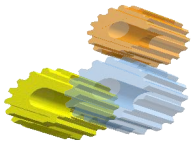
technicks

your tech side



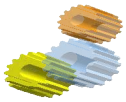
the online
technical journal
for the avid
scientific writer.

VOLUME I

email your articles to anupam.vit@gmail.com

www.anupamsingh.cjb.net

technicks
your tech side



Dear Friends,

It gives me immense pleasure to introduce to you technics, an online newsletter, journal or magazine. All articles here are written by the avid techies who love to write for science. Mostly students, they all love to share their knowledge with the community because they believe that learning is all about sharing and experiencing. The best articles every month are forwarded to IEEE Madras Section for publishing in their newsletter.

I would love to share technics with all you friends and would really appreciate your articles. You can email them to me at anupam.vit@gmail.com.

Spread technics to your friends and join the scientific community of learning, exploring and sharing.....!

With Warm Regards

Anupam Singh
104, Nelson Mandela Block
Vellore Institute of Technology
Vellore, India.

Artificial Intelligence:
"Emulating Human Conscience"

Anupam Das

It is not my aim to surprise or shock you--but the simplest way I can summarize is to say that there are now in the world machines that can think, that can learn and that can create. Moreover, their ability to do these things is going to increase rapidly until--in a visible future--the range of problems they can handle will be coextensive with the range to which the human mind has been applied. --Herbert Simon

Definition

Artificial Intelligence (AI) is the area of computer science focusing on creating machines that can engage on behaviors that humans consider intelligent. The ability to create intelligent machines has intrigued humans since ancient times, and today with the advent of the computer and 50 years of research into AI programming techniques, the dream of smart machines is becoming a reality. Researchers are creating systems which can mimic human thought, understand speech, beat the best human chess player, and countless other feats never before possible. The military is applying AI logic to its hi-tech systems, and in the near future Artificial Intelligence may impact our lives in a great way.

An Introduction to Artificial Intelligence.

Artificial Intelligence, or AI for short, is a combination of computer science, physiology, and philosophy. AI is a broad topic, consisting of different fields, from machine vision to expert systems. The element that the

fields of AI have in common is the creation of machines that can "think". In order to classify machines as "thinking", it is necessary to define intelligence. To what degree does intelligence consist of, for example, solving complex problems, or making generalizations and relationships? And what about perception and comprehension? Research into the areas of learning, of language, and of sensory perception has aided scientists in building intelligent machines. One of the most challenging approaches facing experts is building systems that mimic the behavior of the human brain, made up of billions of neurons, and arguably the most complex matter in the universe. Perhaps the best way to gauge the intelligence of a machine is British computer scientist Alan Turing's test. He stated that a computer would deserve to be called intelligent if it could deceive a human into believing that it was human.

AI has always been on the pioneering end of computer science. Advanced-level computer languages, as well as computer interfaces and word-processors owe their existence to the research into artificial intelligence. The theory and insights brought about by AI research will set the trend in the future of computing. The products available today are only bits and pieces of what are soon to follow, but they are a movement towards the future of artificial intelligence. The advancements in the quest for artificial intelligence have, and will continue to affect our jobs, our education, and our lives

Methods to create AI

In the quest to create intelligent machines, the field of Artificial Intelligence has split into several different approaches based on the opinions about the most promising methods and theories. These rivaling theories have lead researchers in one of two basic approaches; bottom-up and top-down. Bottom-up theorists believe the best way to achieve artificial intelligence is to build electronic replicas of the human brain's complex network of neurons, while the top-down approach attempts to mimic the brain's behavior with computer programs.

Branches of AI

Logical AI

What a program knows about the world in general the facts of the specific situation in which it must act, and its goals are all represented by sentences of some mathematical logical language. The program decides what to do by inferring that certain actions are appropriate for achieving its goals.

Search

AI programs often examine large numbers of possibilities, e.g. moves in a chess game or inferences by a theorem proving program. Discoveries are continually made about how to do this more efficiently in various domains.

Pattern recognition

When a program makes observations of some kind, it is often programmed to compare what it sees with a pattern.

Representation

Facts about the world have to be represented in some way. Usually languages of mathematical logic are used.

