Cloud Computing

Prof. Soumya Kanti Ghosh
Department of Computer Science and Engineering
Indian Institute of Technology Kharagpur

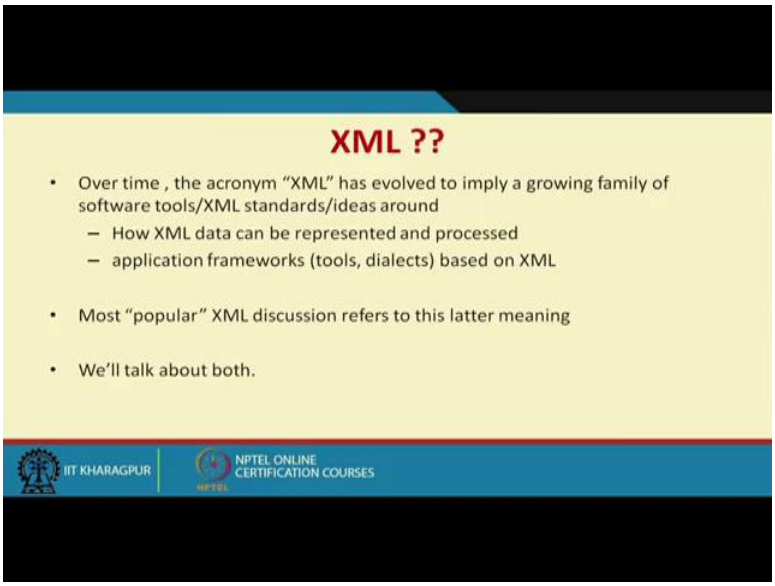
Lecture – 08
XML Basics

Hello. So, welcome to this course of this cloud computing. We will continue our talk on different aspects of cloud. So, today what we are trying to look at that, what is the underlining technology or underlining protocol which binds together. As we have discussed that there are in cloud, there are various type of services or XaaS, anything, as a service type of model it service model it was. So, we need to have some way or other a protocol which allows that to interoperate between different things.

Now, if you look at whenever a user or a customer or a consumer of cloud, is taking the service of the cloud provider. So, it is independent of where, how this cloud is hosted the provider ends. How the user client is accessing the things. So, it is more of service oriented architecture, what we are basically implement in this cases. So, web service oriented architecture or web services as we have as we know that, allows us to interoperate between loosely coupled heterogeneous services, to achieve some expected output right. So, all this type of service exchanges or service driven architecture the basic one of the basic component or one of the basic building block is XML, right. So, what we will discuss today maybe one or two lectures that the basics of XML.

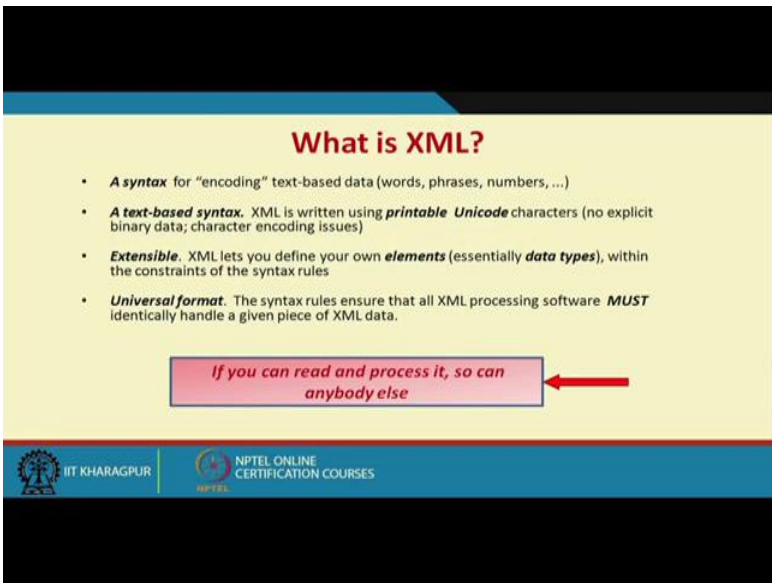
What I understand that, most of you are used to XML or you know what is XML. But for making it little those who are not accustomed or making that looking at all sorts of viewers and listeners of this particular lecture series we want, we will discuss a very basics of XML and try to see what are it properties and how it allows us to interoperate, right.

(Refer Slide Time: 03:02)



So, if we look at XML. So, it is acronym XML as evolved from a employee for a growing family of software tool XML standards and ideas. So, what we will try to see that, what are the different properties of XML? How it is, what is how it allows that to interoperate. What are the different types of parser or what are the type of other technologies, which are there in the XML.

(Refer Slide Time: 03:27)



So, it is extensible markup language right. We are used to HTML, hypertext markup language, which are primarily used in this world of internet, to visualize document, right.

