

XML چیست و چرا دارای اهمیت فراوان است ؟ (بخش سوم)

در **بخش اول**، مفاهیم اولیه تکنولوژی XML بررسی و پس از معرفی خانواده بزرگ تکنولوژی های مرتبط با XML، سه استاندارد لازم بمنظور تعریف نوع سندهای XML تشریح گردید (XML، Namespace و XML-Schema) و یا (DTD) . در **بخش دوم** به بررسی نحوه انتشار سندهای XML اشاره گردید. در این بخش، نحوه پردازش سندهای XML و دلایل توجه جدی به این تکنولوژی، تشریح می گردد.

پیاده کنندگان نرم افزار در زمان استفاده از XML علاقه مند به کار در دو سطح متفاوت می باشند: پردازش های سطح پائین و پردازش های سطح بالا. در پردازش های سطح پائین، تمام سند خوانده شده و به بخش های مورد نظر برای پردازش، تقسیم می گردد. در پردازش های سطح بالا، از ابزارهای لازم برای انجام فعالیت های سطح پائین استفاده شده و در برخی حالات نیز لازم است که کدها و اسکریپت های خاصی نوشته شده تا پردازش های لازم بر روی یک سند XML صورت پذیرد. برای پردازش های سطح پائین، می بایست از استانداردهای موجود در رابطه با Parsing استفاده کرد. در پردازش های سطح بالا، می بایست از استاندارد تبدیل XML استفاده کرد.

پارسر (Parser)

هر پردازنده XML (مرورگرها، پردازنده های Schema، ویرایشگرها، پردازنده های Stylesheet) دارای یک پارسر می باشند. پارسر یک سند XML را خوانده و آن را به بخش های متفاوت تقسیم و در حافظه مستقر می نماید. (معیار تقسیم بندی می تواند بر اساس المان و یا خصلت باشد). پس از استقرار بخش های متفاوت یک سند XML در حافظه، امکان انجام عملیات بر روی هر یک از بخش های موجود توسط پردازنده فراهم می گردد. (نظیر داده های موجود در یک بانک اطلاعاتی) :

- انتقال یک سند از یک فرمت به فرمت دیگر
- بازسازی مجدد یک سند XML با توجه به اولویت های اطلاعاتی مورد نظر
- اعمال فرمت های دلخواه بمنظور چاپ و یا نمایش

اکثر پیاده کنندگان نرم افزار های مبتنی بر XML ، ضرورتی به فراگیری جزئیات مربوط به نحوه پارسینگ ، نداشته و می توانند آن را به ابزارهای ارائه شده توسط تولیدکنندگان واگذار نمایند . (در محیط های پیاده سازی امکانات مربوطه در این زمینه پیش بینی می گردد). با مطالعه پارسر های XML ، شناخت مناسبی نسبت به امکانات و ویژگی های موجود، ایجاد خواهد شد . XML دارای دو اینترفیس برنامه نویسی DOM و SAX است .

DOM (Model Document Object) بر خلاف نام خود یک مدل واقعی نیست . DOM یک API (اینترفیس برنامه نویسی) برای نوشتن کدهای لازم بمنظور انجام عملیات دلخواه بر روی بخش هایی از یک سند XML پس از استقرار آن در حافظه است . رفتار استاندارد فوق با اطلاعات موجود در یک سند XML ، استفاده از یک ساختار درختی برای ذخیره سازی اطلاعات ، بهمراه دستوراتی برای حرکت در طول درخت و بازیابی اطلاعات مورد نظر است . دستورات بر اساس یک مدل شی گراء عمل می نمایند. مهمترین مزیت پارسرهای سازگار با DOM ، ارائه استانداردهای لازم API بمنظور انجام عملیات دلخواه در رابطه با اطلاعات است .

SAX (Simple API for XML) ، بر اساس یک مدل شی گراء رفتار نمی نماید. در اینترفیس فوق ، از مجموعه ای رویداد استفاده می گردد . رویدادها، پس از خواندن یک سند XML توسط پارسر های سازگار با SAX ، با توجه به شرایط مربوطه ، فعال خواهند شد. یکی از دلایل مطرح شدن SAX با توجه به وجود DOM ، عدم امکان استفاده از دستورات DOM تا زمانی است که تمام سند در حافظه مستقر گردد . استفاده و بکارگیری از دستورات DOM صرفاً پس از استقرار تمام سند XML در حافظه امکان پذیر خواهد بود . بدیهی است با توجه به رویکرد عملیاتی فوق ، حجم بالای سندهای XML می تواند باعث بروز مسائل و مشکلاتی در این رابطه گردد . مثلاً " بمنظور دستیابی به یک المان خاص می بایست در انتظار استقرار تمام سند XML در حافظه گردید . (از دست دادن زمان و مصرف بیهوده حافظه) . با استفاده از تکنولوژی SAX می توان بسرعت یک سند را فعال و در ادامه المان های موجود را بمنظور استفاده از محتویات مورد نظر جستجو کرد .

