INSO
15743

1stEdition

2019

Identical with
ISO 19118: 2011

جمهوری اسلامی ایران
Islamic Republic of Iran
سازمان ملی استاندارد ایران

Iranian National Standardization Organization

استاندارد ملی ایران
۱۵۷۴۳
چاپ اول

۱۳۹۷

اطلاعات مکانی- کدگذاری

Geographic information - Encoding

ICS: 35.240.70

استاندارد ملی ایران شمارهٔ ۱۵۷۴۳(چاپ اول): سال ۱۳۹۷

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.gov.ir

وب‌گاه: http://www.isiri.gov.ir

Iranian National Standardization Organization (INSO)

No. 2592 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.gov.ir

Website: http://www.isiri.gov.ir

به نام خدا

آشنایی با سازمان ملی استاندارد ایران

سازمان ملی استاندارد ایران به موجب بند یک مادهٔ ۳ قاعده اصلاح قواعد و مقررات مؤسسهٔ استاندارد و تحقیقات صنعتی ایران، مصوب بهمن ماه ۱۳۷۱ تنها مرجع رسمی کشور است که وظیفهٔ تعیین، تدوین و نشر استاندارد 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 استاندارد 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 استانداردی 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، استانداردی 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 قاعدهی (OIML)[3] است و به عنوان تنها رابط[4] کمیسیون کدکس غذایی (CAC)[5] در کشور فعالیت می‌کند. در تدوین استاندارد ملی ایران ضمن توجه به شرایط کلی و نیازمندی‌های خاص کشور، از آخرین پیشرفت‌های علمی، فنی و صنعتی جهان و استاندارد بین‌المللی بهره‌گیری می‌شود.

سازمان ملی استاندارد ایران می‌تواند با رعایت موازین پیش‌بینی شده در قاعده، برای حمایت از مصرف‌کنندگان، حفظ سلامت و ایمنی فردی و عمومی، حصول اطمینان از کیفیت محصولات و ملاحظات زیست‌محیطی و اقتصادی، اجرای بعضی از استاندارد 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 صدور گواهی سامانه‌های 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 استاندارد 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

«اطلاعات مکانی- کدگذاری»

رئیس:	**سمت و/یا محل اشتغال:**
صبح‌دل، حسن (کارشناسی ارشد مهندسی عمران-سازه)	اداره کل استاندارد آذربایجان غربی
دبیر:	
جعفری خطایلو، مجتبی (کارشناسی مهندسی عمران)	اداره کل استاندارد آذربایجان غربی
اعضا: (اسامی به ترتیب حروف الفبا)	
پاک مهر، علی‌محمد (کارشناسی ارشد شیمی معدنی)	شرکت بازرسی دقیق مصالح آزمون
پناهی، سمیه (کارشناسی ارشد سیستم‌های اطلاعات مکانی)	اداره کل نقشه برداری شمال غرب
جاویدانه، علی (دکتری نقشه‌برداری سامانه‌های اطلاعات مکانی)	سازمان نقشه‌برداری کشور
خدادوست، مهدی (کارشناسی مهندسی عمران)	اداره کل استاندارد آذربایجان غربی
درستی، رحمن (کارشناسی مهندسی برق الکترونیک)	اداره کل استاندارد آذربایجان غربی
رستگارنیا، مهدی (کارشناسی ارشد مهندسی مکانیک)	اداره کل استاندارد آذربایجان غربی
رضائی، زهرا (کارشناسی ارشد سنجش از دور و سامانه اطاعات مکانی)	سازمان نقشه‌برداری کشور

اعضا: (اسامی به ترتیب حروف الفبا)	**سمت و/یا محل اشتغال:**
درزی، هما (کارشناسی ارشد مهندسی عمران- سیستم اطلاعات جغرافیایی)	سازمان نقشه‌برداری کشور
ساجدی درشکی، توحید (کارشناسی ارشد مهندسی عمران- ژئوتکنیک)	اداره کل استاندارد آذربایجان غربی
سعادت پور، نسیم (کارشناسی زبان و ادبیات انگلیسی)	عضو مستقل
صالح امین، فرزاد (کارشناسی ارشد سنجش از دور و سامانه اطاعات مکانی)	مهندسین مشاور ارکان پدید آب
علی‌آبادی، شمس‌الملوک (کارشناسی ارشد ریاضی کاربردی)	سازمان نقشه‌برداری کشور
علیزاده، شاهرخ (کارشناسی مهندسی معماری)	آزمایشگاه همکار مصالح آزمون آذربایجان
مرادی، محسن (کارشناسی ارشد فیزیک)	اداره کل استاندارد آذربایجان غربی
معینیان، شجاع الدین (کارشناسی مهندسی عمران)	آزمایشگاه همکار مصالح آزمون آذربایجان
ملک، محمدرضا (دکتری عمران، نقشه‌برداری سامانه‌های اطلاعات مکانی)	عضو هیات علمی دانشگاه خواجه نصیرالدین طوسی