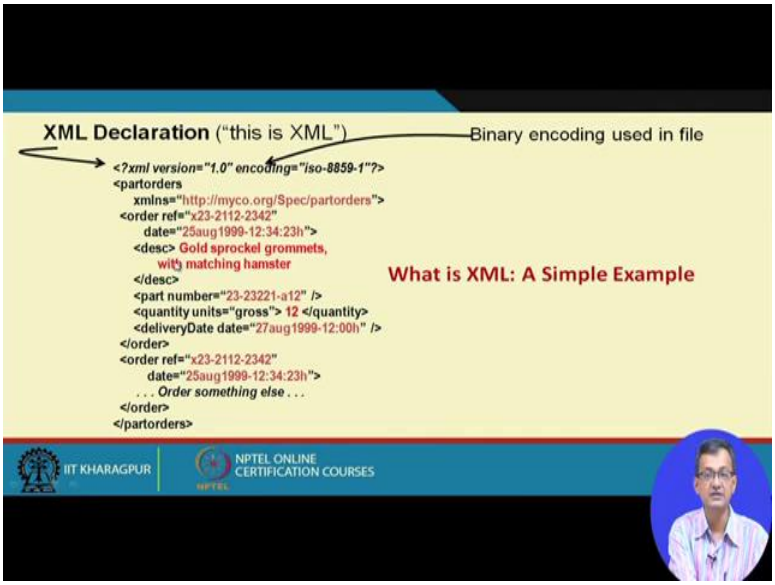
So, XML is a extensible markup language and unlike HTML which is more of a data display or data representation language. Your information representation language XML is more of a data transformation language, right.

So, it allows us, we will see your; it allows us to achieve interoperations, right. So, the syntax for encoding is text based. So, words phrase numbers. So, it is a readable, a text based syntax and XML is when well written using printable Unicode character right, no explicit binary data; character encoding issues etc right. It is extensible; XML let is you define your own elements. Like data types, like unlike html those who are used to HTML or though as have seen the HTML, we have a predefined set of tags right.

Whereas in XML you can defined your own elements or own tags. So, it is within a constant of syntax rule definitely you can define within a context of syntax. So, it is also sometimes referred to a universal format right. A syntax rule ensures that all XML processing software must identically handle given piece of XML data. So, it is universal. So, if you have a XML data, and XML processing software, or say XML parser, it will be able to handle all sort of XML universal right. So, that is the typical basic characteristics of XML, which makes it ubiquitous.

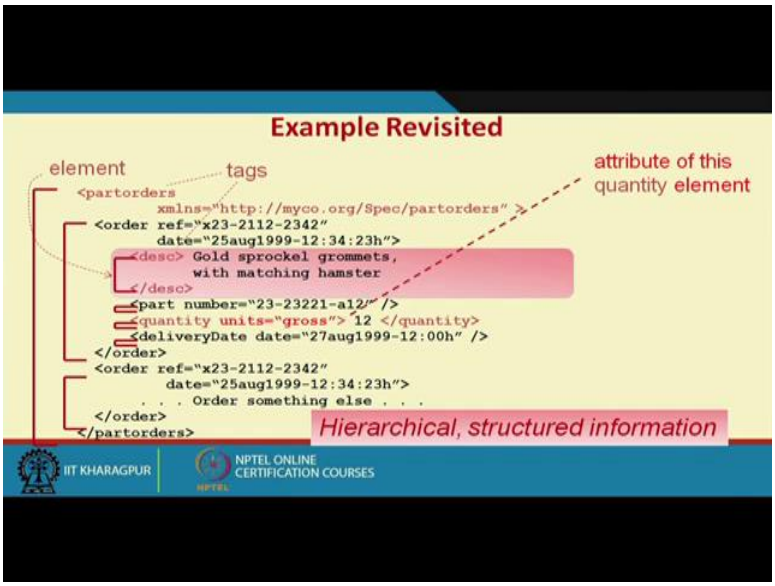
So, if you can read and process it; so, anybody else, this is the means basic; so, if you if at one and if you can read and process it so that everybody else can read and process the same data.

(Refer Slide Time: 05:53)



So, it is if I look at a simple example like we have taken from resource. So, that there are some of the things like there is a XML declaration. Like we says the version and there is a will come slowly we will come to those things, there is a XML name space and there are definition of different elements and type of things here, right.

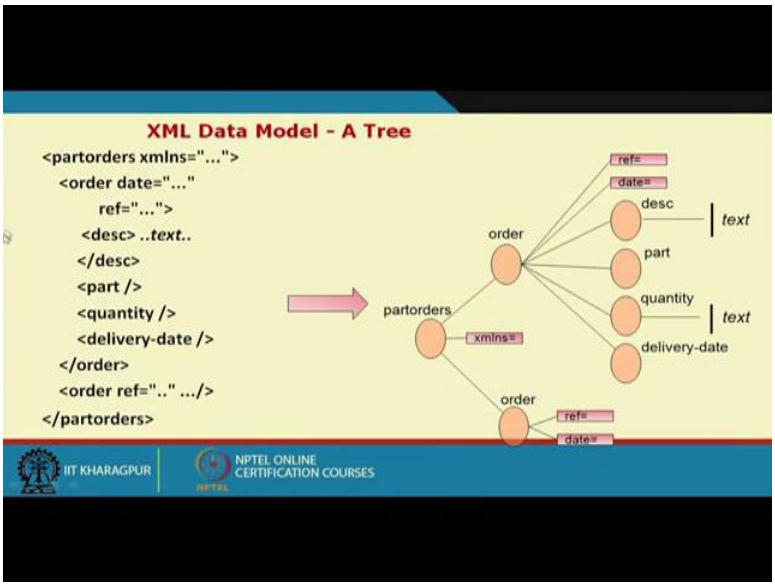
(Refer Slide Time: 06:18)