قابلیت SAX ، یک پیش نیاز برای پارسرهای سازگار با DOM است ، چون یک پارسر DOM می بایست در ابتدا سند را خوانده و در ادامه از رویدادهای مشابه SAX بمنظور تشخیص المان ها استفاده و آنها را در حافظه ایجاد نماید . بعنوان جمع بندی می بایست به این نکته اشاره گردد که اغلب پارسرها امکانات مربوط به DOM و SAX را حمایت کرده و می توان از پارسر های مشابه بمنظور انجام هر یک از حالات پردازش استفاده کرد .

XSLT Transformation (تبدیل)

تبدیل ، فرآیندی است که با استفاده از XSLT ، یک سند XML به یک فرمت دیگر (XML و یا غیر XML) تغییر می

گردد . تبدیلات، بمنزله بلاک های ایجاد برنامه های مبتنی بر XML مطرح می باشند. با استفاده از تبدیلات ، امکان انجام عملیات دلخواه در رابطه با اطلاعات بصورت بهینه فراهم می گردد .

یک XSLT stylesheet ، شامل تمپلیت ها و الگوها است . تمپلیت ساختار تبدیل مورد نیاز را مشخص و الگوها ، تمپلیت را به ساختار موجود در سند ، نسبت دهی خواهند کرد. مدل پردازش XSLT ، بصورت تعریفی و خود بازگشتی است . پردازنده StyleSheet ، بر روی سند XML پارس شده ، حرکت و در هر گره ، تمپلیت را بکارگرفته تا بر اساس آن الگوهائی که واجد شرایط هستند، انتخاب گردند. دستورالعمل های موجود در تمپلیت ها ، قادر به مرتب سازی ، سازماندهی مجدد و سایر عملیات مورد نیاز بمنظور ایجاد خروجی مورد نظر می باشند . برنامه زیر یک تمپلیت ساده را نشان می دهد . فرض کنید سند XML دارای دو المان <author> مشابه زیر باشد .

مثال : یک سند ساده XML به همراه دو المان

```
<author>
  <firstname>Ali</firstname><lastname>Irani</surname>
</author>
<author>
  <firstname>Reza</firstname><lastname>Tehrani</surname>
</author>
```

فرض کنید StyleSheet ، شامل تمپلیتی با الگوی مورد نظر برای تطبیق المان های <author> بصورت زیر باشد :

مثال : تمپلیت برای تطبیق المان های <author>

```
<xsl:template match="author">
  <xsl:value-of select="lastname"/>, <xsl:value-of
select="firstname"/>
<BR/>
</xsl:template>
```

زمانیکه پردازنده stylesheet بر روی سند حرکت می نماید ، هر زمان که با یک المان جدید برخورد نماید ، بررسی آن بمنظور تطبیق با شرایط موجود در تمپلیت انجام خواهد شد. پردازنده در مواجهه با المان <author> متوجه می گردد ، که یک مورد

مطابقت وجود دارد. بدین ترتیب با تاکید بر تمپلیت تعریف شده، خروجی مورد نظر ایجاد خواهد شد. بدین ترتیب خروجی بصورت زیر خواهد بود. (ترکیب عبارات موجود در تمپلیت، دو المان `<xsl:value-of>` به همراه تگ `
`).

Irani, Ali
Tehrani, Reza

پردازش XSLT، یک مدل قدرتمند و بسیار مفید برای استفاده در برنامه های XML B2B است. در برنامه های فوق، تبدیلات متعددی صورت می پذیرد (از یک فرمت XSLT به فرمت دیگر). XSLT ابزاری مناسب بمنظور انجام عملیات فوق است.

خلاصه

استانداردهای اساسی بمنظور پردازش سندهای XML شامل SAX, DOM و XSLT می باشد.

- پارسر های استاندارد شده امکان مبادله پردازنده های XML را فراهم می نمایند.
- تبدیلات، بعنوان بلاک های ایجاد برنامه های مبتنی بر XML مطرح می باشند.

تاکنون با XML و استانداردهای اصلی آن آشنا شدیم. در ادامه، چهار دلیل اهمیت تکنولوژی XML اشاره تا تأکیدی مجدد بر تاثیر تکنولوژی فوق در روش های طراحی و پیاده سازی سیستم های اطلاعاتی باشد. تأثیری که فرآیند طراحی و پیاده سازی سیستم های نرم افزاری را دچار تحولات فراوانی خواهد کرد.