Inference

From some facts, others can be inferred. Mathematical logical deduction is adequate for some purposes, but new methods of non-monotonic inference have been added to logic since the 1970s. The simplest kind of non-monotonic reasoning is default reasoning in which a conclusion is to be inferred by default, but the conclusion can be withdrawn if there is evidence to the contrary.

Common sense knowledge and reasoning

This is the area in which AI is farthest from human-level, in spite of the fact that it has been an active research area since the 1950s. While there has been considerable progress, e.g. in developing systems of non-monotonic reasoning and theories of action, yet more new ideas are needed.

Learning from experience

Programs do that. The approaches to AI based on connectionism and neural nets specialize in that. There is also learning of laws expressed

in logic. Programs can only learn what facts or behaviors their formalisms can represent, and unfortunately learning systems are almost all based on very limited abilities to represent information.

Planning

Planning programs start with general facts about the world (especially facts about the effects of actions), facts about the particular situation and a statement of a goal. From these, they generate a strategy for achieving the goal. In the most common cases, the strategy is just a sequence of actions.

Epistemology

This is a study of the kinds of knowledge that are required for solving problems in the world.

Ontology

Ontology is the study of the kinds of things that exist. In AI, the programs and sentences deal with various kinds of objects, and we study what these kinds are and what their basic properties are.

Heuristics

A heuristic is a way of trying to discover something or an idea imbedded in a program. The term is used variously in AI. Heuristic functions are used in some approaches to search to measure how far a node in a search tree seems to be from a goal. Heuristic predicates that compare two nodes in a search tree to see if one is better than the other, i.e. constitutes an advance toward the goal, and may be more useful

Genetic programming

Genetic programming is a technique for getting programs to solve a task by mating random Lisp programs and selecting fittest in millions of generations.

Applications of AI

Speech recognition

In the 1990s, computer speech recognition reached a practical level for limited purposes. Thus United Airlines has replaced its keyboard tree for flight information by a system using speech recognition of

flight numbers and city names. It is quite convenient. On the other hand, while it is possible to instruct some computers using speech, most users have gone back to the keyboard and the mouse as still more convenient.

Understanding natural language

Just getting a sequence of words into a computer is not enough. Parsing sentences is not enough either. The computer has to be provided with an understanding of the domain the text is about, and this is presently possible only for very limited domains.

Expert systems

A ``knowledge engineer'' interviews experts in a certain domain and tries to embody their knowledge in a computer program for carrying out some task. How well this works depends on whether the intellectual mechanisms required for the task are within the present state of AI. When this turned out not to be so, there were many disappointing results.

Heuristic classification

One of the most feasible kinds of expert system given the present knowledge of AI is to put some information in one of a fixed set of categories using several sources of information. An example is advising whether to accept a proposed credit card purchase. Information is available about the owner of the credit card, his record of payment and also about the item he is buying and about the establishment from which he is buying it (e.g., about whether there have been previous credit card frauds at this establishment).

Conclusion

The progress of AI has been phenomenal in the last decade and the way it's growing it won't be out of way to say that soon machines will be living a common life with us as every other human being. Human life may completely metamorphise into an array of Bio-Mechanical elements. Every single characteristics of the human being can now be emulated by the machines, emotions not withstanding. It may well be debated upon whether computers are the right kind of

machines that should be entrusted with the human type intelligence. The learning procedure of a machine is still not easy. Reaching higher strata of knowledge comes through experience and the machines are still to conquer that. The originality of the human mind is still a hard nut to break through. Still it will be desirable to keep the growth and development of the machines inside the scope of human manipulation.

POP3 Servers and its Application

Anupam Singh and Anupam Mishra

What is POP

(1) Short for ***P****ost* ***O****ffice* ***P****rotocol*, a protocol used to retrieve e-mail from a mail server. Most e-mail applications (sometimes called an *e-mail client*) use the POP protocol, although some can use the newer IMAP (Internet Message Access Protocol).
There are two versions of POP. The first, called *POP2*, became a standard in the mid-80's and requires SMTP to send messages. The newer version, POP3, can be used with or without SMTP.

