INSO
20892-5
1st. Edition
2016

جمهوری اسلامی ایران
Islamic Republic of Iran
سازمان ملی استاندارد ایران
Iranian National Standardization Organization

استاندارد ملی ایران
۲۰۸۹۲-۵
چاپ اول
۱۳۹۴

فناوری اطلاعات- زبان‌های تعریف طرحواره‌ی سند (DSDL) - قسمت ۵: انواع داده‌ای توسعه‌پذیر

Information technology — Document Schema Definition Languages (DSDL) — Part 5: Extensible Datatypes

ICS:35.240.30

سازمان ملی استاندارد ایران

تهران، ضلع جنوب غربی میدان ونک، خیابان ولیعصر، پلاک ۲۵۹۲

صندوق‌پستی: ۶۱۳۹-۱۴۱۵۵ تهران- ایران

تلفن: ۵-۸۸۸۷۹۴۶۱

دورنگار: ۸۸۸۸۷۰۸۰ و ۸۸۸۸۷۱۰۳

کرج ، شهر صنعتی، میدان استاندارد

صندوق پستی: ۱۶۳-۳۱۵۸۵ کرج - ایران

تلفن: ۸- ۳۲۸۰۶۰۳۱ (۰۲۶)

دورنگار: ۳۲۸۰۸۱۱۴ (۰۲۶)

رایانامه: standard@isiri.org.ir

وبگاه: http://www.isiri.org

Iranian National Standardization Organization (INSO)

No.1294 Valiasr Ave., South western corner of Vanak Sq., Tehran, Iran

P. O. Box: 14155-6139, Tehran, Iran

Tel: + 98 (21) 88879461-5

Fax: + 98 (21) 88887080, 88887103

Standard Square, Karaj, Iran

P.O. Box: 31585-163, Karaj, Iran

Tel: + 98 (26) 32806031-8

Fax: + 98 (26) 32808114

Email: standard@isiri.org.ir

Website: http://www.isiri.org

به نام خدا

آشنایی با سازمان ملی استاندارد ایران

سازمان ملی استاندارد ایران به موجب بند یک مادهٔ ۳ قانون اصلاح قوانین و مقررات مؤسسهٔ استاندارد و تحقیقات صنعتی ایران، مصوب بهمن ماه ۱۳۷۱ تنها مرجع رسمی کشور است که وظیفهٔ تعیین، تدوین و نشر استانداردهای ملی (رسمی) ایران را به عهده دارد.

تدوین استاندارد در حوزه‌های مختلف در کمیسیون‌های فنی مرکب از کارشناسان سازمان، صاحب‌نظران مراکز و مؤسسات علمی، پژوهشی، تولیدی و اقتصادی آگاه و مرتبط انجام می‌شود و کوششی همگام با مصالح ملی و با توجه به شرایط تولیدی، فناوری و تجاری است که از مشارکت آگاهانه و منصفانهٔ صاحبان حق و نفع، شامل تولیدکنندگان، مصرف‌کنندگان، صادرکنندگان و واردکنندگان، مراکز علمی و تخصصی، نهادها، سازمان‌های دولتی و غیردولتی حاصل می‌شود. پیش‌نویس استانداردهای ملی ایران برای نظرخواهی به مراجع ذی‌نفع و اعضای کمیسیون‌های مربوط ارسال می‌شود و پس از دریافت نظرها و پیشنهادها در کمیتهٔ ملی مرتبط با آن رشته طرح و در صورت تصویب، به عنوان استاندارد ملی (رسمی) ایران چاپ و منتشر می‌شود.

پیش‌نویس استانداردهایی که مؤسسات و سازمان‌های علاقه‌مند و ذی‌صلاح نیز با رعایت ضوابط تعیین شده تهیه می‌کنند درکمیته ملی طرح، بررسی و درصورت تصویب، به عنوان استاندارد ملی ایران چاپ و منتشر می‌شود. بدین ترتیب، استانداردهایی ملی تلقی می‌شود که بر اساس مقررات استاندارد ملی ایران شمارهٔ ۵ تدوین و در کمیتهٔ ملی استاندارد مربوط که در سازمان ملی استاندارد ایران تشکیل می‌شود به تصویب رسیده باشد.

سازمان ملی استاندارد ایران از اعضای اصلی سازمان بین‌المللی استاندارد (ISO)[1]، کمیسیون بین‌المللی الکتروتکنیک (IEC)[2] و سازمان بین‌المللی اندازه‌شناسی قانونی (OIML)[3] است و به عنوان تنها رابط[4] کمیسیون کدکس غذایی (CAC)[5] در کشور فعالیت می‌کند. در تدوین استانداردهای ملی ایران ضمن توجه به شرایط کلی و نیازمندی‌های خاص کشور، از آخرین پیشرفت‌های علمی، فنی و صنعتی جهان و استانداردهای بین‌المللی بهره‌گیری می‌شود.

سازمان ملی استاندارد ایران می‌تواند با رعایت موازین پیش‌بینی شده در قانون، برای حمایت از مصرف‌کنندگان، حفظ سلامت و ایمنی فردی و عمومی، حصول اطمینان از کیفیت محصولات و ملاحظات زیست‌محیطی و اقتصادی، اجرای بعضی از استانداردهای ملی ایران را برای محصولات تولیدی داخل کشور و/یا اقلام وارداتی، با تصویب شورای عالی استاندارد، اجباری کند. سازمان می‌تواند به منظور حفظ بازارهای بین‌المللی برای محصولات کشور، اجرای استاندارد کالاهای صادراتی و درجه‌بندی آن را اجباری کند. همچنین برای اطمینان بخشیدن به استفاده‌کنندگان از خدمات سازمان‌ها و مؤسسات فعال در زمینهٔ مشاوره، آموزش، بازرسی، ممیزی و صدورگواهی سیستم‌های مدیریت کیفیت و مدیریت زیست‌محیطی، آزمایشگاه‌ها و مراکز واسنجی (کالیبراسیون) وسایل سنجش، سازمان ملی استاندارد این‌گونه سازمان‌ها و مؤسسات را بر اساس ضوابط نظام تأیید صلاحیت ایران ارزیابی می‌کند و در صورت احراز شرایط لازم، گواهینامهٔ تأیید صلاحیت به آن‌ها اعطا و بر عملکرد آن‌ها نظارت می‌کند. ترویج دستگاه بین‌المللی یکاها، واسنجی وسایل سنجش، تعیین عیار فلزات گرانبها و انجام تحقیقات کاربردی برای ارتقای سطح استانداردهای ملی ایران از دیگر وظایف این سازمان است.

1- International Organization for Standardization
2- International Electrotechnical Commission
3- International Organization for Legal Metrology (Organisation Internationale de Metrologie Legals)
4- Contact point
5- Codex Alimentarius Commission

کمیسیون فنی تدوین استاندارد

« فناوری اطلاعات- زبان‌های تعریف طرحواره‌ی سند (DSDL) - قسمت ۵: انواع داده‌ای توسعه-پذیر»

