CSC 3380 Aymond

Section 1

2/19/2020

Homework Assignment
- Enterprise Architect Component Diagram, Due 11PM TONIGHT

Project News
- Next Milestone: #2
  - Due Friday 2/21, 11PM
  - Upload to Moodle (1 upload for entire team)
  - Outline is in Project Kickoff Lecture Notes
  - BE SURE TO UPDATE SECTIONS FROM MILESTONE #1
  - All UML diagrams must be developed in EA
- In-class Milestone #2 Presentations
  - Monday, March 2
  - Chanuka's teams: 1240 PFT
  - Clinton's teams: 1245 PFT
  - Qing's teams: 1258 PFT

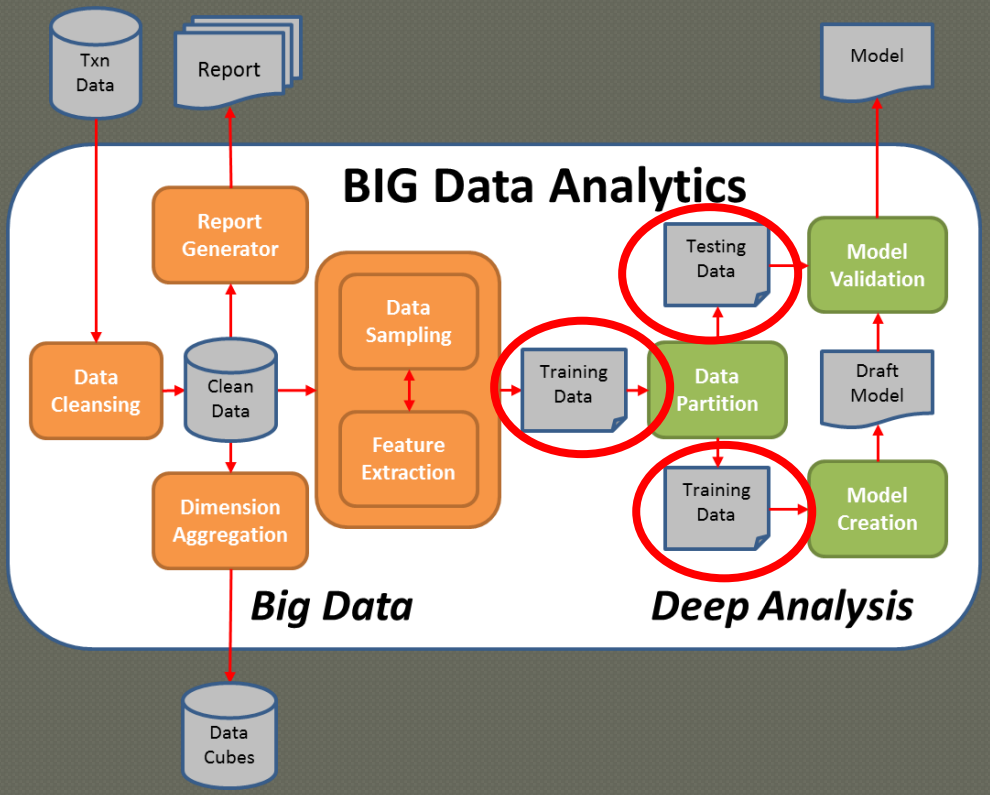
Midterm Exam
- Wednesday, March 4

Live Designing

https://www.jbklutse.com/using-word-count-tool/

Component Communication: The Great Data Exchange

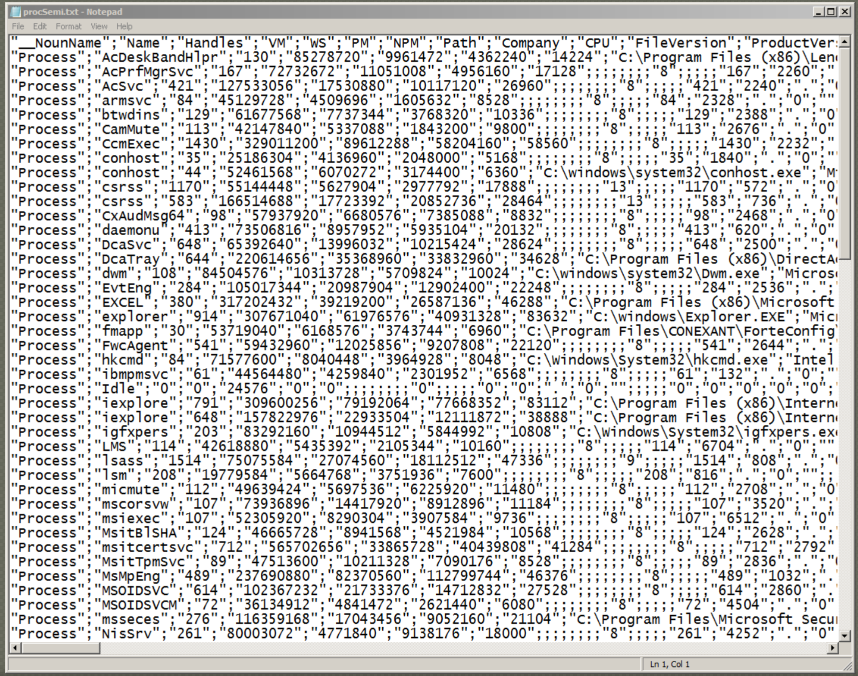
- Data exchange between components is often done via text files