(2) Short for ***p****oint* ***o****f* ***p****resence*, an access point to the Internet. ISPs have typically multiple POPs. A point of presence is a physical location, either part of the facilities of a telecommunications provider that the ISP rents or a separate location from the telecommunications provider, that houses servers, routers, ATM switches and digital/analog call aggregators

How pop3 works

POP3 (Post Office Protocol 3) is the most recent version of a standard protocol for receiving e-mail. POP3 is a client/server protocol in which e-mail is received and held for you by your Internet server. Periodically, you (or your client e-mail receiver) check your mail-box on the server and download any mail, probably using POP3. This standard protocol is built into most popular e-mail products, such as Eudora and Outlook Express. It's also built into the Netscape and Microsoft Internet Explorer browsers.

POP3 is designed to delete mail on the server as soon as the user has downloaded it. However, some implementations allow users

or an administrator to specify that mail be saved for some period of time. POP can be thought of as a "store-and-forward" service.

An alternative protocol is Internet Message Access Protocol (IMAP). IMAP provides the user more capabilities for retaining e-mail on the server and for organizing it in folders on the server. IMAP can be thought of as a remote file server.

POP and IMAP deal with the receiving of e-mail and are not to be confused with the Simple Mail Transfer Protocol (SMTP), a protocol for transferring e-mail across the Internet. You send e-mail with SMTP and a mail handler receives it on your recipient's behalf. Then the mail is read using POP or IMAP.

Why POP3 is needed

The POP3 service supplements existing functionality in Windows Server 2003 provided by the SMTP service, which receives e-mail messages. The POP3 service makes e-mail messages available for download from a server, enabling a server to host e-mail accounts and provide basic e-mail access.
The POP3 service performs the tasks of message download and request handling on a Windows-based server, where message download consists of transmitting the messages from a folder in the file system to clients and request handling is performed according to the POP3 protocol, which defines how the server responds to requests sent from an e-mail client.
Basic e-mail access depends on the POP3 protocol implemented by the POP3 service..

The POP3 service is a scaleable e-mail server that meets the basic e-mail access needs of businesses and organizations. Any client that supports RFC 1939 can be configured to use the POP3 service to download e-mail. The POP3 service supports the following client authentication methods: Active Directory integrated authentication, Local Windows accounts authentication, and Encrypted Password file authentication. Because the POP3 service mail store exists in the file system, message storage and retrieval do not require specific database technologies on the server.

Some more applications of POP3

The POP3 service is designed to meet the needs of businesses and organizations with a small number of users needing basic e-mail access. However, the POP3 service is capable of scaling up to meet the needs of Internet service providers (ISPs) offering basic e-mail access to a large number of users.

Small Businesses and Organizations

Many small businesses have an investment in a Windows Server infrastructure that is suitable for an e-mail platform like POP3. POP3 enables organizations to host e-mail services for their Internet domain and support basic e-mail service. E-mail clients receive e-mail from the POP3 server and send e-mail by using the SMTP service. Groupware and scheduling functionality are not available with the POP3 service.

Internet Service Providers

ISPs offer e-mail service to customers as an added incentive; therefore the cost of e-mail service must be low enough that it does not impact the bottom line. Because POP3 use is so widespread, system administrators are familiar with existing POP3-compatible technology. The POP3 service in Windows Server 2003 supports a variety of authentication methods and scales up well to meet the demands of the ISP. Additionally, ISPs with large user bases can take advantage of the Active Directory directory service to store account information.

Some other new alternatives to POP3

POP3 is not the only protocol to support message retrieval. Internet Message Application Protocol version 4 (IMAP4) is similar in purpose to POP3. The IMAP4 protocol defines an extensive command set, including message searching, downloading of individual message body parts, and message tagging. IMAP4 implementations are not as common as POP3 because of the complexity of the IMAP4 protocol. Consequently, the use of IMAP4 is not as widespread as POP3 although the use of IMAP4 is growing as users demand more features from ISPs..

Messaging API (MAPI) is a protocol used with Microsoft Exchange Server and various e-mail clients for feature-rich messaging, including groupware and scheduling. One of the capabilities of MAPI is message retrieval for clients, such as Microsoft Outlook 2003. MAPI provides access to messages and folders and the associated properties of messaging objects. Message retrieval occurs in the context of a remote procedure call (RPC) from the client to the Exchange Server store, with the message contents transmitted in the response to the RPC call.