رئیس:	**سمت و/یا محل اشتغال:**
محرم‌زاده، محمد (کارشناسی ارشد مهندسی مکاترونیک- اتوماتیک و کنترل تولید)	کارشناس اداره کل استاندارد استان آذربایجان شرقی
دبیر:	
محرم‌زاده، معصومه (کارشناسی ارشد مهندسی کامپیوتر- نرم‌افزار)	کارشناس شرکت صبا صنعت سیمای تبریز
اعضا: (اسامی به ترتیب حروف الفبا)	
بدری‌زاده، فریبا (کارشناسی مهندسی فناوری اطلاعات- شبکه)	کارشناس اداره کل استاندارد استان آذربایجان شرقی
بهری لاله، سپیده کارشناسی مهندسی فناوری اطلاعات- مخابرات)	کارشناس فناوری اطلاعات و ارتباطات اداره کل استاندارد استان آذربایجان شرقی
بی‌مانند، هدی (کارشناسی ارشد مهندسی کامپیوتر- نرم‌افزار)	کارشناس اداره کل استاندارد استان ایلام
تفسیری، حامد (کارشناسی مهندسی کامپیوتر- نرم‌افزار)	مسئول فناوری اطلاعات و ارتباطات اداره کل استاندارد آذربایجان شرقی
حکم‌آبادی، محمد شهاب (کارشناسی مهندسی کامپیوتر- نرم‌افزار)	کارشناس انفورماتیک شرکت شیرین عسل
شیخی، یونس (کارشناسی ارشد مهندسی برق- الکترونیک)	کارشناس اداره کل استاندارد استان آذربایجان شرقی
صدرالاشرافی، شهرزاد السادات (کارشناسی ارشد مهندسی فناوری الکترونیک)	مدیر کنترل کیفیت شرکت فجر الکتریک

اعضا: (اسامی به ترتیب حروف الفبا)	**سمت و/یا محل اشتغال:**
عباسی، ساناز (کارشناسی ارشد مهندسی کامپیوتر- معماری سیستم‌های کامپیوتری)	کارشناس مستقل
مهرشاد، بتول (کارشناسی ارشد MBE)	مسئول فناوری اطلاعات اداره‌کل استاندارد استان خراسان جنوبی
میرزایی، رضا (کارشناسی ارشد مهندسی مکاترونیک)	کارشناس شرکت صبا صنعت سیمای تبریز
نجار قره‌آغاج، یاشار کارشناسی مهندسی برق- الکترونیک)	کارشناس شرکت بازرسی آراد پایا کیفیت
یحیایی، سمیرا (کارشناسی ارشد مهندسی کامپیوتر-نرم‌افزار)	کارشناس فناوری اطلاعات و ارتباطات اداره کل استاندارد استان سمنان
ویراستار:	
بدلی، بابک (کارشناسی ارشد مهندسی کامپیوتر- نرم‌افزار)	معاون استانداردسازی و آموزش اداره کل استاندارد استان آذربایجان شرقی

فهرست مندرجات

عنوان **صفحه**

پیش‌گفتار

استاندارد « فناوری اطلاعات- زبان‌های تعریف طرحواره‌ی سند (DSDL) - قسمت ۵: انواع داده‌ای توسعه‌پذیر» که پیش‌نویس آن در کمیسیون‌های مربوط تهیه و تدوین شده است، در چهارصد ویکمین اجلاسیه کمیتهٔ ملی استاندارد فناوری اطلاعات مورخ ۱۳۹۴/۱۲/۱۲ تصویب شد. اینک این استاندارد به استناد بند یک مادهٔ ۳ قانون اصلاح قوانین و مقررات مؤسسه استاندارد و تحقیقات صنعتی ایران، مصوب بهمن ماه ۱۳۷۱، به عنوان استاندارد ملی ایران منتشر می‌شود.

استانداردهای ملی ایران بر اساس استاندارد ملی ایران شمارهٔ ۵ (استانداردهای ملی ایران- ساختار و شیوهٔ نگارش) تدوین می‌شوند. برای حفظ همگامی و هماهنگی با تحولات و پیشرفت‌های ملی و جهانی در زمینهٔ صنایع، علوم و خدمات، استانداردهای ملی ایران در صورت لزوم تجدیدنظر خواهند شد و هر پیشنهادی که برای اصلاح و تکمیل این استانداردها ارائه شود، هنگام تجدیدنظر در کمیسیون‌های مربوط مورد توجه قرار خواهد گرفت. بنابراین، باید همواره از آخرین تجدیدنظر استانداردهای ملی ایران استفاده کرد.

منبع و مأخذی که برای تهیه و تدوین این استاندارد مورد استفاده قرار گرفته به شرح زیر است:

ISO/IEC 19757-5:2011, Information technology — Document Schema Definition Languages (DSDL) — Part 5: Extensible Datatypes

فناوری اطلاعات- زبان‌های تعریف طرحواره‌ی سند (DSDL)- قسمت ۵: انواع داده‌ای توسعه‌پذیر

۱ هدف و دامنهٔ کاربرد

هدف از تدوین این استاندارد، تعیین زبان XML است تا کاربران بتوانند کتابخانه‌هایی را با اهداف خود ایجاد نمایند. تعاریف نوع داده‌ای در این کتابخانه‌ها می‌توانند توسط تصدیق کننده‌های[1] XML و دیگر ابزار تصدیق‌کننده محتوا و مقایسه بین مقادیر استفاده شوند.

۲ مراجع الزامی

در مراجع زیر ضوابطی وجود دارد که در متن این استاندارد به صورت الزامی به آن‌ها ارجاع داده شده است. بدین‌ترتیب، آن ضوابط جزئی از این استاندارد محسوب می‌شوند.

در صورتی که به مرجعی با ذکر تاریخ انتشار ارجاع داده شده باشد، اصلاحیه‌ها و تجدیدنظرهای بعدی آن برای این استاندارد الزام‌آور نیست. در مورد مراجعی که بدون ذکر تاریخ انتشار به آن‌ها ارجاع داده شده است، همواره آخرین تجدیدنظر و اصلاحیه‌های بعدی برای این استاندارد الزام‌آور است.

استفاده از مراجع زیر برای کاربرد این استاندارد الزامی است:

2-1 IETF RFC 3023, XML Media Types, Internet Standards Track Specification, January 2001, http://www.ietf.org/rfc/rfc3023.txt

2-2 IETF RFC 3987, Internationalized Resource Identifiers (IRIs), Internet Standards Track Specification, January 2005, http://www.ietf.org/rfc/rfc3987.txt

2-3 ISO/IEC 19757-2:2008, Information technology — Document Schema Definition Language (DSDL) — Part 2: Regular-grammar-based validation — RELAX NG

2-4 W3C XML, Extensible Markup Language (XML) 1.0 (Fourth Edition), W3C Recommendation, 16 August 2006, edited in place 29 September 2006, http://www.w3.org/TR/2006/REC-xml-20060816

2-5 W3C XML Names, Namespaces in XML 1.0 (Third Edition), W3C Recommendation, 8 December 2009, http://www.w3.org/TR/2009/REC-xml-names-20091208/

2-6 W3C XPath 2.0, XML Path Language (XPath) 2.0, W3C Recommendation, 23 January 2007, http://www.w3.org/TR/2007/REC-xpath20-20070123/

1 - Validators

2-7 W3C XPath 2.0 Functions, XQuery 1.0 and XPath 2.0 Functions and Operators, W3C Recommendation, 23 January 2007, http://www.w3.org/TR/2007/REC-xpath-functions-20070123/

2-8 W3C XSLT 2.0, XSL Transformations (XSLT) Version 2.0, W3C Recommendation, 23 January 2007, http://www.w3.org/TR/2007/REC-xslt20-20070123/

2-9 W3C XLink 1.0, XML Linking Language (XLink) Version 1.0, W3C Recommendation, 27 June 2001, http://www.w3.org/TR/2001/REC-xlink-20010627/

۳ اصطلاحات و تعاریف

در این استاندارد، اصطلاحات با تعاریف زیر به کار می‌رود:

۳-۱

مقدار کاندیدا

candidate value

برخی از داده‌های نویسه در یک سند XML که برای داشتن نوع داده آن، آزمون شده است.

۳-۲

نوع داده

datatype

ویژگی نام‌گذاری شده از توالی داده‌های نویسه‌ای است که در برابر تعاریف نوع داده‌ای توصیف شده در این استاندارد تصدیق می‌کند.

۳-۳

تعریف نوع داده

datatype definition

ویژگی رسمی از محدودیت‌ها بر داده‌های نویسه XML است که برای نوع داده تعریف شده است.