http://horicky.blogspot.com/2012/08/big-data-analytics.html

Flat Files

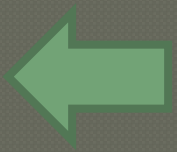
- Historically, component (and system) data exchange has been done via flat files
  - Text files of comma delimited data; each comma indicates the start of a new piece of data
  - The meaning of the data is determined by the position of the entry in the data file (row & “column”)
- This method is very fast and very efficient for the computer
- It is very slow and error-prone for the human
- Data files are often very large and looked a lot like this:



```
"__NounName";"Name";"Handles";"VM";"WS";"PM";"NPM";"Path";"Company";"CPU";"FileVersion";"ProductVer
"Process";"AcDeskBandHlpr";"130";"85278720";"9961472";"4362240";"14224";"C:\Program Files (x86)\Len
"Process";"AcPrfMgrSvc";"167";"72732672";"11051008";"4956160";"17128";;;;;;;;"8";;;;;"167";"2260";"
"Process";"AcSvc";"421";"127533056";"17530880";"10117120";"26960";;;;;;;;"8";;;;;"421";"2240";".";"
"Process";"armsvc";"84";"45129728";"4509696";"1605632";"8528";;;;;;;;"8";;;;;"84";"2328";".";"0";""
"Process";"btwdins";"129";"61677568";"7737344";"3768320";"10336";;;;;;;;"8";;;;;"129";"2388";".";"0
"Process";"CamMute";"113";"42147840";"5337088";"1843200";"9800";;;;;;;;"8";;;;;"113";"2676";".";"0"
"Process";"CcmExec";"1430";"329011200";"89612288";"58204160";"58560";;;;;;;;"8";;;;;"1430";"2232";"
"Process";"conhost";"35";"25186304";"4136960";"2048000";"5168";;;;;;;;"8";;;;;"35";"1840";".";"0";"
"Process";"conhost";"44";"52461568";"6070272";"3174400";"6360";"C:\windows\system32\conhost.exe";"M
"Process";"csrss";"1170";"55144448";"5627904";"2977792";"17888";;;;;;;;"13";;;;;"1170";"572";".";"0
"Process";"csrss";"583";"166514688";"17723392";"20852736";"28464";;;;;;;;"13";;;;;"583";"736";".";"
"Process";"CxAudMsg64";"98";"57937920";"6680576";"7385088";"8832";;;;;;;;"8";;;;;"98";"2468";".";"0
"Process";"daemonu";"413";"73506816";"8957952";"5935104";"20132";;;;;;;;"8";;;;;"413";"620";".";"0"
"Process";"DcaSvc";"648";"65392640";"13996032";"10215424";"28624";;;;;;;;"8";;;;;"648";"2500";".";"
"Process";"DcaTray";"644";"220614656";"35368960";"33832960";"34628";"C:\Program Files (x86)\DirectA
"Process";"dwm";"108";"84504576";"10313728";"5709824";"10024";"C:\windows\system32\Dwm.exe";"Micros
"Process";"EvtEng";"284";"105017344";"20987904";"12902400";"22248";;;;;;;;"8";;;;;"284";"2536";".";
"Process";"EXCEL";"380";"317202432";"39219200";"26587136";"46288";"C:\Program Files (x86)\Microsoft
"Process";"explorer";"914";"307671040";"61976576";"40931328";"83632";"C:\windows\Explorer.EXE";"Mic
"Process";"fmapp";"30";"53719040";"6168576";"3743744";"6960";"C:\Program Files\CONEXANT\ForteConfig
"Process";"FwcAgent";"541";"59432960";"12025856";"9207808";"22120";;;;;;;;"8";;;;;"541";"2644";".";
"Process";"hkcmd";"84";"71577600";"8040448";"3964928";"8048";"C:\Windows\System32\hkcmd.exe";"Intel