So, these are different XML tags, if you see this tags are user defined right, tags can be user defined; that means, I can define my own tags, that this part orders or description or part number, these are all user defined tags, right. So, unlike html, where the tags are predefined, you cannot define your own tags here you can defined your own tags. Now if you defined your own tags, the next things which come into play. That there can be you need to say that, how tags how things are defined. Like I say I define a tag called table. Now whether, I am referring to the table, whether this table refers to some furniture classes or group in the category of furniture or the table is basically in the category of word processing or phrase it right.

So, that is important right. I need to know that, if I define a tag or if there are I am using table one from this furniture category and another for the word processing category. Then I have to define, the things or how the other end knows that what I am looking for. So, there is a concept of XML namespace, which primarily allows us to define my own user defined elements and other things. So, if we look at the XML document, it is hierarchical and it is structured information, right.

(Refer Slide Time: 07:51)



So, if I look at this document. So, it is a part order all right. It is the top most node which is a XML namespace, that it says that part order is of this namespace and it has different levels or it looks it basically hierarchical and we generate a XML tree, right.

So, XML comes with this property of hierarchical structured information which can be represented by the XML tree.

(Refer Slide Time: 08:32)

So, again if we come back to our basic premise or basic definition, so it is simple like html, but not quite so simple, html is very vanilla type. So, strict syntax rules to eliminate

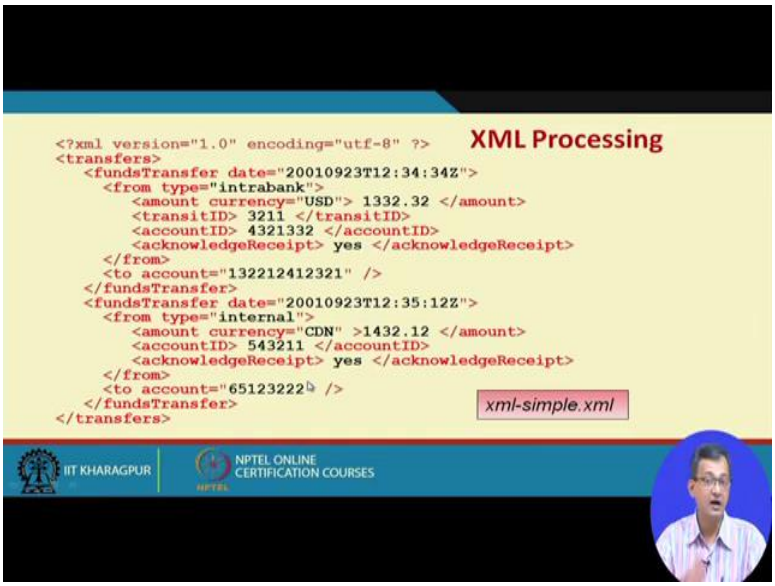
syntax error. So, XML has a very strict syntax rule. Syntax defines structure hierarchical and name structural element names and it is self describing data. So, it is say the I can have my own data, which is self describing. It is extensible unlike html vocab is not fixed in case of a; it is XML. You can define your own vocab or tags or elements and that means, that you can basically extend this data.

Design for distributed environment right, you are like HTML it is designed for distributed environment. It is say; it is able to talk to or it able to communicate or interface or interoperate with different systems in a or different nodes of a distributed systems. So, can create, your own language of tags and elements sorry it can have data all over the place can retrieve and use it reliably. So that means, it is distributed can mix different data types together unlike HTML. So, html can do it. But in XML, I can have multiple sources and mix them and generate another data out of it.

It is like that, I from the 2 repositories. I take different component of data and finally, generate another data which is a mix of these two things as they HTML, XML allows me to do that. It is extremely helpful in different cases, like you are getting two types of data one form your, say for in academic institution, one data I am getting from academic section, other maybe from all managements centre that regarding students and then I want to club this data and try to find out some things, like I want to find out that who is the best all rounder of this particular semester or particular batch or particular year. So, I need to look at different student activity data, student academic data and other type of data impossible and then generate another data set.

So, this sort of distributed diverse data set on the fly how you can basically interoperate, how you can mix them and generate a thing is this XML is one of the is helpful in this type of cases. So, it is a mix of different data types together.