۳-۴

کتابخانه نوع داده

datatype library

مجموعه‌ای از تعاریف نوع داده‌ای که فضای نام[1] XML یکسان را به اشتراک می‌گذارد.

1 - Namespace

۵-۳

سند انواع داده‌ای توسعه‌پذیر

Extensible Datatypes document

سند XML که برای طرحواره‌ی الزامی معتبر است در این استاندارد ارائه شده و با قوانین استاندارد مطابق است.

۶-۳

حالت روبه جلو سازگار

forwards-compatible mode

حالت عملیاتی که یک پیاده‌سازی، ساختارهای زبانی را که با عنوان داشتن نسخه بعدی برچسب شده‌اند را نادیده می‌گیرد. در صورتی که این ساختارها به صورت صریح با عنوان پردازش مورد نیاز برچسب شده باشند، نادیده گرفته نمی‌شود.

۷-۳

پیاده‌سازی

implementation

پیاده‌سازی انواع داده‌ای توسعه‌پذیر که با این استاندارد تطابق دارد.

۸-۳

پیاده‌سازی توسعه یافته

extended implementation

پیاده‌سازی که با این استاندارد تطابق دارد و ارائه عملکردهای تکمیلی که توسط سازوکارهایی از انواع داده‌ای توسعه‌پذیر فراهم شده است.

۴ مرور کلی طرحواره‌ی انواع داده‌ای توسعه‌پذیر

طرحواره برای انواع داده‌ای توسعه‌پذیر به صورت بخش‌هایی در داخل شرح متن در این استاندارد پراکنده است و در یک پس‌زمینه خاکستری ارائه شده است. زبان طرحواره مورد استفاده، نحو[1] فشرده از RELAX NG است که به وسیله پیوست ت از استاندارد ISO/IEC 29500: 2008 تعریف شده است.

یادآوری ۱- در کل، بر حسب استاندارد ISO/IEC 19757-2: 2008 کلمات کلیدی نحو فشرده RELAX NG استفاده شده به عنوان شناسه‌ها در طرحواره، پیشوندی با نویسه "\" هستند.

1 - Syntax

یادآوری ۲- فضای نام تهی[1] به پیشوند "local" مقید شده است به طوری که می‌توان بعدا در طرحواره به آن ارجاع داد.

```
default namespace dt =
   "http://purl.oclc.org/dsdl/extensible-datatypes"
namespace local = ""
```

کتابخانه‌های انواع داده‌ای که در استاندارد ISO/IEC 19757-2: 2008 تعریف شده‌اند به وسیله یک IRI، با هر نوع داده در کتابخانه نوع داده، توسط یک NCName، شناسایی شده است. یک سند انواع داده‌ای توسعه پذیر یک یا چند کتابخانه نوع داده‌ای را برای پیاده‌سازی نشان می‌دهد. هر تعریف نوع داده یک نام واجد شرایط دارد؛ فضای نامی IRI اینکه کتابخانه نوع داده به کدام نوع داده‌ای تعلق دارد را شناسایی می‌کند و قسمت محلی نام نوع داده درون کتابخانه نوع داده را شناسایی می‌کند.

۵ ساختارهای عام

۵-۱ انواع عام

۵-۱-۱ عبارت‌های XPath

عبارت‌های W3C XPath 2.0 جهت انقیاد مقادیر به متغیرها یا ویژگی‌ها و نیز برای بیان آزمون‌ها در شرایط استفاده می‌شوند. پیاده‌سازی‌های انطباق از انواع داده‌ای توسعه‌پذیر باید به صورت تکمیلی توابع زیر از W3C XSLT 2. 0 را پیاده‌سازی کنند.

- document (W3C XSLT 2.0, Section 16.1)
- format – number (W3C XSLT 2.0, Section 16.4)
- function – available (W3C XSLT 2.0, Section 18.1.1)

این توابع باید در پیاده‌سازی‌ها با استفاده از نام فاقد شرایط توابع قابل فراخوانی باشند.

یادآوری- تابع function- available می‌تواند برای دسترس‌پذیری توابع توسعه XPath با اهداف اعتبار بهبود یافته[2] در پیاده‌سازی‌های توسعه‌یافته استفاده شوند.

```
XPath = text
```

گره محتوا برای ارزیابی عبارات XPath در انواع داده‌ای توسعه‌پذیر یک گره متنی است که تنها فرزند گره ریشه می‌باشد و مقدار گره، مقدار کاندیدای نرمال‌سازی شده با فضای خالی[3] است. موقعیت و اندازه متن هر دو، یک است. مجموعه‌ای از انقیادهای متغیر، متغیرهای درون دامنه هستند که در زیربند ۹-۴-۱ تعریف

1 - Null
2 - Enhance
3 - Whitespace

شده است. مجموعه‌ای از اعلان‌های فضای نامی برای عبارات درون دامنه‌ای هستند آنها برای عناصری که XPath ارائه می‌کند درون دامنه‌ای هستند.

۵-۱-۲ مقادیر بولی

جایی که مقدار بولی باید مشخص شود رشته‌های حرفی "true" و "false" استفاده می‌شوند.

```
Boolean = "true" | "false"
```

۵-۱-۳ عبارات منظم

عبارات منظم در W3C XPath 2.0 تعریف شده‌اند.

```
Regular-expression = text
```

۵-۱-۴ محتوای اختیاری[1]

سند انواع داده‌ای توسعه‌پذیر به وسیله یک طرحواره‌ی باز مدیریت می‌شوند که به منظور توسعه‌پذیری اجازه می‌دهد تا محتوای اختیاری در نقاط خاص رخ دهد. چنین محتوایی می‌تواند هر محتوای XML شامل عناصر یا صفات همراه با فضای نام XML انواع داده‌ای توسعه‌پذیر باشد.

```
anything =
 mixed {
  element * - dt:* {
   attribute * - dt:* { text }*,
   anything
  }*
 }
```

۵-۲ صفات رایج

۵-۲-۱ صفت version

مقدار صفت version، نسخه انواع داده‌ای توسعه‌پذیر را که عنصری در آن رخ می‌دهد، مشخص می‌کند. نسخه‌ای که در این استاندارد توصیف شده است "1.0" می‌باشد.

1 - Arbitrary

۵-۲-۲ صفت ns

مقدار صفت ns، فضای نام IRI از انواع داده‌ای که تعریف شده در عنصر داده‌ای که صفت name آن دارای پیشوند نیست، را مشخص می‌کند. بنابراین برای انواع داده‌ای وابسته، کتابخانه انواع داده تعیین می‌شود. این مقدار باید همانطوری که در IETF RFC 3987 تعریف شده است، یک IRI باشد.

```
ns = attribute ns {text}
```

۵-۲-۳ صفت name

صفت name، نام نوع داده، پارامتر، متغیر یا ویژگی را تعیین می‌کند. مقدار صفت name یک نام واجد شرایط می‌باشد. اگر هیچ پیشوندی مشخص نشده باشد، فضای نام IRI با وابستگی‌های نام به عنصری که صفت name رخ می‌دهد، می‌پیوندد. اگر صفت name در عنصر datatype رخ دهد، فضای نام IRI است که در صفت ns از عنصر datatype یا نزدیک‌ترین عنصر جد[1] آن یک صفت ns دارد در صورتی که یک فضای نام IRI وجود داشته باشد یا وجود نداشته باشد. در غیر این صورت نام بدون پیشوند فضای نام IRI ندارد.