"Process";"ibmpmsvc";"61";"44564480";"4259840";"2301952";"6568";;;;;;;;"8";;;;;"61";"132";".";"0";"
"Process";"Idle";"0";"0";"24576";"0";"0";;;;;;;;"0";;;;;"0";"0";".";"0";"";;;;;"0";"0";"0";"0";"0";
"Process";"iexplore";"791";"309600256";"79192064";"77668352";"83112";"C:\Program Files (x86)\Intern
"Process";"iexplore";"648";"157822976";"22933504";"12111872";"38888";"C:\Program Files (x86)\Intern
"Process";"igfxpers";"203";"83292160";"10944512";"5844992";"10808";"C:\Windows\System32\igfxpers.ex
"Process";"LMS";"114";"42618880";"5435392";"2105344";"10160";;;;;;;;"8";;;;;"114";"6704";".";"0";""
"Process";"lsass";"1514";"75075584";"27074560";"18112512";"47336";;;;;;;;"9";;;;;"1514";"808";".";"
"Process";"lsm";"208";"19779584";"5664768";"3751936";"7600";;;;;;;;"8";;;;;"208";"816";".";"0";"";;
"Process";"micmute";"112";"49639424";"5697536";"6225920";"11480";;;;;;;;"8";;;;;"112";"2708";".";"0
"Process";"mscorsvw";"107";"73936896";"14417920";"8912896";"11184";;;;;;;;"8";;;;;"107";"3520";".";
"Process";"msiexec";"107";"52305920";"8290304";"3907584";"9736";;;;;;;;"8";;;;;"107";"6512";".";"0"
"Process";"MsitBlSHA";"124";"46665728";"8941568";"4521984";"10568";;;;;;;;"8";;;;;"124";"2628";".";
"Process";"msitcertsvc";"712";"565702656";"33865728";"40439808";"41284";;;;;;;;"8";;;;;"712";"2792"
"Process";"MsitTpmSvc";"89";"47513600";"10211328";"7090176";"8528";;;;;;;;"8";;;;;"89";"2836";".";"
"Process";"MsMpEng";"489";"237690880";"82370560";"112799744";"46376";;;;;;;;"8";;;;;"489";"1032";".
"Process";"MSOIDSVC";"614";"102367232";"21733376";"14712832";"27528";;;;;;;;"8";;;;;"614";"2860";".
"Process";"MSOIDSVCM";"72";"36134912";"4841472";"2621440";"6080";;;;;;;;"8";;;;;"72";"4504";".";"0"
"Process";"msseces";"276";"116359168";"17043456";"9052160";"21104";"C:\Program Files\Microsoft Secu
"Process";"NisSrv";"261";"80003072";"4771840";"9138176";"18000";;;;;;;;"8";;;;;"261";"4252";".";"0"
```

XML

- **XML**
- XSD
- XSLT



Markup Language

- A Markup language provides human readable formatting for machine readable files
- Markup languages provide hierarchical representations of data
  - Meaning organization of data
  - Eliminates redundancy that is needed for flat files

Extensible Markup Language (XML)

- XML is to data what HTML is to display
  - HTML provides a simply language to communicate display via text markup
  - XML Provides a simply language to communicate data via text markup
- Information is contained in elements and attributes

Extensible Markup Language (XML)

- Elements
  - Delimit data sets
  - Indicate the type of data
  - Are defined by the user, not by the standard
    - In other words, we can make the elements whatever we want them to be

```
<SimpleProgram>
   <FamousOutput>Hello, world!</FamousOutput >
</SimpleProgram>
```