ویراستار:	
ترابی، مهرنوش (کارشناسی ارشد مهندسی فن‌اوری اطلاعات)	کارشناس استاندارد- کارشناس ارشد شبکه و سخت‌افزار- شرکت برق منطقه‌ای هرمزگان

مندرجات فهرست

عنوان **صفحه**

پیش‌گفتار ح
مقدمه ط
۱ هدف و دامنهٔ کاربرد ۱
۲ انطباق ۱
۲-۱ مقدمه ۱
۲-۲ کلاس‌های انطباق مربوط به قواعد کدگذاری ۱
۲-۳ کلاس‌های انطباق مربوط به سرویس‌های کدگذاری ۲
۳ مراجع الزامی ۲
۴ اصطلاحات و تعاریف ۳
۵ نمادها و کوته‌نوشت‌ها ۱۱
۶ مفاهیم و فرضیات بنیادی ۱۱
۶-۱ مفاهیم ۱۱
۶-۲ تبادل داده ۱۲
۶-۳ الگوی کاربردی ۱۴
۶-۴ قاعده کدگذاری ۱۵
۶-۵ سرویس کدگذاری ۱۶
۶-۶ سرویس انتقال ۱۸
۷ مجموعه نویسه ۱۸
۸ مدل نمونه عمومی ۱۹
۸-۱ مقدمه ۱۹
۸-۲ ارتباط بین UML و مدل نمونه ۲۳
۹ قواعد کدگذاری ۲۴
۹-۱ مقدمه ۲۴
۹-۲ الزامات عمومی کدگذاری ۲۵
۹-۳ ساختار داده ورودی ۲۹
۹-۴ ساختار داده خروجی ۲۹
۹-۵ قواعد تبدیل ۲۹
۹-۶ مثال‌ها ۳۰
۱۰ سرویس کدگذاری ۳۰

عنوان **صفحه**

پیوست الف (الزامی) قاعده کد گذاری بر مبنای XML ۳۲
پیوست ب (الزامی) تسلسل آزمون انتزاعی ۳۴
پیوست پ (آگاهی دهنده) قاعده کدگذاری XML مبنا که مورد استفاده جوامع می‌باشند ۴۰
کتاب نامه ۹۶

پیش‌گفتار

استاندارد « اطلاعات مکانی- کدگذاری» که پیش‌نویس آن در کمیسیون‌های مربوط بر مبنای پذیرش استاندارد بین‌المللی به عنوان استاندارد ملی ایران به روش اشاره شده در مورد الف، بند ۷، استاندارد ملی ایران شمارهٔ ۵ تهیه و تدوین شده، در چهارمین اجلاسیه کمیتهٔ ملی استاندارد نقشه و اطلاعات مکانی مورخ ۱۳۹۷/۱۲/۱۴ تصویب شد. اینک این استاندارد به استناد بند یک مادهٔ ۳ قاعده اصلاح قواعد و مقررات مؤسسه استاندارد و تحقیقات صنعتی ایران، مصوب بهمن ماه ۱۳۷۱، به عنوان استاندارد ملی ایران منتشر می‌شود.

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

این استاندارد ملی بر مبنای پذیرش استاندارد بین‌المللی زیر به روش «معادل یکسان» تهیه و تدوین شده و شامل ترجمه تخصصی کامل متن آن به زبان فارسی می‌باشد و معادل یکسان استاندارد بین‌المللی مزبور است:

ISO 19118: 2011: Geographic information - Encoding

مقدمه

این استاندارد الزاماتی برای تبیین قواعد کدگذاری مورد استفاده برای تبادل اطلاعات مکانی در طیف مجموعه استاندارد بین المللی معروف به سری ISO 19100 را تعیین می‌کند. یک قاعده کدگذاری، به اطلاعات مکانی تعریف شده توسط الگوهای کاربردی و استاندارد شده، امکان کدگذاری درون ساختارهای مستقل داده را می‌دهد که مناسب فرایندهای ذخیره سازی و انتقال فرمت داده باشد. قاعده کدگذاری، انواع داده که باید کدگذاری شوند، قواعد دستوری، ساختار و الگوهای کدگذاری مورد استفاده در ساختار داده حاصله را تعیین می‌کند. ساختار داده حاصل می‌تواند بر روی رسانه‌های رقمی[1] ذخیره شده و یا با استفاده از پروتکل‌های انتقال فرمت، به فرمت دیگری تبدیل شود. هدف این است که داده توسط رایانه‌ها قابل خواندن و تفسیر باشد، اما در عین حال داده باید قابل خواندن برای انسان هم باشد.