یادآوری- اسامی با الگوی text تعریف شده توسط استاندارد ISO/IEC 29500-2:2008 مطابقت دارند و با اسامی که در ویژگی‌های طرحواره‌ی دیگر است کمتر محدود است.

```
name = attribute name {text}
```

۵-۲-۴ صفات توسعه

صفات توسعه، صفاتی در هر فضای نام غیرتهی از فضای نام انواع داده‌ای توسعه‌پذیر هستند. این صفات می‌توانند در هر عنصر انواع داده‌ای رخ دهند. حضور چنین صفاتی نباید رفتار عناصر انواع داده‌ای توسعه‌پذیر را که در این استاندارد تعریف شده است، تغییر دهد: یک پیاده‌سازی باید نتیجه یکسانی را برگرداند چه صفات توسعه پردازش شوند چه نشوند.

```
extension-attribute = attribute * - (local:* | dt:*) { text }
```

۵-۳ عناصر توسعه

عناصر توسعه، عناصری در هر فضای نام غیر از فضای نام انواع داده‌ای توسعه‌پذیر هستند. امکان دارد این عناصر توسط پیاده‌سازی‌های توسعه یافته پردازش شوند. سه کلاس از عنصر توسعه وجود دارد:

- عناصر توسعه سطح بالا که به عنوان فرزند عنصر سند یا عناصر div رخ می‌دهند.
- عناصر توسعه تعریف که به عنوان فرزند عناصر datatype رخ می‌دهند.
- عناصر توسعه انقیاد جایی که مقداری مقید می‌شود رخ می‌دهد. (به عنوان مثال به یک متغیر)

1 - Ancestor

```
extension-element =
  element * - dt:* {
    must-implement?,
    attribute * - ( dt:* | must-implement) { text }*,
    anything
  }
```

۵-۴ نسخه و سازگاری

یک عنصر انواع داده‌ای توسعه‌پذیر در حالت سازگار رو به جلو پردازش می‌شود در صورتی که آن عنصر با نزدیکترین جد آن دارای صفت version باشد، مقدار صفت version آن بزرگتر از "1.0" است. هنگامی که عنصری در فضای نام انواع داده‌ای توسعه‌پذیر توسط این استاندارد توصیف نشده باشد در حالت سازگار رو به جلو پردازش شده، صفات و فرزندان آن باید نادیده گرفته شوند مگر در صورتی که دارای صفت must-implement با مقدار true باشد که در این مورد اجرا باید متوقف شده و یک پیغام خطا دهد.

```
must-implement  =  attribute must-implement { boolean }
```

۶ ساده‌سازی

قبل از اینکه یک سند انواع داده‌ای توسعه‌پذیر برای اعتبارسازی به کار رود، به وسیله پردازش عناصر include و حل رویدادهای چندگانه از انواع داده‌ای با نام یکسان با یک تعریف واحد، به یک واحد منطقی منفرد ساده‌سازی می‌شوند.

۶-۱ عناصر include

عناصر include به کتابخانه‌های انواع داده‌ای توسعه‌پذیر دیگر اشاره می‌کند. آنها انواع داده‌ای را از این کتابخانه‌ها استخراج می‌کنند یا با استفاده از تعاریف موجود در سند میزبان آنها را دوباره تعریف می‌نمایند.

```
\include =
    element include {
    ns?,
    attribute href { text },
    extension-attribute*,
    top-level-element*
    }
```

صفت ns در include برای باطل کردن فضای نام انواع داده استخراجی به صورت تعریف شده با استفاده از صفت ns موجود در سند مشمول، استفاده شده است.

صفت href، مرجع IRI را مشخص می‌کند. این مرجع IRI ابتدا توسط نویسه‌های غیرمجاز باقی‌مانده[1] تبدیل می‌شود که این امر در بند ۵-۴ از W3C XLink 1. 0 مشخص شده است. در صورتی که این تبدیل قطعی نباشد، مرجع IRI به یک شکل قطعی تبدیل می‌شود که در بند ۵ از IETF RFC 3987 با استفاده از مبنای IRI از عنصر include توضیح داده شده است.

مقدار صفت href، همان‌گونه که در بند زیر آمده است برای ایجاد عنصر datatype استفاده می‌شود. مرجع IRI متشکل از خود IRI و یک شناسه چندپاره[2] اختیاری است. منبع شناسایی شده توسط IRI، بازیابی می شود. نتیجه هستار[3] MIME است: توالی از هشت‌تایی‌ها با یک نوع رسانه MIME بر چسب شده‌اند. نوع رسانه مشخص می‌کند که چگونه یک عنصر از هستار MIME و شناسه چندپاره اختیاری ساخته شده است.

هنگامی که نوع رسانه، application/xml یا text/xml باشد، هستار MIME باید به عنوان یک سند XML مطابق با RFC قابل اجرا (در زمان نوشتن [RFC 3023]) و یک عنصر تجزیه شود که باید یک عنصر datatype در فضای نام انواع داده توسعه‌پذیر باشد که از نتیجه تجزیه تشکیل شده است. خصوصا پارامتر charset باید به گونه‌ای که توسط RFC مشخص شده، به کار گرفته[4] شود. این ویژگی انجام انواع رسانه از Text/xml, application/xml را تعریف نمی‌کند. صفت href نباید شامل شناسه چندپاره باشد مگر اینکه که ثبت نوع رسانه مرجع توسط صفت شناسایی شود که صفت تفسیری از شناسه‌های چندپاره را برای آن نوع رسانه تعریف می‌کند.

یادآوری- مرجع [RFC 3023] تفسیری از شناسه‌های چندپاره برای application/xml یا text/xml را تعریف نمی‌کند.

عنصر datatype که توسط مقدار صفت href تعین شده بود، چنان که عناصر include آن حل شوند، پردازش می‌شود. این امر اجازه نمی‌دهد تا به حلقه وارد شود. به عبارت دیگر، عنصر datatype نباید به یک عنصر include غیرمرجع با یک صفت href با مقدار یکسان نیاز داشته باشد. این امر در تعدادی از تعاریف انواع داده‌ای نتیجه می‌دهد. در صورتی که عنصر include شامل هر عنصر datatype باشد در آن صورت هر نوع داده تعریف شده در عنصر include، باید یک تعریف نوع داده در کتابخانه ارجاع شده با نام یکسان باشد. تمامی تعاریف نوع داده ارجاع شده با نام یکسان به عنوان تعریف نوع داده در عنصر include، نادیده گرفته می‌شوند.

عنصر include مانند یک عنصر div با همان صفات بجز صفت href رفتار می‌کند. اولین فرزند از عنصر div معادل عنصر div دیگری است که صفات و فرزندان همانند عناصر datatype مرجع هستند به استثنای آن‌هایی که به وسیله تعریف include که در بالا تعریف شد لغو می‌شوند. بقیه فرزندان معادل div، فرزندان عنصر include می‌باشند.

1 - Escaping
2 -Fragment
3 - Entity
4 - Handled

۶-۲ انواع داده‌ای یکسان نامگذاری شده

```
\combine = attribute combine { "choice" | "all" }
```

اگر در نتیجه درج (همانطور که در زیربند ۶-۱ شرح داده شد) یا به روش دیگری دو یا چند تعریف نوع داده‌ای دارای نام واجد شرایط یکسانی باشند با هم ترکیب می‌شوند. برای هر نام، نباید بیش از یک عنصر datatype با نامی که صفت combine ندارد وجود داشته باشد. برای هر نامی در صورتی که عنصر datatype با صفت combine و با مقدار "choice" وجود داشته باشد در آن صورت باید هیچ عنصر datatype با آن نام که دارای صفت combine و مقدار "all" باشد وجود نداشته باشد. بنابراین برای هر نامی در صورتی که بیش از یک عنصر datatype با آن نام وجود داشته باشد در آن صورت یک مقدار منحصر بفرد برای صفت combine با آن نام وجود دارد. بعد از تعیین این مقدار منحصر بفرد، صفات combine حذف می‌شوند. اگر هر دو عنصر datatype دارای یک عنصر param با همان نام باشند، آن عناصر param باید همان نام و مقدار را مشخص کنند. تعاریف تا زمانی که تنها یک عنصر datatype برای هر نام وجود داشته باشد، با یکدیگر ترکیب می‌شوند.