References:

1.www.hoestuffswork.com
2.www.verizononline.con
3.www. searchwebservices.techtarget.com
4. www.webopedia.com

The Basic Concepts of the Text Editor

Anupam Singh

TextEditor

->A Computer Program that allows the user to create and revise the target document.

Document:

- **Program(C editor,VC++ Editor)**
- **Text(notepad , MSWord)**
- **Equations(Excel)**
- **Lineart(MSWord,Paint)**
- **Table(MSWord,Excel)**
- **Diagram(Paint,Flash,Microsoft journel)**
- **Photo(Photoshop)**

Line Based Editors

The line based editor presents a single line for editing. Each line may be called up in turn. The left and right cursor keys may be used to edit the line of text. Standard facilities exist to perform search/find/replace/move items of text. These types of editors are cheap, provide a basic set of functions,

and are reasonably small in terms of code size. A limited number of commands are offered that people can quickly learn. Examples of line-based editors are edlin and vi.

Screen Based Editors

These provide a range of enhanced features, and editing is performed using the whole screen (multiple lines are shown at a time). The cursor may be moved in any of the four directions. Provision is made for scrolling the text when the cursor exceeds the boundary of the display window. Screen based editors are normally written for specific types of computers, and thus tend to be more costly, but provide a greater range of facilities, and can be more readily customized for particular applications (such as standard mail/form generation).

The screen editors may be command driven, where all functions to be performed are entered as commands on a single command line, or may be key generated, where pressing certain keys perform the desired function.

Text Editor

->Basically have four tasks

1.Select the part of the document to be viewed and manipulated.

2.Detemine how to format this view online and how to display it.

3.Specify and execute operations that modify the target document.

4.Update the view appropriately.

1.Selection Part of the document:

a.Travelling

-> Traversing thorugh the document.

b.Searching

->searching for a particular text.(next screenful,bottm & find pattern)

c.Filtering

->after selecting the taget text it has to be filtered.

d.Formatting

->Determines how the result of filtering will be seen as a visible representation.

Editing functions are often specialized to operate on elements meaningful to the type of editor.

Eg:1.Manuscript oriented editor operate on elements such as Single character,lines,words,sentences & paragraphs.

2.Program oriented editor might operate on elements such as identifiers,Keywords,and statements.

Editor is a userinterface which is concerned with input devices,outputdevices,and the interaction languages of the system.

Input Devices:->used to enter the text.

To enter commands.

Divided into 3 categories:

1.Text device(QWERTY keys)

2.Button Device(Function keys)

3.Locator Device(Mouse and data tablet)

4.Voice input device.(mike)

Output devices:

1.Teletypewriter.

2.CRT

Interaction Languages:

1.Typing Oriented or Text command oriented.(Command prompt)

2.Menu oriented(VB,VC++,MSWORD)

- BellLabsFamily editors are centered around the use of RegularExpressions and may include both FullScreen and LineEditors??
- BriefFamily editors were popular in the MSDOS days.
- DecFamily of text editors are popular on VAX/VMS and OpenVMS systems
- CuaFamily editors pioneered the use of CUA (common user access) style of editing now commonplace in MsWindows?? yet was present in mainframe editors such as the XeditEditor. The category of MicrosoftWindowsEditors or MsDosEditors is probably more specific.

- Mainframe
  - CuaFamily

    IbmEditorFamily
  - Non Cua Family

- PC Editors
    - MicrosoftWindowsEditors
    - BriefFamily
    - CuaFamily
        - IbmEditorFamily
    - MsDosEditors
    - WordStarFamily
- UnixEditors??
    - BellLabsFamily
    - EmacsFamily
    - ViFamily
- DecFamily
- SpecializedFamily

Cua (Common User Access)Family appears several times because it is a major feature of certain text editors. Also, families such as ViFamily and

EmacsFamily appear under unix because that is the originating family. Clearly, there are versions for nearly every platform out there...

I've also been asked to add a category for TweeEditors?? (I'm guessing Twee is Three?) which are very small editors. I put a category of TinyEditors out there already....

