

A Simpler Schema

```
<element name="Book">
```

```
<complexType>
```

```
  <element name="author" type="xsd:string"/>
```

```
  <element name="preface" type="xsd:string"/>
```

```
  <element name="intro" type="xsd:string"/>
```

```
</complexType>
```

```
</e:Book>
```

Another Example Instance

<e:Book>

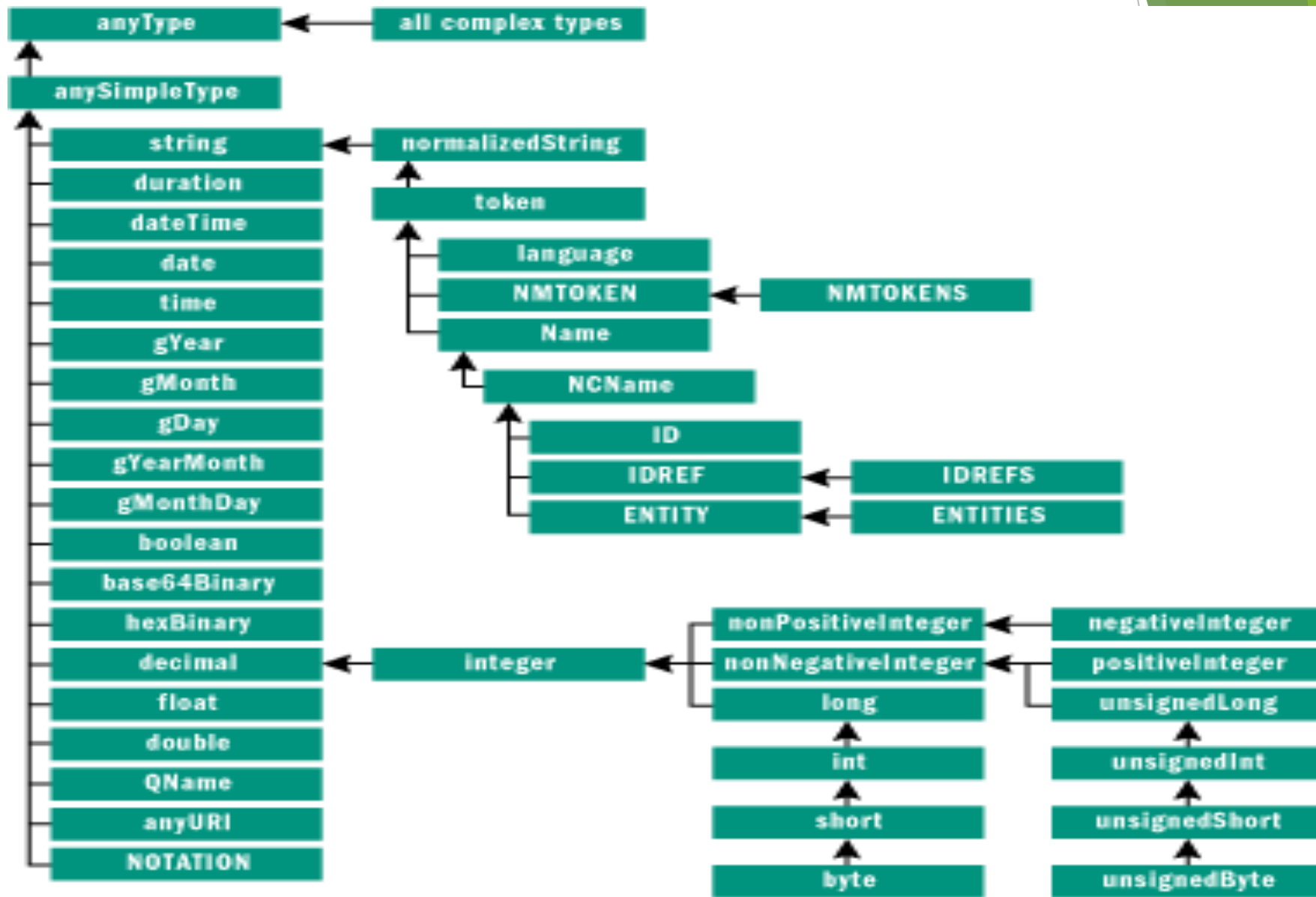
<author>Henry Ford</author>

<preface>Prefatory text</preface>

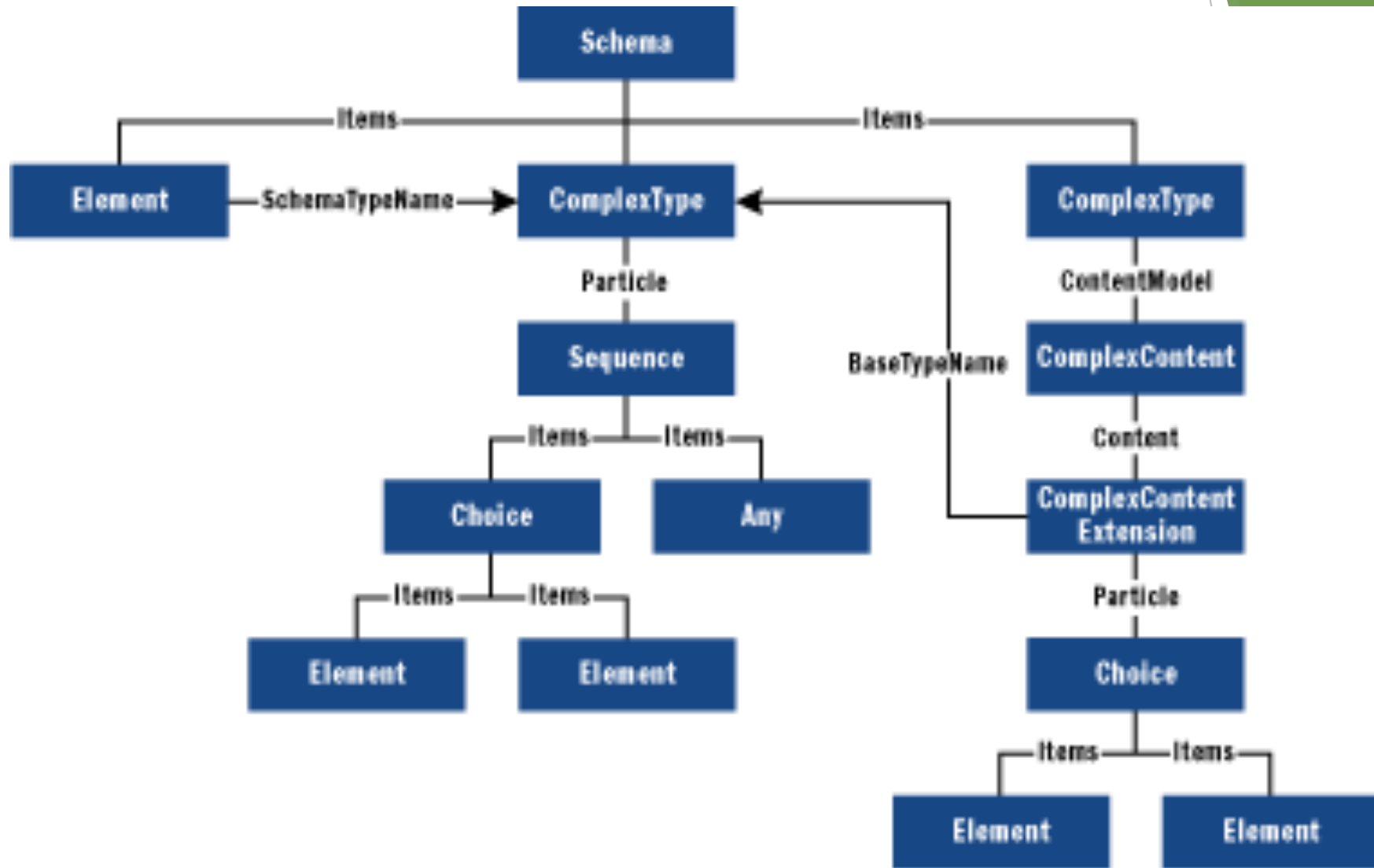
<intro>This is a book.</intro>

</e:Book>

XML Schema Defined Types



Class Library Data Hierarchy



Reading in XML Data

```
XmlReader reader  
    = new  
        XmlTextReader("http://foo.com/don.xsd");  
XmlSchema schema = XmlSchema.Load(reader, null);  
schema.Compile(null); // turn xml into objects  
reader.Close();
```

ALL Inter process Communication via SOAP

- ▶ ALL Interprocess communication (across network or on same machine) is through SOAP
 - ▶ Simple Object Access Protocol
 - ▶ It's a way of exchanging data and even calling other methods/threads, all via XML and plain old HTTP requests

Example SOAP Request

POST /StockQuote HTTP/1.1

Host: www.stockquoteserver.com

Content-Type: text/xml; charset="utf-8"