مثال- دو تعریف زیر برای تشخیص رنگ است.

```
<datatype name="colour" combine="choice">
   <regex>#([0-9A-Fa-f]{2})([0-9A-Fa-f]{2})([0-9A-Fa-f]{2})</regex>
   <property name="red" type="hexByte" select="$_1"/>
   <property name="green" type="hexByte" select="$_2"/>
   <property name="blue" type="hexByte" select="$_3"/>
</datatype>

<datatype name="colour" combine="choice">
   <regex>#([0-9A-Fa-f])([0-9A-Fa-f])([0-9A-Fa-f])</regex>
   <property name="red" type="hexByte" select="concat($_1,$_1)"/>
   <property name="green" type="hexByte" select="concat($_2,$_2)"/>
   <property name="blue" type="hexByte" select="concat($_3,$_3)"/>
</datatype>
```

توسط فرآیند ترکیب، به یک تعریف واحد معادل با کد زیر ترکیب می‌شوند:

```
<datatype name="colour">
   <choice>
     <all>
       <regex>#([0-9A-Fa-f]{2})([0-9A-Fa-f]{2})([0-9A-Fa-f]{2})</regex>
       <property name="red" type="hexByte" select="$_1"/>
       <property name="green" type="hexByte" select="$_2"/>
       <property name="blue" type="hexByte" select="$_3"/>
```

```
    </all>
    <all>
      <regex>#([0-9A-Fa-f])([0-9A-Fa-f])([0-9A-Fa-f])</regex>
      <property name="red" type="hexByte" select="concat($_1,$_1)"/>
      <property name="green" type="hexByte" select="concat($_2,$_2)"/>
      <property name="blue" type="hexByte" select="concat($_3,$_3)"/>
    </all>
  </choice>
  </datatype>
```

اگر صفت choice، «all» را تشخیص دهد در آن صورت تمامی تعاریف به کار می‌رود بنابراین دو تعریف

```
<datatype name="pricing-currency" combine="all">
  <regex>[A-Z]{3}</regex>
</datatype>
<datatype name="pricing-currency" combine="all">
  <regex>[A-Z]{3}</regex>
  <property name="currency-code" value="$_0"/>
   <condition test="$currency-code='EUR' or $currency-code='USD'"/>
 </datatype>
```

هستند که توسط فرآیند ترکیب، به یک تعریف واحد معادل با کد زیر ترکیب می‌شوند:

```
<datatype name="pricing-currency">
  <all>
    <all>
      <regex>[A-Z]{3}</regex>
    </all>
    <all>
      <regex>[A-Z]{3}</regex>
      <property name="currency-code" value="$_0"/>
      <condition test="$currency-code='USD' or $currency-code='GBP'"/>
    </all>
  </all>
</datatype>
```

۷ عنصر سند

عنصر سند از یک سند انواع داده‌ای توسعه‌پذیر datatype است. این عنصر به یک صفت version (به زیربند ۵-۲-۱ مراجعه شود) و یک صفت ns اختیاری (به زیربند ۵-۲-۲ مراجعه شود) نیاز دارد.

```
start = \datatypes

\datatypes =
   element datatypes {
     version, ns?, extension-attribute*, top-level-element*
   }
version = attribute version { "1.0" }
```

۸ عناصر بالاترین سطح

عناصر بالاترین سطح به عنوان فرزندان عنصر سند وجود دارند.

```
top-level-element = \include | named-datatype | \div | extension-top-level-element
```

۸-۱ عنصر div

عناصر div برای افراز[1] یک کتاب‌خانه نوع داده استفاده می‌شوند.

یادآوری- استفاده از این عناصر معادل با عناصر div در استاندارد ISO/IEC 19757-2: 2008 می‌باشد.

```
\div =
   element div {
     ns?, version?, extension-attribute*, top-level-element*
   }
```

۸-۲ عناصر توسعه بالاترین سطح

عناصر توسعه بالاترین سطح می‌تواند برای نگهداری محتوایی که در کتاب‌خانه نوع داده (مانند فهرست کدهای استفاده شده برای آزمون مقادیر شمارش‌پذیر) ، مستندسازی، یا دیگر اطلاعاتی که توسط پیاده‌سازی‌های توسعه یافته استفاده می‌شوند، به کار رود. به عنوان مثال، یک عنصر توسعه سطح بالا می‌تواند توسط پیاده‌سازی توسعه یافته برای تعریف توابع توسعه استفاده شود (برای مثال استفاده از XSLT) که می‌تواند در عبارات XPath در کتابخانه نوع داده استفاده شود.

```
extension-top-level-element = extension-element
```

1 - Partition

با عناصر توسعه بالاترین سطح همانند سایر عناصر توسعه رفتار می‌شود (به زیربند ۵-۳ مراجعه شود).

۹ تعریف نوع داده

تعاریف نوع داده در کتاب‌خانه نوع داده می‌توانند با نام یا بدون نام باشند.

۹-۱ انواع داده نام‌گذاری شده

تعاریف انواع داده نام‌گذاری شده برای کتاب‌خانه نوع داده سطح بالا در سند کتاب‌خانه نوع داده با استفاده از عناصر datatype مشخص شده است. هر تعریف نوع داده نام‌گذاری شده یک نام مشخص در صفت name دارد که به صورت منحصر به فرد آن را شناسایی می‌کند (به زیربند ۵-۲-۳ مراجعه شود).

```
named-datatype =
   element datatype {
     name,
     ns?,
     preprocess?,
     combine?, extension-attribute*,
     param*,
     datatype-definition-element*
   }
```

۹-۲ انواع داده‌های بی‌نام

انواع داده‌های بی‌نام (به زیربند ۹-۴-۱-۵ مراجعه شود) برای تعریف انواع داده‌ای استفاده می‌شود که نمی‌توان با نام به آن‌ها مراجعه کرد.

```
anonymous-datatype =
   element datatype {
     preprocess?, extension-attribute*, datatype-definition-element*
   }
```

۹-۳ پردازش فضای خالی

صفت normalize-whitespace تعیین می‌کند چگونه یک مقدار کاندیدا قبل از اینکه در برابر عناصر تعریف انواع داده آزموده شود، فضای خالی نرمال‌سازی شده است. اگر normalize-whitespace دارای مقدار

preserve باشد هیچ فضای خالی نرمال‌سازی شده، رخ نخواهد داد. اگر normalize-whitespace دارای مقدار replace باشد تمامی نویسه‌های فضای خالی (فاصله‌ها، کلیدهای جهش[1]، خطوط جدید و سرسطر رفتن) توسط یک نویسه فاصله جایگزین می‌شود (U+0020). به عبارت دیگر (اگر normalize-whitespace دارای مقدار collapse باشد یا مشخص نشده باشد) فضای خالی پیش و پس حذف می‌شود و تمامی دنباله‌های داخلی از نویسه‌های فضای خالی توسط یک نویسه فاصله واحد جایگزین می‌شود.

```
preprocess =
   attribute normalize-whitespace { "preserve" | "replace" | "collapse" }
```

۹-۴ سازوکارهای تعریف انواع داده

یک تعریف نوع داده شامل تعدادی از عناصر است که مقادیر را مورد آزمون قرار می‌دهد و متغیرها و ویژگی‌ها را تعریف می‌کند. در صورتی که یک مقدار کاندیدا تابع محدودیت‌های مشخص شده توسط عناصر باشد در آن صورت یک مقدار معتبر برای نوع داده خواهد بود.

```
datatype-definition-element =
   property
   | variable
   | regex
   | \list
   | condition
   | valid
   | except
   | choice
   | all
   | extension-definition-element
```

۹-۴-۱ ویژگی‌ها، متغیرها و پارامترها

param، property و variable، هر سه، متغیرها را بیان می‌کند در نتیجه به عنوان عناصر انقیاد متغیرها شناخته می‌شوند. نام متغیر تعیین شده در صفت name از عنصر انقیاد متغیرها است (به زیربند ۵-۲-۳ مراجعه شود).