دلیل اول: مبادله اطلاعاتی فراگیر

مبادله اطلاعاتی فراگیر، فرایندی بمنظور تبادل اطلاعات بین سیستم ها و یا عناصر با استفاده از فرمت های از قبل تعریف شده پیام، بدون آگاهی از جزئیات سایر سیستم ها است. مفهوم تعریف فوق بدین منزله است که: " اگر شما یک پیام مشابه این را به آدرس من ارسال نمائی، من نیز یک پیام مشابه آن را به آدرس شما ارسال خواهم کرد. " بدین ترتیب مشکلی که طراحان و برنامه نویسان سالیان متمادی است با آن درگیر بوده و همواره بدنبال حل این مسئله مهم بوده اند، برطرف می گردد. سیستم ها با استفاده از یک فرمت قابل قبول اطلاعاتی، بیکدیگر مرتبط خواهند شد. پیام های مبتنی بر XML، مستقل از نرم افزار

پردازش کننده بوده و بدین ترتیب می توان سیستم های بیشتری را در شبکه درگیر و ماهیت این نوع از سیستم ها را تغییر داد. بدون نیاز به اعمال تغییر در مکانیزم پیام ها)

XML، سرعت و در مقیاسی گسترده بعنوان گزینه ای مناسب برای فعالیت های: B2B، پیام شی به شی، مطرح شده است. در سراسر دنیا، سازمانهایی که نیازمند مبادله اطلاعاتی با یکدیگر می باشند، بهم ملحق و کمیته های تجاری را ایجاد نموده اند. در کمیته های فوق زبانهای XML تعریف و بر اساس آنها داده ها مبادله می گردند. در چنین مواردی Schema مربوطه، بمنظور نشر بر روی وب سایت های اصلی و مرکزی قرار گرفته است. برای آگاهی بیشتر در رابطه با مفاهیم و کمیته های مربوطه می توانید سایت های <http://www.xml.org> و <http://www.biztalk.org> را مشاهده نمایید.

دلیل دوم: تشریح خودکار داده ها

یک سند XML شامل اسامی مربوط به اشیاء/ ستون /المان ها بوده و بصورت اختیاری ممکن است شامل Schema و یا مراجعه به آن باشد. بدین ترتیب یک سند XML بنوعی خود را تشریح می نماید، روشی که تاکنون در این سطح استفاده نشده است. بدین ترتیب امکان پردازش های سمت سرویس گیرنده بر روی سندهای XML فراهم خواهد شد. زمانیکه یک سند XML برای یک دستگاه ارسال می گردد، نرم افزار موجود بر روی دستگاه قادر به نمایش و پردازش اطلاعات خواهد بود. با ترکیب داده هائی که خود را تشریح می نمایند و پردازش های سمت سرویس گیرنده، می توان برنامه های قدرتمندی را ایجاد و شاهد نسل جدیدی از سندهای هوشمند با برنامه نویسی اندک، باشیم.

دلیل سوم: استاندارد جهانی

XML یک استاندارد جهانی بوده و بدین ترتیب صلابت و گسترش آن تضمین شده است. بدین ترتیب می توان:

- بسادگی اطلاعات مورد نظر را بین سیستم های ناهمگن مبادله کرد.
- ابزارهای تولید شده رقابتی را بر اساس نیاز انتخاب کرد.
- روتین هائی را با قابلیت استفاده مجدد ایجاد کرد.
- ویژگی جدیدی را بدون درگیر شدن با مسائل سازگاری اضافه کرد.

دلیل چهارم : ابزاری مناسب برای ارائه اطلاعات

دلیل نهائی در رابطه با اهمیت XML ، قابلیت استفاده XML برای انواع متفاوت اطلاعات با اهداف گوناگون است . XML در ابتدا از SGML بمنظور استفاده در وب طراحی گردید ولی هم اکنون ، زمینه استفاده از آن در وب و سایر برنامه ها ئی که تحت وب نمی باشند ، نیز فراهم شده است . مبادله اطلاعات ، پیوستگی سیستم ها ، پردازش های توزیع شده ، ذخیره سازی محلی متادیتا ، نمونه ها ئی از میدان وسیع استفاده از XML می باشند . علاوه بر موارد فوق ، لازم است به این نکته مهم نیز توجه گردد که XML ، امکان استفاده از Unicode را نیز فراهم می نماید (بیش از ۶۴.۰۰۰ کارکتر) .