts is to come up with a format which is more compact than IGES. IGES is a very verbose format in the sense even if I represent let’s say a single line or a triangle and try to generate a file, I may have a few hundred lines of code which is basically because you have to represent the complete information which is necessary including all the attributes, delimiters etc. So if I want let’s say more compact format, the effort was done and SET standards are used for representing. So if I have to basically work with let’s say French industry and if they have an option of SET then this may be an advantage.

(Refer Slide Time: 00:51:35 min)



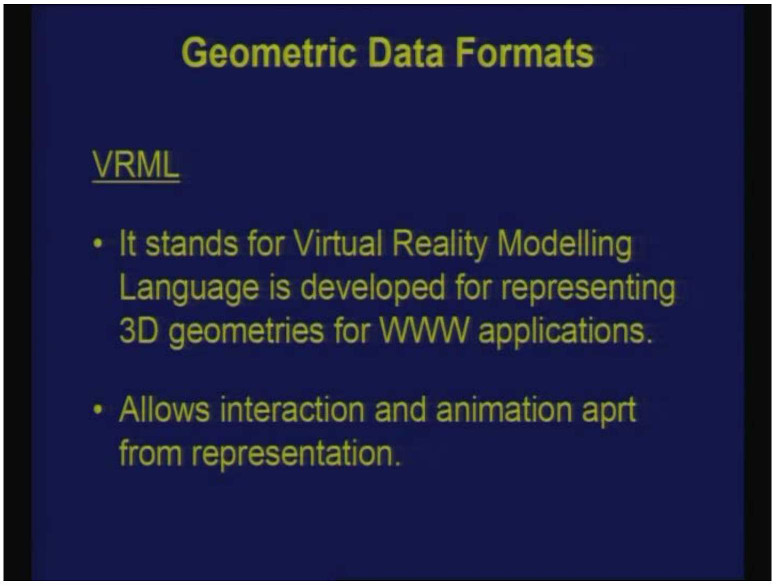
Geometric Data Formats

DXF

- Developed by Autodesk for geometric data transfer using ASCII definitions.
- A simple and popular neutral format for data exchange.

DXF is some of the standards which we have seen like IGES etc SET, VDA, those are IGES is an international standard, SET and VDA are the national standards. DXF is one of the neutral formats; this is basically popularized by a company which developed auto desk which also has a product auto CAD. So auto CAD gives you an option like if I am working with auto CAD, to either store files in its own native format which have an extension like DWG or I can also dump the geometry as a DXF file. The advantage of DXF file is, suppose if I am writing an application, I can read the DXF file and interpret and use that geometry for my applications. Though it was developed by one specific company but it's quite popular for variety of applications where it has almost become a standard in many senses because many packages give an option of DXF as a data exchange option.

(Refer Slide Time: 00:52:47 min)



Geometric Data Formats

VRML

- It stands for Virtual Reality Modelling Language is developed for representing 3D geometries for WWW applications.
- Allows interaction and animation aprt from representation.

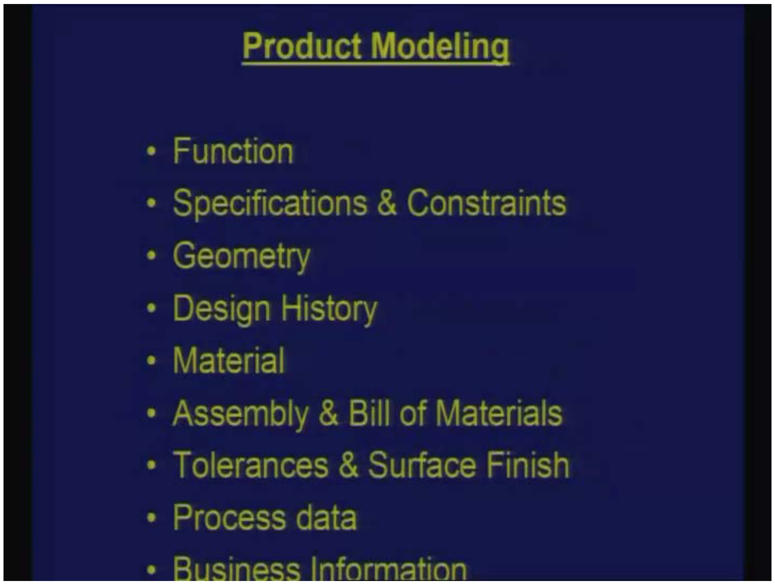
VRML is another standard. This is again more of an international effort where idea was to come up with exchange of geometry over a World Wide Web application. It's like, now suppose I have a let's say I am modeling a certain object, it could be a mechanical object or it could be anything let's say it may be a building. Now, once I build a CAD model of the object and convert into VRML format and post it on the web, so anybody who has a VRML browser can open and see it. the advantage of this is it also gives you, the user can interact with the object like I can have a walk through inside the object or a simple animations are also possible like I can give one very good example for which VRML is used by certain people is for assembly animation like I buy a product and this product which I buy consists of many components.