هدف از انقیاد متغیر پیروی از هم‌نیاهای[2] متعلق به عنصر انقیاد یافته متغیر و فرزندان آن‌ها است.

یادآوری- یک متغیر یا ویژگی تعیین شده در یک choice در خارج از choice قابل دسترسی نمی‌باشد.

1 - Tabs
2 -Siblings

۹-۴-۱-۱ ویژگی‌ها

عنصر property یک ویژگی از یک مقدار کاندیدا را تعیین می‌کند. هنگامی که یک مقدار کاندیدا در برابر یک نوع داده تصدیق شود، این مقوله با سه گانه نام/ نوع/ مقدار برای هر ویژگی در ارتباط است. اگر دو مقدار کاندیدا سه گانه نام/نوع/مقدار یکسان داشته باشند معادل در نظر گرفته می‌شود. برابری در مقادیر بر اساس نوع ویژگی ارزیابی می‌شود.

اگر تنها یک ویژگی برای مقدار کاندیدا مشخص شود در آن صورت این ویژگی ممکن است نامی نداشته باشد که در این مورد صفت name حذف شده است. اگر بیش از یک ویژگی برای یک مقدار کاندیدا مشخص شده باشد در این صورت تمامی ویژگی‌ها باید نام‌ها را مشخص نمایند.

اگر هیچ ویژگی توسط تعریف نوع داده به مقدار کاندیدا انتساب داده نشود در آن صورت سه گانه نام/ نوع/ مقدار از (val,' ',' ') انتساب می‌شود که val مقدارکاندیدا برای فضای خالی نرمال‌سازی شده می‌باشد.

```
property =
   element property { name?, type?, binding, extension-attribute* }
```

مثال- در نظر بگیرید:

```
<datatype name="color">
  <choice>
   <all>
    <regex ignore-regex-whitespace="true" case-insensitive="true">
     #([0-9A-F]{2})([0-9A-F]{2})([0-9A-F]{2})
    </regex>
    <property name="red" type="hexByte" select="$_1"/>
    <property name="green" type="hexByte" select="$_2"/>
    <property name="blue" type="hexByte" select="$_3"/>
   </all>
   <all>
     <regex case-insensitive="true">white</regex>
     <property name="red" type="hexByte" value="FF" />
     <property name="green" type="hexByte" value="FF" />
     <property name="blue" type="hexByte" value="FF" />
   </all>
  …
  </choice>
</datatype>
```

مقدار کاندیدای WHITE به مقادیر سه گانه نام/ نوع/ مقدار (('red', 'hexByte', 'FF')، ('green', 'hexByte', 'FF') و('blue', 'hexByte', 'FF')) انتساب خواهد یافت.

مقدار کاندیدای #FFFFFF به مقادیر سه گانه یکسان نام/ نوع/ مقدار انتساب خواهد یافت بنابراین تشخیص داده خواهد شد که دو مقدار برابر هستند.

۹-۴-۱-۲ متغیرها

عنصر variable یک مقدار را به یک متغیر مقید می‌سازد. متغیرها عیناً به ویژگی‌ها عمل می‌کنند مگر در مواردی که مقدار آنها به هنگام قضاوت در مورد تساوی، به کار برده نشود.

یادآوری- متغیرها می‌توانند برای محاسبات میانی استفاده شوند به عنوان مثال برای ارتقا قابلیت خواندن تعاریف نوع داده.

```
variable =
   element variable { name, type?, binding, extension-attribute* }
```

۹-۴-۱-۳ پارامترها

هنگامی که یک مقدار کاندیدا با یک نوع داده تعیین می‌شود، تعدادی از مقادیر پارامترها تشخیص داده می‌شوند. عناصر param در یک عنصر datatype مشخص می‌کند کدام پارامترها امکان دارد مقادیر را انتساب دهند و مقادیر پیش‌فرض به آن پارامترها که مقادیر انتساب داده نشده است. مقادیری که به پارامترها انتساب داده شده است از طریق انقیاد متغیر در تعریف نوع داده قابل دسترس است. اگر هیچ انقیادی برای پارامتر مشخص نشود، مقدار آن یک رشته خالی است.

```
param =
   element param { name, type?, binding?, extension-attribute* }
```

۹-۴-۱-۴ تصریح‌کننده‌های مقدار

دو روش برای تشخیص مقدار انتخابی برای ویژگی، متغیر یا پارامتر یا زمان آزمودن اعتبار یک مقدار وجود دارد: از طریق یک صفت value که مقدار حرفی را نگه می‌دارد یا از طریق یک صفت select که عبارت W3C XPath 2.0 را نگه می‌دارد. امکان دارد پیاده‌سازی‌ها عناصر انقیاد توسعه خود را برای ارائه مقدار انتخابی، تعریف نمایند. اگر نوعی مشخص شود (به زیربند ۹-۴-۱-۵ مراجعه شود) در آن صورت مقدار انتخابی باید در برابر آن نوع معتبر باشد.

```
binding = ( literal-value | select ), extension-binding-element*
```

اگر یک صفت value مشخص شود، متن تولید شده مقدار ویژگی، متغیر یا پارامتر را می‌گیرد.

```
literal-value = attribute value { text }
```

اگر یک مقدار صفت select مشخص شود، عبارتی را که مشمول آن است ارزیابی می‌شود (به زیربند ۵-۱-۱ مراجعه شود). اگر نوعی مشخص شود (به زیربند ۹-۴-۱-۵ مراجعه شود) در آن صورت مقدار انتخابی، مقدار رشته‌ای از نتیجه است به عبارت دیگر آن نتیجه‌ای از ارزیابی عبارت است:

```
select = attribute select { XPath }
```

عناصر انقیاد توسعه می‌توانند برای ارائه روش‌های جایگزین (مانند MathML یا XSLT) برای تعیین مقدار پارامتر، ویژگی یا متغیر استفاده شوند. در صورتی که پیاده‌سازی هیچ یک از عناصر انقیاد توسعه را پشتیبانی نکند در آن صورت به جای آن باید مقدار مشخص شده انتخابی را توسط صفت value یا select به متغیر انتساب دهد. اگر یک پیاده‌سازی بیش از یک عنصر انقیاد، را پشتیبانی نماید باید اولین عنصر انقیاد توسعه را استفاده نماید که آن را برای محاسبه مقدار متغیر پیاده‌سازی می‌کند.

```
extension-binding-element = extension-element
```

۹-۴-۱-۵ تصریح‌کننده‌های نوع

دو روش برای تشخیص یک نوع وجود دارد: از طریق مشخصه type و تعدادی از پارامترهای انقیادها، یا از طریق عنصر datatype بدون نام. در صورتی که پارامترها برای نوع مشخص می‌شوند، تعریف نوع داده برای آن نوع شامل آن پارامترها باشد.

```
type =
   (attribute type { text },
    param*)
   | anonymous-datatype
```

در صورتی که هیچ نوعی برای متغیر یا پارامتر یا ویژگی انتخاب نشده باشد، نوع مورد استفاده، از نوع XPath از مقدار انتخابی می‌باشد (رشته، عدد، اعداد بولی یا مجموعه- گره). در صورتی که مقدار انتخابی از یک ویژگی یا پارامتر یک مجموعه گره باشد در آن صورت باید با استفاده از یک سازوکار یکسان با تابع string() که در زیربند ۲-۳ از توابع W3C XPath 2.0 شرح داده شده است، به رشته تبدیل شود. پارامترها اجازه ندارند که اعداد یا اعداد بولی در آن‌ها قرار داده شود. در صورتی که مقدار انتخابی عدد یا عدد بولی باشد به جای آن رشته‌ای از عدد یا عدد بولی به عنوان مقدار انتخابی استفاده می‌شود.