(Refer Slide Time: 11:44)

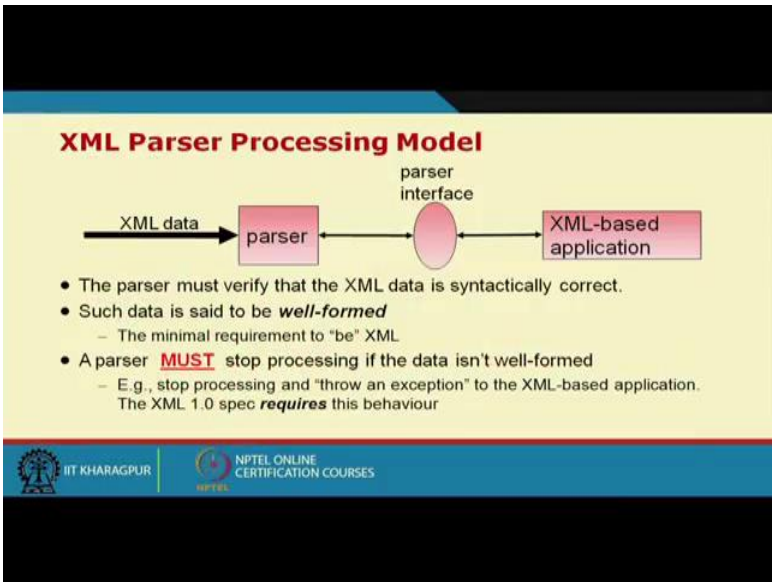


Now, other access is how to process XML data. So, one thing what we have seen, it say tree type of structure. Now in order to now unlike other standard like HTML or where you have no user defined things. So, you know the not only the syntax they are elements etc defined.

But here in this case, I need to first of all find out that, what are the in numbers of element, what are whichever is users defines and I need to find out that what they mean in the first place. So, in order to process this data, suppose I have a data like this, taken from some Internet resources. Like, file transfer date is so much type of things is interbank and so much amount transit ID and type of things, different type of transactions. So, it is like bank transactions, some are internal that is within the bank and some are interbank or between different banks so, then how to process these data.

So, for this I need to first of all extract the information from the XML data all right then my processing will be going on. It is basically enveloped in a XML type of language and then I have do extract these and process it using some application some sort of a tool that I have to extract the data; that what we do is doing through a XML parser.

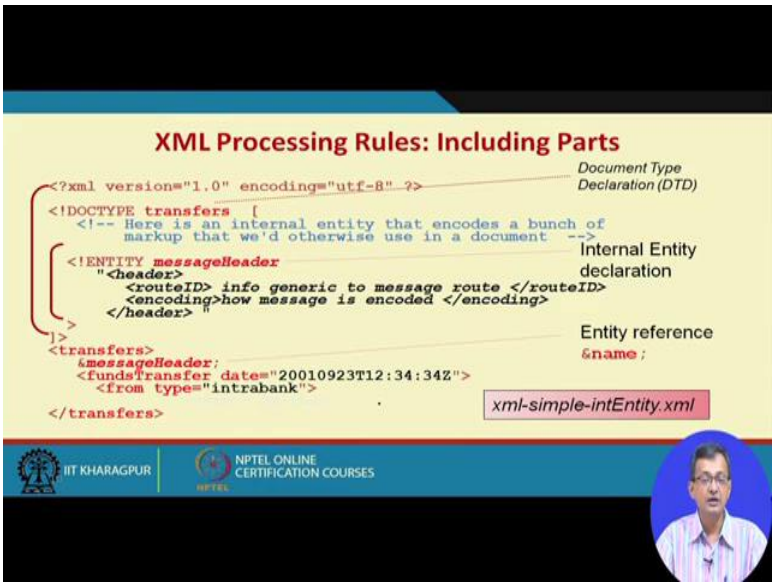
(Refer Slide Time: 13:17)



So, XML parser processing model, what it does that XML data the parser, it passes through the parser and there is a parser interface than XML based application on the other side. So, XML data then the processer, it extracts the things that XML based applications are triggered, all right.

So, the parser must verify the XML data is syntactically correct. So, what is the role of the parser, first of all the parser should verify the XML data is syntactically correct. So, there is no syntax error in the data. Secondly, such data is said to be well formed right. If it is a syntactically correct, it is a well formed XML. So, the minimum requirement to be a XML is this it should be well formed, so that it is processable, right, I can process these data. The parser must of processing, if the data is not well formed, right. So, that is stop processing and throw an exception to the XML based application.