انتخاب یک قاعده کدگذاری برای تبادل داده، مستقل از کاربرد، حوزه‌های کاربردی و جوامع مختلف را از تعریف و به‌کار بردن قواعد کدگذاری مختص خود که مستقل از بستر[2] باشد و با توجه به اندازه داده یا پیچیدگی فرایندهای پردازشی آن‌ها موثرتر باشد، منع نمی‌کند. زبان نشانه‌گذاری توسعه‌پذیر (XML)[3] یک زیرمجموعه از منبع [۵] کتاب‌نامه بوده و به دلیل مستقل بودن از بستر محاسباتی و تعامل‌پذیری[4] آن با اینترنت[5] انتخاب شده است.

این استاندارد به سه بخش منطقی تقسیم شده است؛ الزامات ایجاد قواعد کدگذاری براساس الگوهای ساختاری زبان مدل‌سازی متحد[6] (UML) که در بندهای ۶ تا ۹ تبیین شده‌اند. الزامات ایجاد سرویس کدگذاری تبیین‌شده در بند ۱۰، و الزاماتی برای XML بر مبنای قواعد کدگذاری که در پیوست **الف** تعیین شده‌اند.

قاعده کدگذاری مبتنی بر XML برای استفاده جهت تبادل بی‌طرفانه[7] داده در نظر گرفته می‌شود. این قاعده براساس (XML) و استاندارد مجموعه نویسه[8] استاندارد ISO/IEC 10646 است.

استاندارد اطلاعات مکانی در مجموعه‌ی استاندارد بین‌المللی معروف به مجموعه استانداردهای ISO 19100 سازماندهی می‌شوند. پیشینه و ساختار کلی این مجموعه از استاندارد و روش‌های اساسی توصیف در منبع-های [۱۲]،[۱۳] کتاب‌نامه و استاندارد ISO/TS 19103 تعریف شده‌اند.

1- Digital
2 - Platform
3- Extensible Markup Lanquage
4 - Interoperable
5 - World Wide Web
6- Unified Modelling Language
7 - Neutral
8- Character Set

کاربران این استاندارد میتوانند الگوهای کاربردی را برای توصیف رسمی اطلاعات مکانی توسعه دهند. یک الگوی کاربردی توسط یک‌پارچه‌سازی اجزا از سایر الگوهای مفهومی استاندارد شده (برای مثال منبع [۵] کتاب‌نامه) گردآوری می‌شود. چگونگی شکل‌گیری این یکپارچه‌سازی در استاندارد ISO 19109 توصیف می‌شود. همچنین مجموعه استاندارد بین‌المللی معروف به مجموعه استانداردهای ISO 19100 یک مجموعه از سرویس‌های متداول را که در هنگام توسعه برنامه‌های کاربردی اطلاعات مکانی در دسترس می‌باشند، تبیین می‌کند. خدمات متداول و بطور کلی دسترسی به آن‌ها، پردازش‌های آن‌ها، اطلاعات مکانی مطابق با مدل اطلاعاتی رایج آن‌ها، همگی در منبع [۲۴] کتاب‌نامه تبیین می‌شود. این استاندارد مباحث پیاده سازی این موضوعات را نیز پوشش می‌دهد.

اطلاعات مکانی- کدگذاری

۱ هدف و دامنهٔ کاربرد

هدف از تدوین این استاندارد، تعیین الزاماتی در تعریف قواعد کدگذاری جهت تبادل داده در تطبیق اطلاعات مکانی مطابق با مجموعه استاندارد بین‌المللی معروف به مجموعه استانداردهای ISO 19100 می‌باشد.

این استاندارد موارد زیر را مشخص می‌نماید:

- الزاماتی برای ایجاد قواعد کدگذاری براساس الگوهای UML؛
- الزاماتی برای ایجاد سرویس‌های کدگذاری؛ و
- الزاماتی برای قواعد کدگذاری بر مبنای XML جهت تبادل بی‌طرفانه[1] داده.

این استاندارد هیچ‌گونه رسانه رقمی را تعیین نمی‌کند، هیچ‌گونه سرویس‌های انتقال یا پروتکل‌های انتقال داده را تعریف نکرده و چگونگی کدگذاری درون‌خطی تصاویر بزرگ را نیز تعیین نمی‌کند.

۲ انطباق

۲-۱ مقدمه

دو مجموعه کلاس[2] انطباق در این استاندارد شرح داده شده است.

۲-۲ کلاس‌های انطباق مربوط به قواعد کدگذاری

تمام قواعد کدگذاری باید تمام موارد آزمون مجموعه آزمون انتزاعی زیربند ب-۱ را برآورده نمایند. تمام قواعد کدگذاری باید تمام موارد آزمون مجموعه زیربند ب-۲ و یا ب-۳ را برآورده نمایند.