From the unofficial Twee Editors page: http://www.modest-proposals.com/Twee.htm

A 'twee' editor is one that is only a few multiples of the minimum size for a functional editor, without compression.

These are not really text editor families. They are families of software that can produce plain text but are actually meant for something else:

- WordProcessor
- IDE

IbmEditorFamily

These are TextEditors that grew out of IBM products and/or research or seem to be in some way influenced by that style of editor. If you're looking for just IBM PC (or simply MSDOS) editors, try MsDosEditors or MicrosoftWindowsEditors.

These include PageMode editors like IspfPdf? and XeditEditor.

metapad

Purpose: Fast, small, powerful replacement for Windows Notepad

Platforms: Windows 9x/NT/XP

Author: Alexander Davidson

Source: http://liquidninja.com/metapad/

Cost: free

Current Version: 3.5 (may be the last)

Size of Executable: 95744 bytes

Source Code Available: no

Documentation: fair to good (get the FAQ from www page)

The Next Generation of Supercomputers

Anupam Singh

The team from Sandia National Laboratories and Intel Corp.'s Tflops supercomputer burst through the seemingly impossible speed barrier of 1.3 trillion calculations per second. Their friendly competitors at Cray Research quickly announced their supercomputer had duplicated the feat.

The speed barrier researchers are racing to break through isn't caused by any design limitation of today's machines, but by the materials used in computer circuits.

Tflops, for example, consists of 76 large cabinets (left) sprawling over 1600 sq. ft. Inside these cabinets are 9072 Pentium Pro processors. The consensus among industry experts is that this sort of silicon-based technology will hit the wall sometime between 2010 to 2015.

Three new technologies are in the running to replace silicon-based supercomputing. One of the new classes of machines–the optical computer–uses light in place of circulating electrons, eliminating heating problems and the bottlenecks that develop at metallic connections. The second class of machines, called quantum computers, finds answers by taking advantage of a quirky law of quantum mechanics that says, in effect: Until someone peeks at it, a subatomic particle can be everywhere at once. The third, and perhaps strangest, type will use biological materials in place of silicon. These so-called DNA computers will solve problems by following the same rules living organisms obey to pass genetic information from generation to generation.

Optical Computers

Optical computers will be faster because light moves through space and fiber optic cable quicker than electrons do through silicon-based circuits. And there is another powerful advantage. While today's computers work with single bits of information, optical computers will be able to process billions of bits at the same time. This will result in a huge increase in computing speed.

Recently researchers in USA have combined transistors and lasers in a new microchip they hope will simplify fiber optic communications. The technology may also lead to the development of such miniaturized optical devices as tiny lasers and implantable medical sensors.

Kristina Johnson, a computer scientist at the University of Colorado, in Boulder, has designed an optical processor that can identify cancer cells on a pap smear in a fraction of the time it takes a conventional computer to do the job.

Quantum Computers

Combining conventional information theory and quantum mechanics, "quantum computing" is beginning to tantalize theorists with the prospect of breaking through barriers that may be insurmountable with current technology.

Researchers at the National Institute of Standards and Technology, in Boulder, Colorado, have demonstrated that they can trap a beryllium ion electromagnetically. Laser pulses cool the ions. Then, when a combination of fast laser pulses excites the ion, it fluoresces, emitting photons. Lasers can also be used to change information stored as ions. In the quantum NOT gate, another laser pulse causes the spin of each ion to change. This has the same effect as performing what mathematicians call a Boolean Negation, a fundamental operation for all computers.

DNA Computers

Researchers at the University of Rochester, in New York, have taken a first step at incorporating organic components into computers by creating DNA logic gates. Electronic computers use logic gates to convert binary data into meaningful signals for performing operations. The DNA logic gates detect specific fragments of genetic blueprint as input and splice these together to form output.

The researchers note that one pound of DNA can store more information than all the electronic computers ever built, and that a DNA chip the size of a droplet would have the processing power of the best supercomputer currently available.

The never ending research goes on still in the top labs of research institutions worldwide as researchers keep racing against time to slog on the expedition to make the computer of next millennium. Whether it would be DNA based or optical oriented, the next generation supercomputer would be a phenomenon. We have to wait and watch.