Let's say it's a toy and this toy consists of many components and there are assembly instructions which are given. So when you buy this particular product, take home, you read the assembly instructions, assemble the product and your product is ready for usage. Sometimes the assembly instructions may be like quite complex that means you are not able to interrupt. So you can go to the user side, they have assembly animation of this particular product. So you just open a VRML file, you can watch how the object is being assembled. You can do it using your own internet because VRML is primarily developed for World Wide Web applications just like we have a zip and jpg for image representation.

So I have a VRML for three D representation. I will watch a VRL animation at the site of the company which is marketing or which is producing this product. Now the assembly instrumentations are much more clearer because I am not just reading the instruction but I am watching the assembly which is been carried out. So I can do it in a much more convenient manner. So this just one application like in one of the other applications where people did is the new airport which was coming was completely modeled as a CAD model and then you post it as a VRML. So before the airport is coming up, people can look at what would be like the complete layout of airport where the landing is going to happen. And so the complete airport is modeled as a CAD model and dumped as a VRML. It’s like giving an option to all the people to critic on let’s say layout and layout and structure which is coming. So there are numbers of, whenever I use the internet and if I have to deal with three dimensional geometry VRML is a very good option.

VRML again store the geometry more like B-Ref type of format. I can store the object as a polyhedron with boundaries and others and offers number of primitives to build more complex objects. And lastly so far we have been discussing about geometric modeling but now it has been realized that exchange of only geometric data is not enough. I think this we have discussed earlier too that one should be in a position to exchange a product information which is, which not only has a geometry but there is other information like I am storing a product geometry but I am also storing a complete process plan for that particular product how the component will be manufactured, what kind of machines and tooling will be used, can all be stored if I go for a more elaborate product model.

(Refer Slide Time: 00:56:52 min)

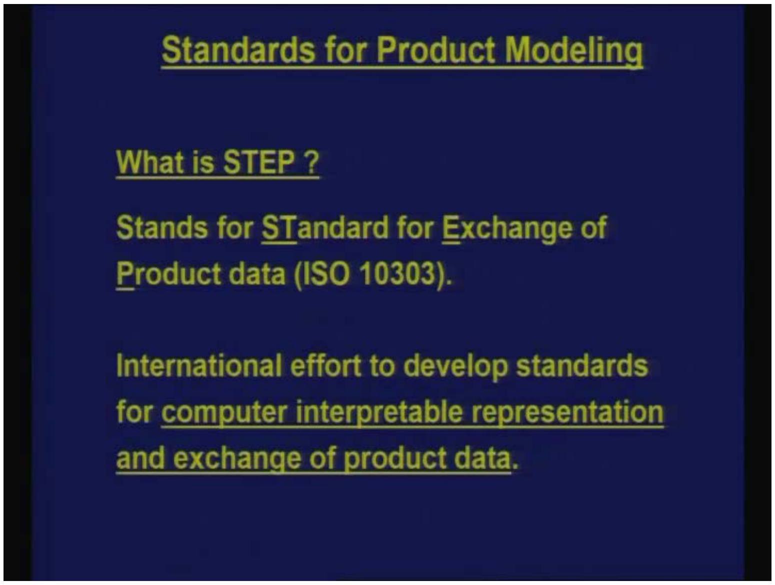


Product Modeling

- Function
- Specifications & Constraints
- Geometry
- Design History
- Material
- Assembly & Bill of Materials
- Tolerances & Surface Finish
- Process data
- Business Information

And like as we looked at, one can store the function aspect of the product, its specifications, its geometry, its design history, the material information, assembly information, tolerances, process data and business information should be stored not necessarily the geometry.