صفت type یک نوع داده با نام را مشخص می‌کند. مقدار صفت type یک نام واجد شرایط می‌باشد. اگر هیچ پیشوندی مشخص نشده باشد در آن صورت فضای نام IRI نام واجد شرایط، صفت ns عنصری است که روی صفت typeای که اتفاق افتاده یا نزدیکترین عنصر جدی که یک صفت ns دارد ارائه می‌شود (اگر یک ns وجود دارد) یا در صورتی که یک صفت ns وجود نداشته باشد بدون فضای نام ns است.

نام واجد شرایط توسعه یافته توسط صفت type باید با نام واجد شرایط توسعه یافته از تعریف نوع داده‌ای که در همان نمونه انواع داده‌ای توسعه‌پذیر بعد از هر ساده‌سازی اعلان شده است تطبیق یابد. (به بند ۶ مراجعه شود).

۹-۴-۲ تجزیه

تجزیه مقادیر کاندیدا، دو تابع را اجرا می‌کند: اینکه آیا مقدار کاندیدا از یک قالب خاصی تبعیت می‌کند. و امکان دارد موجب تعدادی از انتساب‌ها شود که باعث ایجاد آزمون‌های بیشتر یا انتساب به ویژگی‌ها و متغیرها می‌شود.

۹-۴-۲-۱ تجزیه Rejex

عنصر regex تجزیه را با استفاده از یک عبارت منظم مطابق با عبارت منظم زبان W3C XPath 2.0، مشخص می‌کند. برای اینکه مقداری معتبر باشد مقدار کاندیدای فضای خالی- نرمال‌سازی شده کل باید توسط عبارات منظم تطبیق یابند. هنگام انجام تطبیق عبارت منظم، پیاده‌سازی باید طولانی‌ترین رشته ممکن مطابق با عبارت منظم را تطبیق دهد.

مثال ۱- رشته "FFFF" و عبارات منظم " ([A-Z] {1,2}) ([A-Z] {1,2}) ([A-Z] {1,2}) " گروه‌های تطبیقی به ترتیب "FF"، "F" و "F" هستند.

یادآوری ۱- با وجود اینکه استفاده از ^ و $ برای علامت آغاز و پایان رشته تطبیقی، مجاز است اما لازم نیست.

عنصر regex انقیادهای متغیر مربوط به زیر عبارات تطبیقی در عبارات منظم را فراهم می‌نماید. نام هر متغیر باید دارای قالب "{number}_" باشد که "{number}" مبنی بر ترتیبی از زیر عبارات تطبیقی است. مقدار انقیاد متغیر زیر رشته تطبیقی نوع " " است. متغیری که "0_" نامیده شده است باید دارای مقدار کلی رشته تطبیقی باشد.

یادآوری ۲- در صورتی که زیر عبارت چندین بار تطبیق داده شود آخرین مقدار تطبیق شده باید به متغیر گروه محدود شود.

```
regex =
   element regex {
    (regex-flags & extension-attribute*), regular-expression
    }
```

مثال ۲- رشته "zxc" تطبیقی با عبارت منظم "([A-Z] (A-Z) (A-Z) [A-Z])" باید باعث چهار انقیاد متغیر مانند 1_$ که دارای مقدار "Z"، 2_$ دارای مقدار "x"، 3_$ دارای مقدار "C" و 0_$ دارای مقدار "zxc" باشد.

۹-۴-۲-۱-۱ پرچم‌های عبارت منظم

چون عنصر regex همیشه رشته‌های تکی را تطبیق می‌دهد، عبارات منظم که با پرچم s به کار برده می‌شوند به درست تنظیم می‌شوند (یعنی حالت " تک خط" یا "تمام نقطه"). فرانویسه[1] تمامی نویسه‌ها را تطبیق می‌دهد که شامل نویسه خط جدید می‌باشد. پرچم m همیشه به نادرست تنظیم می‌شود (یعنی حالت چند خطی) مانند فرانویسه "^" شروع رشته کلی و "$" پایان یک رشته کلی، را تطبیق می‌دهد.

دو صفت روشی را که در عبارات منظم بکار برده می‌شوند را تصحیح می‌کند. این عبارت‌ها با پرچم‌های i و x موجود در XPath 2.0 معادل هستند.

به صورت پیش‌فرض، عبارت منظم نسبت به بزرگ یا کوچک بودن حروف حساس است. در صورتی که case-insensitive="true" باشد، تطبیق نسبت به بزرگ یا کوچک بودن حروف حساس است. قوانین برای حساسیت بزرگ یا کوچک بودن حروف همان قوانین W3C XPathe 2.0 هستند.

```
regex-flags = attribute case-insensitive { boolean }? &
   attribute ignore-regex-whitespace { boolean }?
```

به طور پیش‌فرض، فضای خالی در عبارت منظم با فضای خالی در رشته تطبیق می‌کند. اگر ignore-regex-whitespace="true" باشد، فضای خالی در عبارت منظم توسط پیاده‌سازی قبل از تطبیق حذف می‌شود. این ویژگی از انواع داده توسعه‌پذیر می‌تواند برای ایجاد بیشتر عبارات منظم قابل خواندن استفاده شود.

یادآوری ۱- تشخیص ignore-regex-whitspce="true" همانند تشخیص normalize- datatype whitespace="collapse">…</datatype نمی‌باشد که این امر موجب پیش‌پردازش خود مقدار کاندیدا می‌شود نه عبارت منظم.

یادآوری ۲- هنگامی که ignore-regex-whitespace="true" است عبارت منظم باید از الگوهایی استفاده نماید که شامل نویسه‌های فضای خالی (مانند "\S") برای تطبیق فضای خالی نمی‌باشد.

مثال- این عبارت منظم برای خوانایی بیشتر به سه خط جدا شده است:

```
<regex ignore-regex-whitespace="true">
   ([0-9]{4})-
   ([0-9]{2})-
   ([0-9]{2})
</regex>
```

1 -meta-character

۹-۴-۲-۲ فهرست‌ها

عنصر list تجزیه مقدار کاندیدا را به فهرستی از مقادیر کاندیدا مشخص می‌کند، به سادگی با استفاده از یک صفت seprator عبارات منظم جهت تجزیه فهرست به موارد را فراهم می‌کند.

صفت seprator یک عبارت منظم را مشخص می‌کند که جداکننده‌ها را در فهرست تطبیق می‌دهد. پیش فرض "+\S" (یک یا بیش از یک نویسه فضای خالی) است. در صورتی که عبارت منظم با یک رشته خالی تطبیق نماید یک خطا است.

```
\list =
   element list {
    attribute separator { regular-expression }?,
    extension-attribute*,
    type
   }
```

هر مورد در فهرست باید در برابر نوع داده مشخص شده توسط صفت type (و عناصر فرزند param) یا نوع داده بی‌نام مشخص شده توسط عنصر فرزند datatype معتبر باشد. برای جزئیات بیشتر در مورد چگونگی نوع به زیربند ۹-۴-۱-۵ مراجعه شود.

مثال- برای تعریف نوع داده توسعه‌پذیر

```
<list separator="\s*,\s*">
  <datatype>
    <regex>[0-9]+</regex>
  </datatype>
</list>
```

پس مقدار کاندیدای ۱، ۲، ۳، ۴۵ معتبر هستند اما مقدار کاندیدای sausages، egg، chips معتبر نیستند.

یادآوری ۱- روشی که به وسیله عناصر فهرست‌هایی جدا شده‌اند از تابع علامت‌گذار به صورتی که در توابع W3C XPath 2.0 §7.6.4. مشخص شده است، پیروی می‌کند.

یادآوری ۲- متن جداکننده به وسیله عبارت منظم مشخص تطبیقی کنار گذاشته می‌شود و نمی‌تواند در ارزیابی اعتبار نوع داده استفاده شود.