جدول۱- کلاس‌های انطباق مربوط به قواعد کدگذاری

کلاس انطباق	زیربند مجموعه آزمون انتزاعی
تمام قواعد کدگذاری	ب-۱
قواعد کدگذاری با تبدیل نمونه	ب-۲
قواعد کدگذاری با تبدیل الگو	ب-۳

1- Neutral
2- Class

۲-۳ کلاس‌های انطباق مربوط به سرویس‌های کدگذاری

تمام سرویس‌های کدگذاری باید تمام کلاس‌های آزمون مجموعه آزمون انتزاعی زیربند ب-۴ را برآورده نمایند. با درنظر گرفتن قابلیت‌های سرویس کدگذاری، این سرویس‌ها باید تمام حالت‌های آزمون اضافی انطباق را با توجه به جدول ۲ برآورده نمایند.

جدول ۲- کلاس‌های انطباق مربوط به سرویس‌های کدگذاری

کلاس انطباق	زیربند ابزار آزمون انتزاعی
تمام قواعد کدگذاری	ب-۴
سرویس کدگذاری عمومی	ب-۵
سرویسی که داده را کدگذاری می‌کند.	ب-۶
سرویسی که داده را کدگشایی می‌کند.	ب-۷
سرویسی که الگوی ساختاری داده خروجی را تولید می‌کند.	ب-۸

۳ مراجع الزامی

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

درصورتی‌که 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 ISO 8601: 2004, Data elements and interchange formats– Information interchange- Representation of dates and times

2-2 ISO/IEC 10646: 2011, Information technology- Universal Coded Character Set (UCS)

2-3 ISO/TS 19103: 2005, Geographic information- Conceptual schema language

2-4 ISO 19109: 2005, Geographic information- Rules for application schema

2-5 Extensible Markup Language (XML) 1.0, W3C Recommendation. Available at http://www.w3.org/TR/REC-xml

۴ اصطلاحات و تعاریف

در این استاندارد، اصطلاحات و تعاریف زیر به‌کار می‌رود:

۱-۴

الگوی کاربردی

application schema

الگوی مفهومی (زیربند ۴-۵) برای **داده** (زیربند ۴-۸) الزامی یک یا چند کاربرد می‌باشد.

[زیربند 4.2، منبع [۱۲] کتاب‌نامه]

یادآوری- یک الگوی کاربردی، محتوا، ساختار و محدودیت‌های قابل‌به‌کارگیری در **اطلاعات** (زیربند ۴-۲۲) در یک حوزه کاربردی ویژه را توصیف می‌کند.

۲-۴

نویسه

character

عضوی از یک مجموعه از اجزا مورد استفاده برای نمایش، سازماندهی یا واپایش[1] **داده** (زیربند ۴-۸) است.

[زیربند 01.02.11، منبع [۱] کتاب‌نامه]

۳-۴

کد

code

نمایش یک برچسب مطابق یک الگوی مشخص می‌باشد.

۴-۴

مدل مفهومی

conceptual model

مدل (۴-۷) مفاهیم یک **جهان گفتمان** (۴-۳۳)می‌کند.

[زیربند 4.4، منبع [۱۲] کتاب‌نامه]

1-Control

۵-۴

الگوی مفهومی

conceptual schema

توصیف قراردادی از یک **مدل مفهومی**(زیربند ۴-۴) است.

[زیربند 4.5، منبع [۱۲] کتاب‌نامه]

۶-۴

زبان الگوی مفهومی

conceptual schema language

زبان قراردادی براساس یک قرارداد مفهومی با هدف نمایش **الگوهای مفهومی** (زیربند ۴-۵) می‌باشد.

[زیربند 4.6، منبع [۱۲] کتاب‌نامه]

مثال: UML، EXPRESS،IDEF1X

یادآوری- یک زبان الگوی مفهومی ممکن است لغوی یا ترسیمی[1] باشد.

۷-۴

قاعده تبدیل

conversion rule

قاعده تبدیل نمونه‌ها در ساختار **داده** (زیربند ۴-۸) ورودی به نمونه‌هایی در ساختار داده خروجی است.

۸-۴

داده

data

نمایشی از **اطلاعات** (زیربند ۴-۲۲) با قابلیت تفسیر مجدد در یک شیوه رسمی مناسب برای ارتباط، تفسیر، یا پردازش می‌باشد.

[زیربند 01.01.02، منبع [۱] کتاب‌نامه]

1 - Graphical

۹-۴

تبادل داده

data interchange

تحویل، دریافت و تفسیر **داده** (زیربند ۴-۸) است.

۱۰-۴

انتقال داده

data transfer

جابجایی **داده** (زیربند ۴-۸) از یک نقطه به نقطه دیگر در بیش از یک **رسانه** (زیربند ۴-۲۶) می‌باشد.