(Refer Slide Time: 14:40)



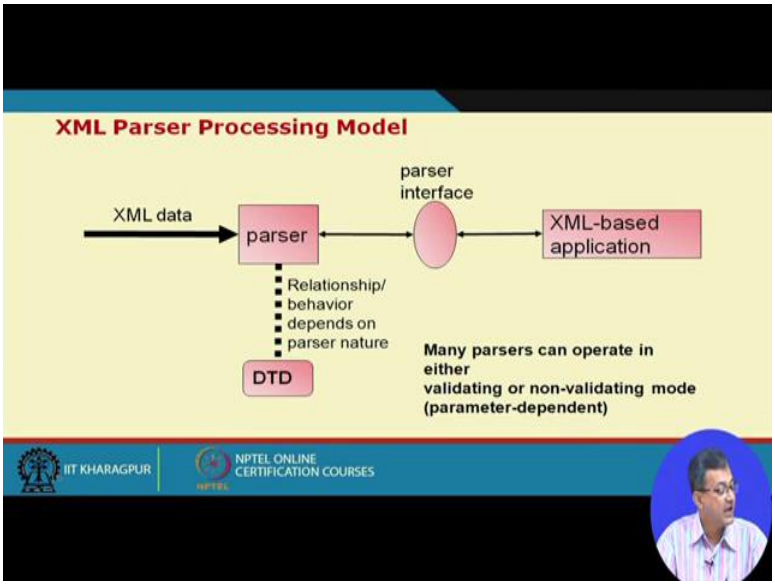
So, that if it is not a well formed; that means, if it is syntactically not a correct data. So, if we look back, the XML processing rule including different parts they so, there is a concept called DTD, that is document type declaration. So, what is important for any XML data, like we have seen in case of a data bases like relational data bases, so, what is what you have to do basically, define a table right or multiple table. So, where the data is different or in form of records or different rows or topples what we say.

Now, this in order to define this table, we need to define the structure of the table or what we say schema of the table all right so that if there are different variable of this table, then which variable is which schema and type of things we need to define, right. So, in case of XML also, as these are user defined structure and it is hierarchical and I do not know that if you define a particular element, then what are the levels of other elements and what are the types of elements are there, and that is why here also, we define we need to define some sort of a schema or in this previously it is used they that stuff called document type declaration.

So, first of all you need to declare the document type. Now based on the document type, if it is a in the then the rest of the data is there. So, one way; one of the work of the parser is to look at whether it is syntactically correct. Like you have syntactically, other than in to look at whether it follows that particular document, definition right or if it is I say that

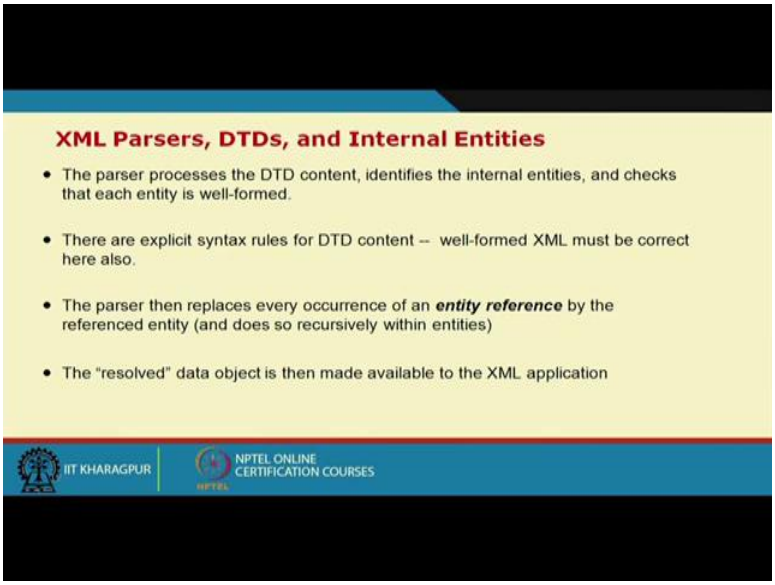
syntactically correct, is a well formed. Then if it is following that schema or document type definition then says it is a valid one XML.

(Refer Slide Time: 16:58)



So, the while getting the XML data, the parser consult this DTD. If it is syntactically correct, to say that the XML is valid for this particular type of operations details, looking for.

(Refer Slide Time: 17:15)

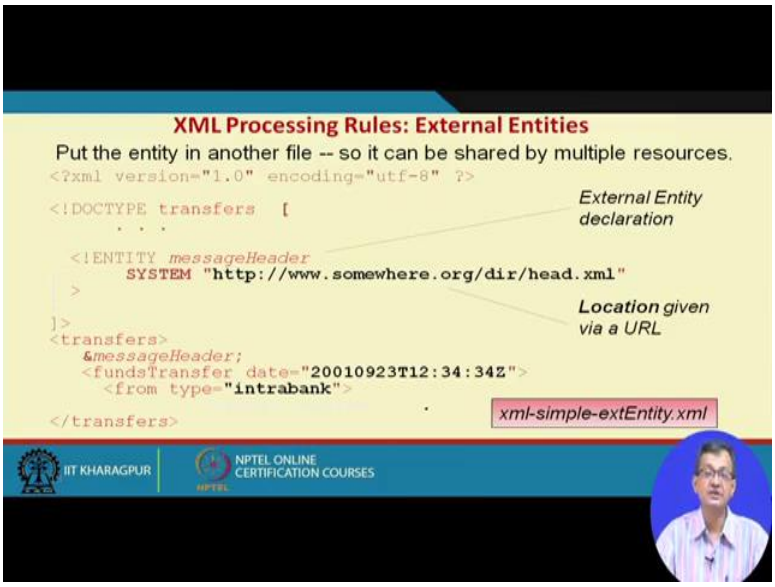


So, we have XML parser, DTD, another internal entities. Like the parser processes, the DTD content identifies the internal entity and checks that the each entity is well formed

correct. So, in term it checks that whether each entity is well formed. There are explicit syntax rule for DTD content, well formed XML must be correct here also right. So, what is there are explicit syntax rules for DTD content so the well form XML must be correct in this respect also.