A XML Fundas-A quick Tutorial

Anupam Singh

Why do we need XML?

Data-exchange

XML is used to aid the exchange of data. It makes it possible to define data in a clear way. Both the sending and the receiving party will use XML to understand the kind of data that's been sent. By using XML everybody knows that the same interpretation of the data is used

Replacement for EDI

EDI (Electronic Data Interchange) has been for several years the way to exchange data between businesses. EDI is expensive, it uses a dedicated communication infrastructure. And the definitions used are far from flexible. XML is a good replacement for EDI. It uses the Internet for the data exchange. And it's very flexible.

More possibilities

XML makes communication easy. It's a great tool for transactions between businesses. But it has much more possibilities. You can define other languages with XML. A good example is W ML (W ireless Markup Language), the language used in W APcommunications. W ML is just an XML dialect.

What is XML ?

Simpler SGML

XML is a meta-language. A meta-language is a language that's used to define other languages. You can use XML for instance to define a language like W ML. XML is a smaller version of SGML. It's easy to master and that's a major advantage compared to SGML which is a very complex meta-language.

XML: What it can do

With XML you can : Define data structures Make these structures platform independent Process XML defined data automatically Define your own tags

With XML you cannot Define how your data is shown. To show data, you need other techniques.

The general structure of XML

Define your own tags

In XML, you define your own tags. If you need a tag <TUTORIAL> or <STOCKRATE>, that's no problem.

DTD or Schema

If you want to use a tag, you'll have to define it's meaning. This definition is stored in a DTD (Document Type Definition). You can define your own DTD or use an existing one. Defining a DTD actually means defining a XML language. An alternative for a DTD is Schema.

Showing the results

Often it's not necessary to display the data in a XML document. It's for instance possible to store the data in a database right away. If you want to show the data, you can. XML itself is not capable of doing so. But XML documents can be made visible with the aid of a language that defines the presentation. XSL (eXtensible Stylesheet Language) is created for this purpose. But the presentation can also be defined with CSS (Cascading Style Sheets).

XML Tags

Tags

XML tags are created like HTML tags. There's a start tag and a closing tag. <TAG>content</TAG> The closing tag uses a slash after the opening bracket, just like in HTML. The text between the brackets is called an element.

Syntax

The following rules are used for using XML tags: Tags are case sensitive. The tag <TRAVEL> differs from the tags <Travel> and <travel> Starting tags always need a closing tag All tags must be nested properly Comments can be used like in HTML: <!-- Comments --> Between the starting tag and the end tag XML expects the content. <amount>135</amount> is a valid tag for an element amount that has the content 135

Empty tags

Besides a starting tag and a closing tag, you can use an empty tag. An empty tag does not have a closing tag. The syntax differs from HTML: <TAG/>

Elements and sub elements

Elements and children

With XML tags you define the type of data. But often data is more complex. It can consist of several parts. To describe the element car you can define the tags <car>mercedes</car>. This model might look like this: <car> <brand>volvo</brand> < type > v4 0 < / type > <color>green</color> </car>

Besides the element car three other elements are used: brand, type and color. Brand, type and color are sub-elements of the element car. In the XML-code the tags of the sub-elements are enclosed within the tags of the element car. Sub-elements are also called children

XML documents

The XML declaration

The first line of an XML document is the XML declaration. It's a special kind of tag: <?xml version="1.0"?> The version 1.0 is the actual version of XML. The XML declaration makes clear that we're talking XML and also which version is used. The version identification will become important after new versions of XML are used.

The root element

All XML documents must have a root element. All other elements in the same document are children of this root element. The root element is the top level of the structure in an XML document.

Structure of an XML page

<?xml version="1.0"?> <root> <element> <sub-element> co n t e n t </sub-element> <sub-element> co n t e n t </sub-element> element> </root> All elements must be nested. The level of nesting can be arbitrarily deep.