۹-۴-۳ آزمون

دو روش برای آزمون مقادیر وجود دارد: آزمون شرایط عمومی با عنصر condition و آزمون اعتبار در برابر نوع داده دیگر با عنصر valid.

۹-۴-۳-۱ شرایط

عنصر condition می‌آزماید که آیا یک شرط خاص توسط یک مقدار برآورده می‌شود یا خیر.

```
condition = element condition { extension-attribute*, test }
```

آزمون‌ها با صفت test که عبارت XPath را نگه می‌دارد، اعلان می‌شوند. در صورتی که مقدار بولی موثر در نتیجه ارزیابی عبارت XPath درست باشد در آن صورت آزمون باید با موفقیت باشد و شرط راضی‌کننده است. مقدار کاندیدا معتبر نیست در صورتی که هر شرط درون دامنه‌ای به نادرست ارزیابی شود.

```
test = attribute test { XPath }
```

مثال- تعریف نوع زیر، نوعی را تعریف می‌کند که باید دو شرط را برآورده سازد.

```
<datatype name="short">
   <condition test=". >= -32768" />
   <condition test=". &lt;= 32767" />
</datatype>
```

۹-۴-۳-۲ آزمون‌های اعتبار

عنصر valid اینکه آیا مقدار کاندیدای انتخابی در برابر یک تعریف نوع داده مشخص معتبر است یا خیر می‌آزماید. مقدار و نوع متغیرها انتخاب شده اند. (به زیربند ۹-۴-۱-۴ و ۹-۴-۱-۵ مراجعه شود) اگر هیچ مقداری مشخص نشده باشد در آن صورت مقدار به "."مقید می‌شود.

```
valid = element valid { extension-attribute*, type, binding? }
```

مثال- تعریف نوع زیر، نوعی را تعریف می‌کند که هم باید مطابق با تعریف نوع دیگر معتبر باشد (برای "int") و هم اینکه دو شرط را برآورده سازد.

```
<datatype name="short">
   <valid type="int"/>
   <condition test=". >= -32768"/>
    <condition test=". &lt;= 32767"/>
</datatype>
```

۹-۴-۴ عناصر منطقی

عناصر choice، all و except برای‌ترکیب آزمون‌ها استفاده می‌شود.

۹-۴-۴-۱ Choice

عنصر choice می‌آزماید آیا مقدار کاندیدا در برابر هر یک از آزمون‌های مشمول آن معتبر است؛ یعنی مقدار کاندیدا تنها زمانی معتبر است که یک یا چند آزمون را با موفقیت طی کند. اولین آزمون که مقادیر ویژگی را به مقدار کاندیدا تخصیص دهد، موفقیت‌آمیز است.

```
choice =
   element choice { extension-attribute*, datatype-definition-element+ }
```

۹-۴-۴-۲ All

عنصر all آزمون‌هایی را که باید برآورده شوند را با هم گروه‌بندی می‌کند. مقدار، کاندیدا معتبر است در صورتی که تمامی آزمون‌ها را برآورده سازد.

```
all = element all { extension-attribute*, datatype-definition-element+ }
```

۹-۴-۴-۳ Except

عنصر except شامل آزمون‌هایی است که نیاز است تا منفی بودن مقدار کاندیدا را ارزیابی نماید. مقدار کاندیدا تنها زمانی معتبر است که هیچ آزمونی به دست آمده در عنصر except را برآورده نکند.

```
except =
   element except { extension-attribute*, datatype-definition-element+ }
```

یادآوری- هر عنصر property در عنصر except نادیده گرفته شده است.

۹-۴-۵ عناصر توسعه تعریف

عناصر توسعه تعریف می‌تواند در هر نقطه از تعریف نوع داده برای مستندسازی، مثال‌ها و آزمون‌های تکمیلی استفاده شوند. رفتار آن‌ها در زیربند ۵-۳ شرح داده شده است.

```
extension-definition-element = extension-element
```

مثال- عناصر تعریف توسعه می‌تواند برای نگهداری مستندسازی در مورد نوع داده استفاده شود. به عنوان مثال یک عنصر eg:example امکان دارد برای ارائه مثال مقادیر قانونی از نوع داده استفاده شود.

```
<datatype name="RRGGBBColour" xmlns:eg="http://www.example.com">
   <eg:example>#FFFFFF</eg:example>
   <eg:example>#123456</eg:example>
   <regex>#(?'RR'[0-9A-F]{2})(?'GG'[0-9A-F]{2})(?'BB'[0-9A-F]{2})</regex>
   …
</datatype>
```

پیوست الف

(الزامی)

طرحواره‌ی RELAX NC برای اسناد انواع داده‌ای توسعه‌پذیر

```
namespace local = ""
default namespace dt = "http://purl.oclc.org/dsdl/extensible-datatypes"
XPath = text
boolean = "true" | "false"
regular-expression = text
anything =
   mixed {
    element * - dt:* {
     attribute * - dt:* { text }*,
     anything
    }*
   }
ns = attribute ns { text }
name = attribute name { text }
extension-attribute = attribute * - (local:* | dt:*) { text }
extension-element =
   element * - dt:* {
    must-implement?,
    attribute * - (dt:* | must-implement) { text }*,
    anything
   }
must-implement = attribute must-implement { boolean }
\include =
   element include {
    ns?,
    attribute href { text },
    extension-attribute*,
    top-level-element*
```

```
    }
combine = attribute combine { "choice" | "all" }
start = \datatypes
\datatypes =
    element datatypes {
      version, ns?, extension-attribute*, top-level-element*
    }
version = attribute version { "1.0" }
top-level-element =
    \include | named-datatype | \div | extension-top-level-element
\div =
    element div {
      ns?, version?, extension-attribute*, top-level-element*
    }
extension-top-level-element = extension-element
named-datatype =
    element datatype {
      name,
      ns?,
      preprocess?,
      combine?,
      extension-attribute*,
      param*,
      datatype-definition-element*
    }
anonymous-datatype =
    element datatype {
      preprocess?, extension-attribute*, datatype-definition-element*
    }
preprocess =
    attribute normalize-whitespace { "preserve" | "replace" | "collapse" }
datatype-definition-element =
    property
    | variable
```

```
   | regex
   | \list
   | condition
   | valid
   | except
   | choice
   | all
   | extension-definition-element
property =
   element property { name?, type?, binding, extension-attribute* }
variable =
   element variable { name, type?, binding, extension-attribute* }
param = element param { name, type?, binding?, extension-attribute* }
binding = (literal-value | select), extension-binding-element*
literal-value = attribute value { text }
select = attribute select { XPath }
extension-binding-element = extension-element
type =
   (attribute type { text },
   param*)
   | anonymous-datatype
regex =
   element regex {
    (regex-flags & extension-attribute*), regular-expression
   }
regex-flags =
   attribute case-insensitive { boolean }?
   & attribute ignore-regex-whitespace { boolean }?
\list =
   element list {
    attribute separator { regular-expression }?,
    extension-attribute*,
     type
   }
```

```
condition = element condition { extension-attribute*, test }
test = attribute test { XPath }
valid = element valid { extension-attribute*, type, binding? }
choice =
   element choice { extension-attribute*, datatype-definition-element+ }
all = element all { extension-attribute*, datatype-definition-element+ }
except =
   element except { extension-attribute*, datatype-definition-element+ }
extension-definition-element = extension-element
```

کتاب‌نامه

[1] XML Schema Part 1: Structures Second Edition, W3C Recommendation 28 October 2004, http://www.w3.org/TR/2004/REC-xmlschema-1-20041028/

[2] XML Schema Part 2: Datatypes Second Edition, W3C Recommendation 28 October 2004, http://www.w3.org/TR/2004/REC-xmlschema-2-20041028/