Content-Length: nnnn

SOAPAction: "Some-URI"

<SOAP-ENV:Envelope

xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"

SOAP-ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">

<SOAP-ENV:Body>

<m:GetLastTradePrice xmlns:m="Some-URI">

<symbol>DIS</symbol>

</m:GetLastTradePrice>

</SOAP-ENV:Body>

</SOAP-ENV:Envelope>

Example SOAP Response

HTTP/1.1 200 OK

Content-Type: text/xml; charset="utf-8"

Content-Length: nnnn

<SOAP-ENV:Envelope

xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"

SOAP-

ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" />

<SOAP-ENV:Body>

<m:GetLastTradePriceResponse xmlns:m="Some-URI">

<Price>34.5</Price>

</m:GetLastTradePriceResponse>

</SOAP-ENV:Body>

</SOAP-ENV:Envelope>

ASP.NET

- ▶ ASP =>
Active Server Pages
 - ▶ Put most of the computation in the server
- ▶ Very simple model to use
- ▶ ADO.NET is the database connection part



November						
S	M	T	W	T	F	S
28	29	30	31	1	2	3
4	5	6	7	8	9	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28	29	30	1
2	3	4	5	6	7	8

<asp:Calendar runat="server" />

Calling Web Services

- ▶ Any class can be converted into an XML Web Service with just a few lines of code, and can be called by any SOAP client.