A real XML page

<?xml version="1.0"?> <sales> < shop > <number> 100 </number> <manager> Ray Bradbury </manager> </shop> <product> <name> carrots </name> <totalprice> 10 </totalprice> </product> </sales>

XML Attributes

Attributes

Elements in XML can use attributes. The syntax is: <element attribute-name = "attribute-value">....</element> The value of an attribute needs to be quoted, even if it contains only numbers. An example <car color = "green">volvo</car> The same information can also be defined without using attributes: <car> <brand>volvo</brand> <color>green</color> </car>

Avoid attributes

W hen possible try to avoid attributes. Data structures are more easy described in XML tags. Software that checks XML-documents can do a better job with tags than with attributes.

Well formed XML documents

Well formedness

An XML document needs to be well formed. W ell formed means that the document applies to the syntax rules for XML.

The rules

To be well formed a document needs to comply to the following rules: it contains a root element all other elements are children of the root element all elements are correctly paired the element name in a start-tag and an end-tag are exactly the same attribute names are used only once within the same element

Note

There are more rules, some of them have to do with entities. In this quick tutorial, entities are not covered.

Valid XML documents

Valid

To be of practical use, an XML document needs to be valid. To be valid an XML document needs to apply to the following rules: The document must be well formed. (More on well formed in the previous page). The document must apply to the rules as defined in a Document Type Definition (DTD), (More on DTD's in the next page) If a document is valid, it's clearly defined what the data in the document really means. There's no possibility to use a tag that's not defined in the DTD. Companies that exchange XML-documents can check them with the same DTD. Because a valid XML document is also well formed, there's no possibility for typo's in t h e t a g s.

Valid is about structure

A valid XML-document has a structure that's valid. That's the part you can check. There's no check for the content.

XML: the DTD

Defining the language

To use XML you need a DTD (Document Type Definition). A DTD contains the rules for a particular type of XML-documents. Actually it's the DD that defines the language.

Elements

A DTD describes elements. It uses the following syntax: The text <! ELEMENT, followed by the name of the element, followed by a description of the element. For instance: <!ELEMENT brand (#PCDATA)> This DTD description defines the XML tag <brand>.

Data

The description (#PCDATA) stands for parsed character data. It's the tag that is shown and also will be parsed (interpreted) by the program that reads the XML document. You can also define (#CDATA), this stands for character data. CDATA will not be parsed or shown.

Sub elements

An element that contains sub elements is described thus: <!ELEMENT car (brand, type) > <!ELEMENT brand (#PCDATA) > <!ELEMENT type (#PCDATA) > -16This means that the element car has two subtypes: brand and type. Each subtype can contain characters.

Number of sub elements

If you use <!ELEMENT car (brand, type) >, the sub elements brand and type can occur once inside the element car. To change the number of possible occurrences the following indications can be used: + must occur at least one time but may occur more often * may occur more often but may also be omitted ? may occur once or not at all

The indications are used behind the sub element name. For instance: <!ELEMENT animal (color+) >

Making choices

With the sign '|' you define a choice between two sub elements. You enter the sign between the names of the sub elements. <!ELEMENT animal (wingsize|legsize) >

Empty elements

Empty elements get the description EMPTY. For instance <!ELEMENT separator EMPTY> that could define a separator line to be shown if the XML document appears in a browser.

DTD: external

A DTD can be an external document that's referred to. Such a DTD starts with the text <!DOCTYPE name of root-element SYSTEM "address"> The address is an URL that points to the DTD. -17In the XML document you make clear that you'll use this DTD with the line: <!DOCTYPE name of root-element SYSTEM "address"> that should be typed after the line <?xml version="1.0"?>

DTD: internal

A DTD can also be included in the XML document itself. After the line <?xml version="1.0"?> you must type <!DOCTYPE name of root-element [followed by the element definitions. The DTD part is closed with]>

Presenting XML documents

Showing XML documents

XML is about defining data. With XML you can define documents that are understood by computers. But to make these documents understandable to humans, you need to show them.

CSS

Cascading Style sheets (CSS offer possibilities to show XML. It works just like adding styles to HTML elements.

XSL

The preferred solution is using XSL (eXtensible Style sheet Language). XSL can convert XML documents into HTML. It can be used client side but the best solution is to use XSL server side. You can convert your XML documents to HTML, thus making them visible to any browser.

email your articles to anupam.vit@gmail.com

www.anupamsingh.cjb.net

technicks
your tech side