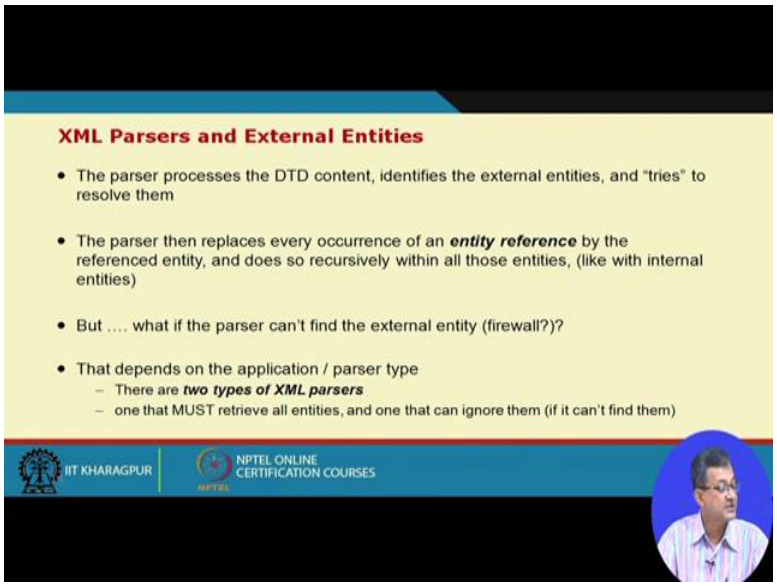
The parser then replaces every occurrence of entity reference by a referenced entity right. So, does the recursively within the entities. So, it reference entity by the referenced entity it replace so that it is entity reference, the resolved data object is then made available to the XML application. So, when it goes to this XML application, if we just remember this previous picture, then by that time it has done It is well form checking syntactical checking and also the checking with the DTD, that is what we can say some sort of a validating, the particular document before pushing it to the XML application.

(Refer Slide Time: 18:45)



So, in XML processing there can be external entities right. Here it was internal entities, previously what we have seen that is internal entities means that DTD is defined here only or the schema is defined here. So, external entity is referred by a URL. So, it basically referred to external structure or schema to look at the entities. So, the parser processes the DTD content, identifies the external entities and tries to resolve them, all right.

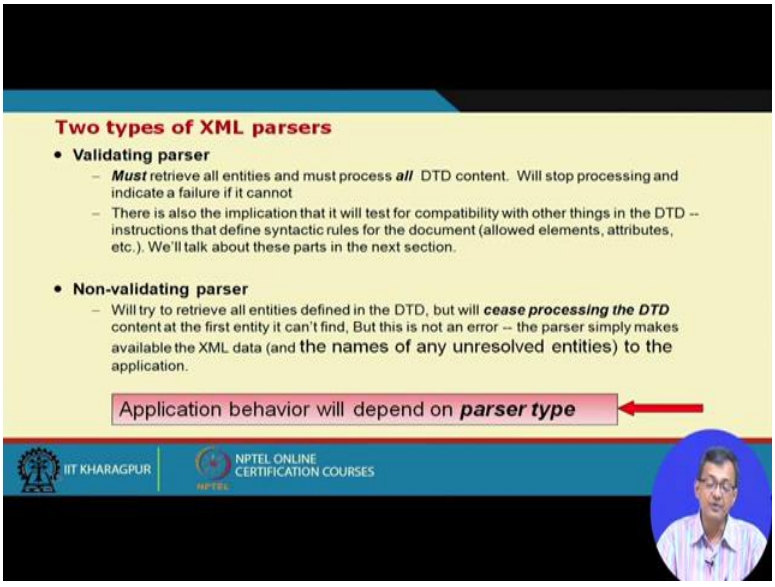
(Refer Slide Time: 19:22)



The parser then replaces every occurrence of the entity reference, by the referenced entity and does so recursively with all entities like with the internal entities. For the external also it find outs and go on replacing this. But what if the parser cannot find the external entity, due to may be due to firewall blockage etc. That depends on the application and parser type. There are two type of XML parser one mass retrieve all entities; one can ignore them if they can find it.

So, it based on the; what is your basic policy or processing rule for this parsers.