```
<?xml version="1.0" encoding="utf-8" ?>
- <PetOrder xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-
  Instance" xmlns="http://tempuri.org/">
  <OrderId>1</OrderId>
  <OrderStatus>P</OrderStatus>
  <OrderDate>Oct 19 2001 6:19PM</OrderDate>
  <ShipToAddress>901 San Antonio Road</ShipToAddress>
  <ShipToCity>Palo Alto</ShipToCity>
  <ShipToState>California</ShipToState>
  <ShipToPostalCode>94303</ShipToPostalCode>
  <TotalPrice>155</TotalPrice>
- <LineItems>
  - <PetOrderLineItem>
    <LineNum>1</LineNum>
    <Name>EST-20</Name>
    <Qty>1</Qty>
    <Price>155.29</Price>
  </PetOrderLineItem>
</LineItems>
</PetOrder>
```

Take-away lessons

- ▶ VM's *are* important
 - ▶ Even Microsoft thinks so
- ▶ Distributed apps are important, but to do so requires standard protocols
 - ▶ Ways of serializing data
 - ▶ Ways of doing RPC

Limitations of the .NET Framework

- ▶ What if you're *not* on the network?
 - ▶ Maybe that's not an issue?
- ▶ Mapping between XML and any object is hard
 - ▶ Any object is controlled by compiler.
XML can be written by anybody with a text editor.
 - ▶ There's a whole bunch of class support for modified serializers and compilers