یادآوری- انتقال **اطلاعات** (زیربند ۴-۲۲) بیانگر انتقال داده می‌باشد.

۱۱-۴

نوع داده

data type

تعیین یک **دامنه مقدار** (زیربند ۴-۳۴) با عملیات مجاز برروی مقادیر در این دامنه است.

[زیربند 4.1.5، استاندارد ISO/TS 19103: 2005]

مثال‌ها: عدد صحیح، حقیقی، بولی[1]، رشته‌ای و تاریخ.

یادآوری- نوع داده توسط یک عبارت مشخص می‌شود، برای مثال عدد صحیح. مقادیر نوع داده از دامنه مقدار تعیین شده می‌باشند برای مثال تمام اعداد صحیح بین ۶۵۵۳۷- و ۶۵۵۳۶ قرار دارند. مجموعه عملگرها می‌توانند +، -، × و ÷ باشد که دارای معنی تعریف‌شده‌ای می‌باشد. نوع داده می‌تواند ساده یا مرکب باشد. نوع داده ساده یک دامنه مقدار معین را تبیین می‌کند که مقادیر آن، یک مقدار معین تجزیه‌ناپذیر به خود می‌گیرند، برای مثال اعداد صحیح. یک نوع داده مرکب یک مجموعه از انواع داده می‌باشد که در کنار یکدیگر گروه‌بندی می‌شوند. یک نوع داده مرکب ممکن است بیانگر یک شی باشد و بنابراین می‌تواند دارای هویت باشد.

۱۲-۴

مجموعه داده

dataset

مجموعه‌ای قابل‌شناسایی از **داده** (زیربند ۴-۸) می‌باشد.

---

1 - Bolean

[زیربند 4.2، منبع [۲۱] کتاب‌نامه]

۱۳-۴

کدگذاری

encoding

تبدیل **داده** (زیربند ۴-۸) به دنباله‌ای از **کدها** (زیربند ۴-۳) می‌باشد.

۱۴-۴

قاعده کدگذاری

encoding rule

مجموعه قابل‌شناسایی از قواعد **تبدیل** (زیربند ۴-۷) بیانگر **کدگذاری** (زیربند ۴-۱۳) برای یک ساختار **داده** (زیربند ۴-۸) خاص است.

مثال‌ها: XML، منبع‌های [۲] و [۹] کتاب‌نامه،

یادآوری- یک قاعده کدگذاری انواع داده که باید تبدیل شوند و همچنین قواعد دستوری[1]، ساختار و کدهای استفاده شده در ساختار داده حاصل را تعریف می‌کند.

۱۵-۴

سرویس کدگذاری

encoding service

مولفه نرم‌افزاری دارای یک قاعده **کدگذاری** (زیربند ۴-۱۴) پیاده‌سازی شده می‌باشد.

۱۶-۴

عارضه

feature

انتزاعی از پدیده‌های جهان حقیقی می‌باشد.

[زیربند 4.11، منبع [۱۲] کتاب‌نامه]

یادآوری- یک عارضه ممکن است به شکل یک نوع یا یک نمونه رخ دهد. نوع عارضه یا نمونه عارضه هنگامی‌که یکی از آن‌ها مورد نظر باشد، استفاده می‌شود.

1- Syntax

۴-۱۷

فایل

file

یک مجموعه نامگذاری شده از سوابق[1] ذخیره شده یا پردازش شده به عنوان یک واحد است.

[زیربند 01.08.06، منبع [۱] کتاب‌نامه]

۴-۱۸

داده‌ی مکانی

geographic data

دادهای (زیربند ۴-۸) با ارجاع صریح یا ضمنی به یک موقعیت نسبت به زمین می‌باشد.

[زیربند 4.12، استاندارد ISO 19109: 2005]

۴-۱۹

اطلاعات مکانی

geographic information

اطلاعاتی (زیربند ۴-۲۲) ارتباط دهنده یک پدیده به یک موقعیت نسبت به زمین به طور صریح یا ضمنی است.

[زیربند 4.16، منبع [۱۲] کتاب‌نامه]

۴-۲۰

شناسه

identifier

دنباله‌ای مستقل از زبان از **نویسه‌ها** (زیربند ۴-۲) قادر به شناسایی منحصر به فرد و دائمی برای مولفه‌ای که آنرا همراهی می‌کند.

[زیربند 4.1.5، منبع [۲۶] کتاب‌نامه]

1- Records

۲۱-۴

قرارداد شناسایی

identification convention

مجموعه‌ای از قواعد برای ایجاد **شناسه**ها (زیربند ۴-۲۰) می‌باشد.

۲۲-۴

اطلاعات

information

دانشی دربارهٔ موضوعاتی مانند واقعیت‌ها، رویدادها، چیزها[1]، فرآیندها، یا ایده‌ها، شامل مفاهیمی دارای معنی خاص که در زمینه‌ای معین است.