(Refer Slide Time: 00:57:12 min)



Standards for Product Modeling

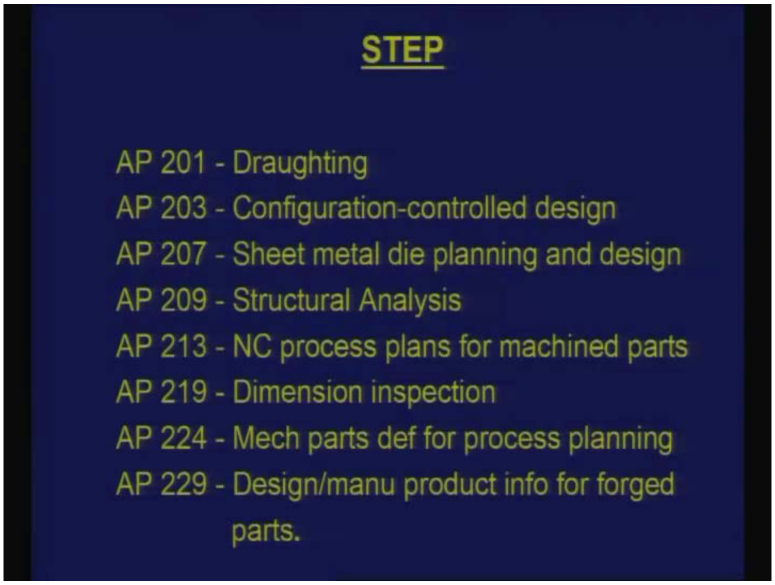
What is STEP ?

Stands for STandard for Exchange of Product data (ISO 10303).

International effort to develop standards for computer interpretable representation and exchange of product data.

And one international effort again, an international standard which has been developed to take care of most of these data exchange not geometric data exchange, product data exchange is what is called as a STEP. STEP is again an initiative of many countries European, US, Japanese and many others and primarily again initiative of you can say US Bureau, US National Bureau of Standards and it has a ISO number 10303. And nowadays some of these information’s like assembly or tolerances can be very well exchanged using a CAD systems. So STEP offers which is already many of the CAD CAM systems give you an option to exchange assembly or tolerance data using a STEP which was not like, assembly information was not possible with IGES like you have all worked with an assembly example. When you are assembling you have a parts but you also have mating relationships. You say, you impose the constraint such that one part is located with respect to other using the mating relationships. So, even those relationships can be stored as a part of a step that is the advantage.

(Refer Slide Time: 00:58:32 min)



STEP

AP 201 - Draughting
AP 203 - Configuration-controlled design
AP 207 - Sheet metal die planning and design
AP 209 - Structural Analysis
AP 213 - NC process plans for machined parts
AP 219 - Dimension inspection
AP 224 - Mech parts def for process planning
AP 229 - Design/manu product info for forged parts.

And these are some of the, step is not like you can say it's not single this thing. It has many application protocols which are called as Ap's. So if I take step AP 201 I can exchange drawings, two dimensional drawings. If I have AP 203 which is called as, if I have step AP 203 translator that means I am able to exchange the geometry, part geometries assembly information etc can be done. So you have verity of cases which can be done for example if I have AP 209, I can exchange the finite element analyses information. I have done finite element analysis, let's say in package and I would like to see the results in another FEM package though analysis is done in this, it should be possible because both of them give you an option to translate the data with AP 209.

So you are going beyond geometry, when it comes to these applications. STEP is still under development but many of them are already international standards. So this offers you can say one of the options for like exchange of product data. What we will do is may be in one of our practical classes we will see some of these formats visually, taking an application that should be much more clearer. So if you have any questions? Student: Sir, like the various formats we are discussed so far like IGES and STL, sir can we classify that standard format we have discussed earlier. Yeah, your question is whether IGES and STL can be translated as standard formats. IGES is international standard, it is an international standard, it has an ISO number. STL is basically developed by one of the rapid prototyping machine manufacturer but it has been accepted by all the RP manufacturing community.

So irrespective of whatever RP machine which I buy most of them they accept STL as one of the formats. So, it's not truly an international standard in the sense but it is an international standard for a certain domain which has been accepted by most of the, a consortium consisting of rapid prototyping machine manufacturers. And there are also efforts to replace, not replace, to find alternative to STL because STL stores in the form of a triangles whereas in rapid prototyping you manufacture the objects slice by slice. Can somebody store directly a slice information. So that is one of the some people given option of CLI format which is basically stores a slice information.

That’s not readily as standard as STL option which is available. Sir in STL format which is a material renders side having normal or the other side. This is outward normal. Outward normal means the one which is pointing it out to outside where there is no material. So if there are no questions then I will stop it here.