XML Documents

- XML documents contain one giant XML element
- The outermost element of the file is called the root element, and is often named "root"
- XML version on the first line of the document is optional, but a good practice

```
<?xml version = "1.0"?>
<customerInfo>
   <name>Mickey Mouse</name>
   <company>Walt Disney World</company>
   <location>
      <street>500 Main Street</street>
      <cityState>
         Orlando, FL 12345
      </cityState>
   </location>
   <phone>(011) 123-4567</phone>
</customerInfo >
```

Attributes

- An attribute specifies a property of an element
- Unlike elements, text values must be enclosed in quotes
- Sub-elements can be repeated within an element, but attributes can only be specified once for the element
- An element can have more than one attribute
- Attributes appear inside the opening element tag

```
<location lat = “28.3852° N” long = “81.5639° W”>
   <street>500 Main Street</street>
   <cityState>
      Orlando, FL 12345
   </cityState>
</location>
```

Disadvantages of XML

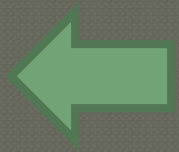
- Parsing XML is much slower than parsing a flat file
- The hierarchical nature of XML does not map well to data that is truly flat
  - i.e., when all combinations of data values is required

Advantages of XML

- Human readable text
- Unnecessary to enter missing data
- Hierarchical nature of the data makes specifying relational information easy
- XML provides mechanisms for validating data and for transforming data

XML

- XML
- **XSD**
- XSLT



XML Schema Definition (XSD)

- An XML Schema Definition (XSD) specifies a validation framework for XML structures
  - What elements are required/optional
  - What datatypes must values be
  - What ranges or sets of values are acceptable?
- Is XML itself
  - Predefined elements

XSD Example

```
<?xml version = "1.0" encoding = "UTF-8"?>
<xs:schema xmlns:xs = "http://www.w3.org/2001/XMLSchema">
   <xs:element name = "contact">
      <xs:complexType>
         <xs:sequence>
            <xs:element name = "name" type = "xs:string" />
            <xs:element name = "company" type = "xs:string" />
            <xs:element name = "phone" type = "xs:int" />
         </xs:sequence>
      </xs:complexType>
   </xs:element>
</xs:schema>
```

https://www.tutorialspoint.com/xml/xml_schemas.htm

Elements

- **Simple Type**

```
<xs:element name = "phone_number" type = "xs:int" />
```

- **Complex Type**

```
<xs:element name = "Address">
   <xs:complexType>
      <xs:sequence>
         <xs:element name = "name" type = "xs:string" />
         <xs:element name = "company" type = "xs:string" />
         <xs:element name = "phone" type = "xs:int" />
      </xs:sequence>
   </xs:complexType>
</xs:element>
```

https://www.tutorialspoint.com/xml/xml_schemas.htm

Global Types

- We can define a single type in our XSD, which can be used by other references

```
<xs:element name = "AddressType">
   <xs:complexType>
      <xs:sequence>
         <xs:element name = "name" type = "xs:string" />
         <xs:element name = "company" type = "xs:string" />
      </xs:sequence>
   </xs:complexType>
</xs:element>

…

<xs:element name = "Address1">
   <xs:complexType>
      <xs:sequence>
         <xs:element name = "address" type = "AddressType" />
         <xs:element name = "phone1" type = "xs:int" />
      </xs:sequence>
   </xs:complexType>
</xs:element>
```

https://www.tutorialspoint.com/xml/xml_schemas.htm

Attributes

- Attributes in XSD provide extra information within an element
- Attributes have name and type property

```
<xs:attribute name = "x" type = "y"/>
```

Benefits of XSD

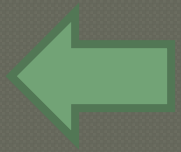
- Our program can validate an input file against an XSD before trying to parse it
- Our program can validate an output file to make sure what we've produced is valid
- Our data analysts can validate input files before providing them for execution
- The validation code provides information about where the problem may be occurring
- LOTS of tools are available
  - e.g., Notepad++ plugin

Benefits of XSD

- XSDs are used to specify classes for automated code generation
- XSDs can be exported from EA design documents
- XSDs can be imported into EA for automatic generation of class diagrams

XML

- XML
- XSD
- **XSLT**



eXtensible Stylesheet Language (XSLT)

- XSLT is a styling language for XML
- Specify commands from transforming information found in an XML file into a different text file format (e.g., HTML)
- Allows us to automatically generate files based on dynamic data