[زیربند 01.01.01، منبع [۱] کتاب‌نامه]

۲۳-۴

مدل نمونه

instance model

مدل (زیربند ۴-۲۷) نمایش برای ذخیره **داده** (زیربند ۴-۸) مطابق با **الگوی کاربردی** (زیربند ۴-۱) می‌باشد.

۲۴-۴

واسط

interface

<UML>، مجموعه عملیات نام‌گذاری شده‌ی مشخص کننده رفتار یک جز[2] می‌باشد.

[منبع [۲۹] کتاب‌نامه]

1 -Thing
2- Element

۲۵-۴

تعامل‌پذیری

interoperability

قابلیت ارتباط، اجرای برنامه‌ها یا انتقال **داده** (زیربند ۴-۸) بین واحدهای عملیاتی مختلف، به طریقی که کاربر را الزام به نداشتن یا داشتن دانش کم از ویژگی‌های منحصر به فرد این واحدها، بتوانند از این واحدها استفاده کنند.

[زیربند 01.01.47، منبع [۱] کتاب‌نامه]

۲۶-۴

رسانه

medium

ابزار یا واسطی برای ذخیره‌سازی یا انتقال **داده** (زیربند ۴-۸) است.

مثال‌ها: لوح فشرده، اینترنت، امواج رادیویی و غیره.

۲۷-۴

مدل

model

انتزاع برخی جنبه‌های واقعیت است.

[زیربند 4.14، استاندارد ISO 19109: 2005]

۲۸-۴

الگو

schema

توصیف قراردادی از یک **مدل** (زیربند ۴-۲۷) است.

[زیربند 4.25، منبع [۱۲] کتاب‌نامه]

۲۹-۴

مدل الگو

schema model

مدل (زیربند ۴-۲۷) نمایش برای ذخیره‌سازی **الگوها** (زیربند ۴-۲۸) است.

مثال: مدل نمایش برای یک انباره الگو

۳۰-۴

کلیشه

stereotype

<UML>، نوع جدید از جز مدل‌سازی بسط‌دهنده قواعد معنایی فرامدل است.

[منبع [۲۹] کتاب‌نامه]

یادآوری- ضروری است که کلیشه‌ها منطبق با انواع موجود معین یا کلاس‌های موجود معین در فرامدل باشد. کلیشه‌ها ممکن است قواعد معنایی را بسط دهند، اما ساختار، انواع و کلاس‌هایی که از قبل وجود داشته را بسط نمی‌دهند. کلیشه‌های معین در UML از پیش تعریف شده‌اند، سایر کلیشه‌ها ممکن است توسط کاربر تعریف شوند. کلیشه‌ها یکی از سه سازوکار توسعه-پذیری در UML می‌باشند؛ سازوکارهای دیگر، به‌کارگیری قیدها و مقادیر برچسب گذاری شده می‌باشند.

۳۱-۴

پروتکل انتقال

transfer protocol

مجموعه رایج از قواعد برای تبیین تعامل‌ها بین سامانه‌های توزیع‌یافته می‌باشد.

۳۲-۴

واحد انتقال

transfer unit

مجموعه‌ای از **داده** (زیربند ۴-۸) با هدف انتقال **داده** (زیربند ۴-۱۰) است.

یادآوری- نیاز نیست که یک واحد انتقال مانند یک مجموعه **داده** (زیربند ۴-۱۲) قابل شناسایی باشد.

۳۳-۴

جهان گفتمان

universe of discourse

چشم‌انداز جهان حقیقی یا فرضی که شامل همه موارد مورد توجه می‌باشد.

[زیربند 4.29، منبع [۱۲] کتاب‌نامه]

۳۴-۴

دامنه مقدار

value domain

مجموعه‌ای از مقادیر پذیرفته شده است.

[زیربند 4.1.15، استاندارد ISO/TS 19103: 2005]

مثال: بازه ۳ تا ۲۸، تمام اعداد صحیح، هرگونه نویسه، شمارش تمام مقادیر پذیرفته شده (آبی، سبز، سفید).