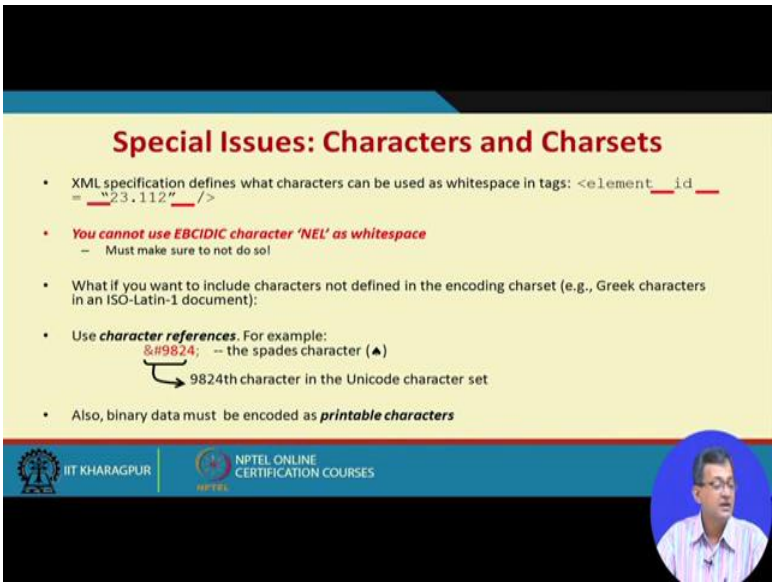
(Refer Slide Time: 20:20)



So, accordingly we have 2 type of XML parser, one wise say validating parser; that means, must retrieve all elements and must process all DTD content all right. We will stop processing indicate a failure, if it is cannot, right. So, what we are trying to say that in case of a validating parser. It should conform to this schema definition or the DTD content, right. Non validating parser will try to retrieve all elements defined in the entity, but will cease processing, the DTD content in the first entity if it cannot find, right. So, it based on validating parser or non validating parser application behavior will depend on the parser type definitely. So, it the application how it will work is based on that what sort of parser.

So, we add on something more, like here XML data then parser. Then it goes to the parser interface to the XML application, right. So, DTD is linked with the parser, which gives relationship, behavior dependency on parser nature. So, these are the things who is the DTD provides to this parser for making for checking, for validating well formed and validating, right.

(Refer Slide Time: 21:51)



So, there are some special issues, characters and character sets right. So, like the XML specification, defines what characters can be used in whitespace tag and like you cannot use EBCIDIC character NEL as whitespace tag. What if you want to include character not defined in the encoding character sets right. So, there are different ways out for

handling this sort of character sets all right. So, the XML provides some mechanisms to handle this sort of character sets.

Now, how do I define language dialect, finally, what we are trying to look at is basically how to define an XML document and how it can basically interact with each other right. So, as we know that in service oriented architecture, we are having 3 major components right. Like consumer, provider and a registry or type of services right or some sort of arrays to depository. Now whenever there is a communication between these different service providers, service consumer, service registry what we need to look is that, how this data communication go, will be executed in a ubiquitous way. Like as we you know that, this is a scope messaging maybe one mechanism which is primarily and XML document all right. Like soap WSDL, UDDI, I all are basically XML documents correct.

So, if you look at, two ways to doing that XML document type, declaration part of core XML spec, XML schema, new XML specification, which allows for stronger constraints on XML document. So, what you have looked at is DTD, now 2001 the XML schema or XSD has been defined. So, it is a higher version of on spec it is a new spec of defining the schema of the structure of the XML document. And one basic or fundamental difference is there other than they primarily do the major same type of job; one is that XML schema is written in XML, all right; whereas DTD is actually written in a different way. So, that now my handling of the schema in XML data goes hand in hand.

So, adding dialect specification implies two classes of XML, one we call well formed an XML document that is syntactically correct right. So, what we say, it is a well formed XML, right and where as there is a thing called valid XML, right. So, XML document that is both well formed and consistent with the specific DTD or XSD or schema all right, these we are called valid. So, it should be well formed along with a; that it is basically conforming to the schema; that means, it is a valid scheme.

So, all valid XML document are well formed. So, what DTD and other schema specify allow elements and attribute data, hierarchical nesting rules, element, content and type restrictions. So, this is a schema or thing specified. So, it is not the data. So, the XML schema or the DTD does not content any XML data right. Information is contented in the XML file, but the structure is defined here. So, this makes a unique property, like whenever there is a two organization A, B want to exchange information.