XML Code

```
<?xml version="1.0" encoding="UTF-8"?>
<catalog>
  <cd>
    <title>Empire Burlesque</title>
    <artist>Bob Dylan</artist>
    <country>USA</country>
    <company>Columbia</company>
    <price>10.90</price>
    <year>1985</year>
  </cd>
  <cd>
    <title>Hide your heart</title>
    <artist>Bonnie Tyler</artist>
    <country>UK</country>
    <company>CBS Records</company>
    <price>9.90</price>
    <year>1988</year>
  </cd>
```

…

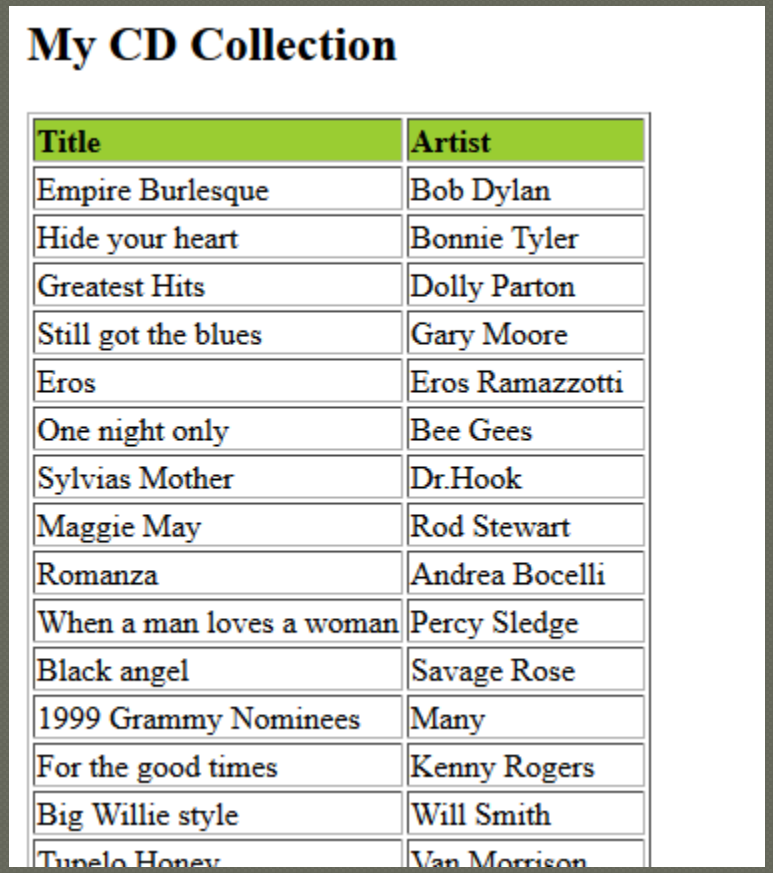
https://www.w3schools.com/xml/tryxslt.asp?xmlfile=cdcatalog&xsltfile=cdcatalog

XSLT Code

```
<?xml version="1.0" encoding="UTF-8"?>
<xsl:stylesheet version="1.0"
xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
<xsl:template match="/">
<html>
<body>
  <h2>My CD Collection</h2>
  <table border="1">
    <tr bgcolor="#9acd32">
      <th style="text-align:left">Title</th>
      <th style="text-align:left">Artist</th>
    </tr>
    <xsl:for-each select="catalog/cd">
    <tr>
      <td><xsl:value-of select="title"/></td>
      <td><xsl:value-of select="artist"/></td>
    </tr>
    </xsl:for-each>
  </table>
</body>
</html>
</xsl:template>
</xsl:stylesheet>
```

https://www.w3schools.com/xml/tryxslt.asp?xmlfile=cdcatalog&xsltfile=cdcatalog

HTML Generated



My CD Collection

Title	Artist
Empire Burlesque	Bob Dylan
Hide your heart	Bonnie Tyler
Greatest Hits	Dolly Parton
Still got the blues	Gary Moore
Eros	Eros Ramazzotti
One night only	Bee Gees
Sylvias Mother	Dr.Hook
Maggie May	Rod Stewart
Romanza	Andrea Bocelli
When a man loves a woman	Percy Sledge
Black angel	Savage Rose
1999 Grammy Nominees	Many
For the good times	Kenny Rogers
Big Willie style	Will Smith
Tupelo Honey	Van Morrison

https://www.w3schools.com/xml/tryxslt.asp?xmlfile=cdcatalog&xsltfile=cdcatalog