۵ نمادها و کوته‌نوشت‌ها

DCE	Distributed Computing Environment
DUID	Domain Unique Identifier
HTML	Hypertext Markup Language
MODIS	Moderate Resolution Imaging Spectroradiometer
POSC	Petroleum Open Standards Consortium
TIFF	Tagged Image File Format
UCS	Universal Multiple-Octet Coded Character Set
UML	Unified Modelling Language
UTF	UCS Transfer Format
UUID	Universally Unique Identifier
XML	Extensible Markup Language

محیط محاسباتی توزیع‌یافته

شناسه منحصر به فرد دامنه

زبان نشانه‌گذاری ابرمتن

طیف‌سنج تصویربرداری با قدرت تفکیک متوسط

کنسرسیوم استانداردهای آزاد پتروشیمی

فرمت فایل تصویر برچسب گذاری شده

مجموعه نویسه کدگذاری شده در قالب چندگانه-هشتایی بین‌المللی

زبان مدلسازی متحد

قالب انتقال UCS

شناسه منحصر به فرد بین‌المللی

زبان نشانه‌گذاری توسعه‌پذیر

۶ مفاهیم و فرضیات بنیادی

۶-۱ مفاهیم

هدف مجموعه استانداردهای ISO 19100 ایجاد قابلیت تعامل‌پذیری بین سامانه‌های اطلاعات مکانی ناهمگون و مختلف می‌باشد. برای رسیدن به قابلیت تعامل‌پذیری بین سامانه‌های اطلاعات ناهمگون، لازم است تا دو مسئله بنیادی تعریف شود. مسئله اول تبیین قواعد معنایی محتوی و ساختارهای منطقی داده‌ی مکانی می-

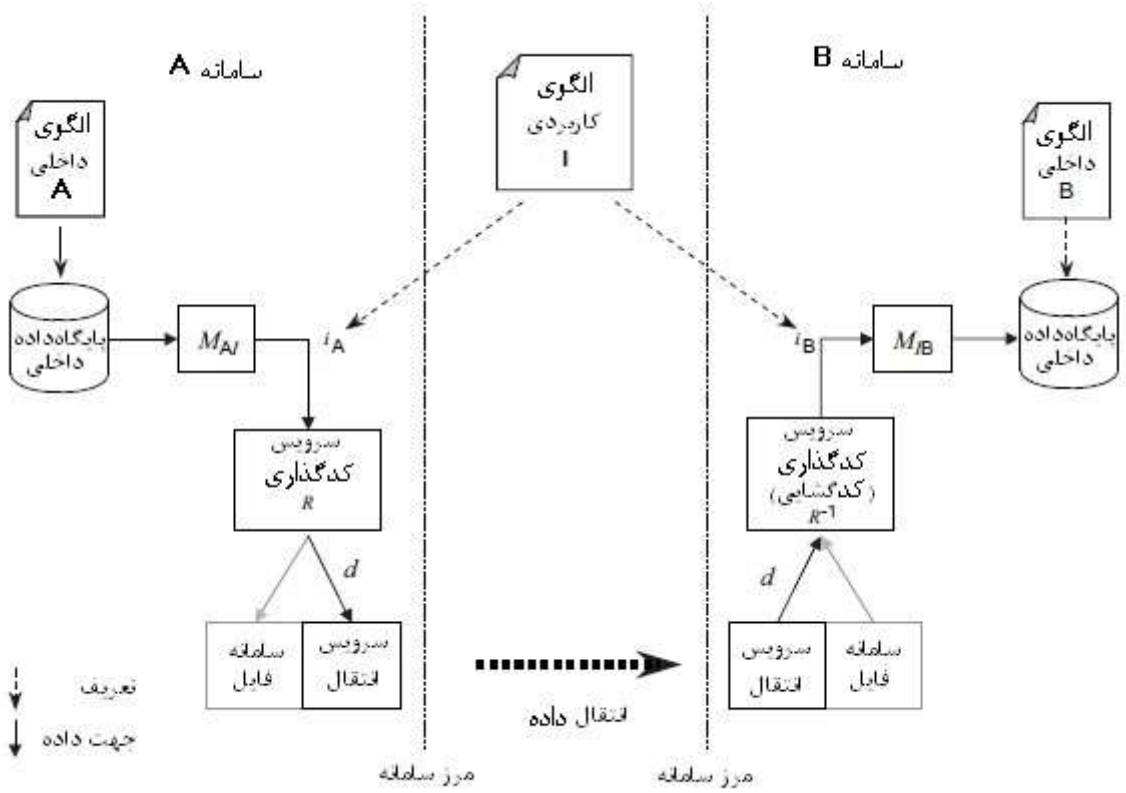
باشد. این موضوع باید در یک الگوی کاربردی صورت پذیرد. مسئله دوم، تبیین یک ساختار داده مستقل از بستر[1] و سامانه می‌باشد که بتواند داده متناظر با الگوی کاربردی را ارائه کند.

مفاهیم بنیادین تبادل داده، یعنی روشی مبتنی بر الگوی کاربردی برای کدگذاری، تحویل، دریافت و تفسیر داده‌ی مکانی در زیربندهای ۶-۲ تا ۶-۶ توصیف می‌شوند. یک مرورکلی از فرآیند تبادل داده در زیربند ۶-۲ توصیف شده؛ زیربند ۶-۳ الگوهای برنامه کاربردی را که اجازه تفسیر داده‌ی مکانی را می‌دهند، معرفی می‌کند؛ زیربند ۶-۴ اهمیت قاعده کدگذاری برای ایجاد ساختارهای داده مستقل از سامانه را توصیف می‌کند؛ زیربند ۶-۵ یک مولفه نرم‌افزاری با نام سرویس کدگذاری برای اجرای قاعده کدگذاری توصیف می‌کند؛ و زیربند ۶-۶ روشی به نام سرویس انتقال، برای تحویل و دریافت توصیف می‌کند.