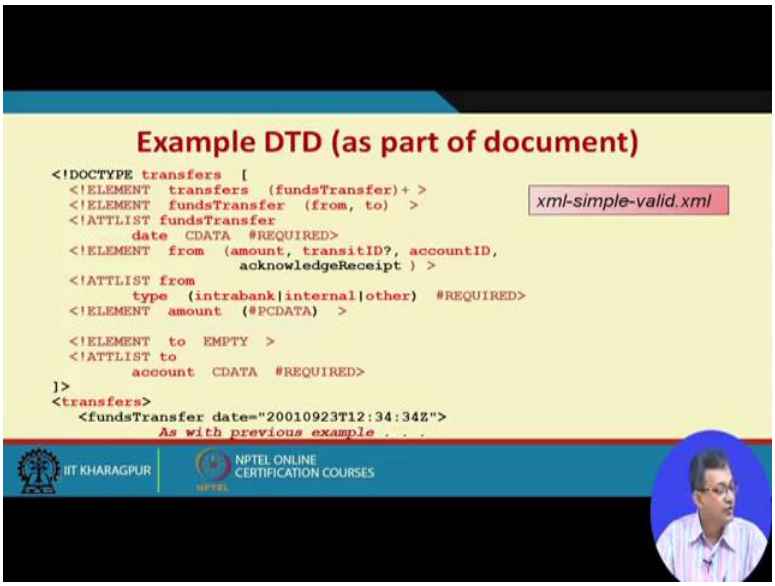
So, what they are looking for is first of all this should agree upon the structure of that exchange protocol right. Like if say for example, IIT Kharagpur having a say data transfer, say payment gateway or some payment transfer, payment interaction with bank or any financial organization. What is important that, first of all the structure of a student data and at the IIT Kharagpur end and the way they stored in the State bank. They are or any banks State bank or Punjab bank or anything, any bank they are should be that the banking organization there should be agreement.

For that date, for that need I do not require any data to be exchanged right? So, that is the basic duty of the thing right. So, for that I do not require any data to be exchange. What we required is more of the structure, to be exchanged and incidentally exchanging structure is not that critical right. So, the data there are privacy issues, there are issues of maintenance costs towards collecting maintenance of this data. If you just like that, say or then it is basically go on out of your hand. So, that is why the data has lots of issues, but if I say that see in IIT Kharagpur, this student structure is like this. So, student schema is like this right. I do not tell that how many students are there what are the properties but the school student's schema is like this.

Then if I am, if it is interacting with some other organization, the schema wise integration is possible. So, that is why allows element attribute, names hierarchical nesting rules element content type restrictions. Schemas are more powerful than DTDs, they are often used for type validation or for relating database schemas to the XML models, right. Schemas are more powerful much more powerful than DTD, right.

They not only keep the structure, but that can be used for different type of things. Like using the schema, I can create a database table and then I can transfer the data ubiquitously among the two things right. So, this gives a extra handling to this schemas. So, these days we are primarily using schemas and schemas plays a important role in handling this sort of inter operations between different parties.

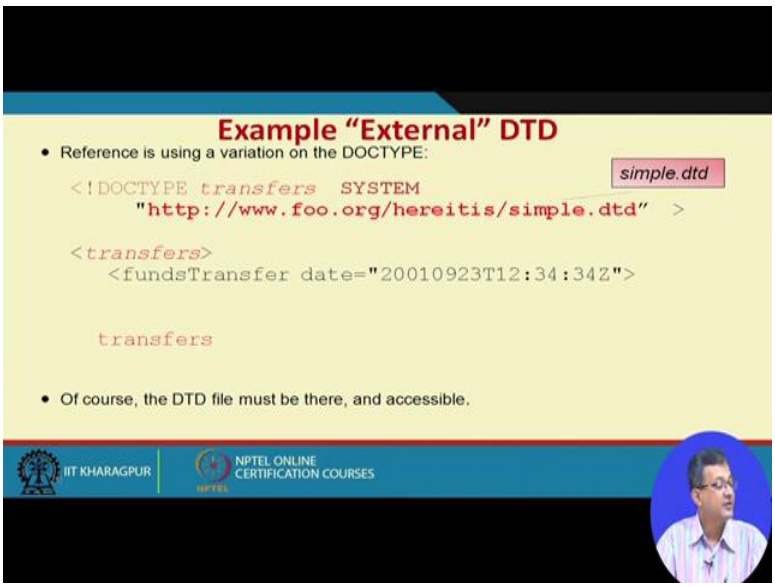
(Refer Slide Time: 29:20)



So, if you look at the example DTD as a part of documents. So, there are document type, elements transfer, this there is a some other type of elements, right.

Now, if you look at this, this is not exactly following the truly the XML file or XML document, all right. Whereas, in case of a schema, it is represent as the XML document, right.

(Refer Slide Time: 29:50)



And there can be external DTD like there is a food.org something DTD. So, it refers externally when it needs to referring. So, this allows us you know to referred to some

others DTD and see that what sort of data is coming right suppose I want to look at I am consuming a data from another provider and I want to know that what sort of data is coming up like whether they are ordered in a particular fashion whether there is a typical hierarchy.

So, at the other end I can do external DTD and check it that what sort of data. So, that allows me allows my parser to filter and extract the data I need for my end to process. So, my objective is that I am getting a data from a source, I want to filter it extract the portion of the data I want to do and may need some transformation like I want to change the unit from centigrade to fahrenheit or meter to feed and type of things, even that require some transformation that I can do at the this end and then I put to my applications which work on this. We will continue our discussion on XML some other means some other properties of XML to what to see that how these are useful for this interoperate is interoperation and how these are useful for realization of a cloud especially the software as a service type of models.

Thank you.