۶-۲ تبادل داده

یک مرورکلی از تبادل داده در شکل ۱ نشان داده شده است. سامانه A خواهان ارسال یک مجموعه داده به سامانه B می‌باشد. برای اطمینان از تبادل موفق، ضروری است که A و B در ارتباط با سه موضوع توافق کنند: الگوی کاربردی مشترک I، کدام قاعده کدگذاری R اعمال شود و چه پروتکلی برای انتقال استفاده شود. درواقع، الگوی کاربردی، اساس انتقال موفق داده بوده و محتوای ممکن و ساختار داده انتقال یافته را تعریف کرده، در حالی‌که قاعده کدگذاری، قاعده تبدیل را برای نحوه کد کردن داده به ساختار داده مستقل از سامانه تبیین می‌کند.

1- Platform



شکل ۱- بررسی اجمالی تبادل داده بین دو سامانه

هردو سامانه A و B داده را مطابق بایک الگوی داخلی درون پایگاه داده داخلی ذخیره می‌کنند، اما الگوها به‌طورمعمول متفاوت می‌باشند یعنی الگوی A با الگوی B یکسان نمی‌باشد. ضروری است تا مراحل منطقی زیر را به منظور انتقال یک مجموعه داده از پایگاه‌داده داخلی سامانه A به پایگاه داده داخلی سامانه B در پیش گرفت.

الف- اولین مرحله برای سامانه A، تبدیل داده‌ی داخلی خود به یک ساختار مطابق با الگو کاربردی مشترک I می‌باشد. این امر با تعریف یک نگاشت از مفاهیم الگوی داخلی به مفاهیم تعریف شده در الگوی کاربردی و نوشتن نرم‌افزار نگاشت برای تبدیل نمونه‌های داده انجام می‌گیرد. در شکل ۱، این نگاشت به صورت M_{AI} مشخص شده است. نتیجه این کار یک ساختار داده مخصوص الگو کاربردی، i_A، می‌باشد. ساختار داده در حافظه یا بروی یک فایل واسط ذخیره شده و وابسته به سامانه می‌باشد، بنابراین این فایل هنوز مناسب برای انتقال نمی‌باشد.

ب- مرحله بعدی استفاده از یک سرویس کدگذاری است که قاعده کدگذاری R را برای ایجاد یک ساختار داده مستقل از سامانه که برای انتقال مناسب است، به کار می‌برد. این مجموعه داده کدگذاری شده d نامیده شده است و ممکن است در یک سامانه فایل ذخیره یا با استفاده از یک سرویس انتقال، منتقل گردد.

پ- سپس سامانه A یک سرویس انتقال را جهت انتقال مجموعه داده کدگذاری شده d به سامانه B فراخوانی می‌کند. این سرویس انتقال از یک پروتکل انتقال برای چگونگی بسته‌بندی و نحوه انتقال داده با استفاده از واسطهای ارتباط برخط[1] یا برون‌خط[2]، پیروی می‌کند. ضروری است که هر دو طرف در پروتکل انتقال استفاده شده توافق داشته باشند.

ت- سرویس انتقال روی سامانه B مجموعه داده‌ی منتقل شده را دریافت کرده، و مطابق با پروتکل، مجموعه داده باز شده را روی، برای مثال یک فایل واسط، به عنوان یک مجموعه داده‌ی کدگذاری شده d ذخیره می‌کند.

ث- به منظور به دست آوردن یک ساختار داده مشخص الگوی کاربردی i_B، سامانه B قاعده کدگذاری معکوس R^{-1} را برای تفسیر داده‌ی کدگذاری شده به کار می‌برد.

ج- برای استفاده از مجموعه داده، ضروری است که سامانه B ساختار داده مخصوص الگوی کاربردی i_B را با پایگاه‌داده داخلی خود تبدیل کند. این کار به وسیله تعریف یک نگاشت از الگوی کاربردی به الگوی داخلی آن و با نوشتن نرم‌افزاری که تبدیل واقعی را انجام می‌دهد، صورت می‌گیرد. در شکل ۱ این نگاشت به صورت M_{IB} مشخص شده است.

این استاندارد فقط الزامات ایجاد قواعد کدگذاری و سرویس‌های کدگذاری را تعیین می‌کند و الزامات کل فرآیند تبادل داده را پوشش نمی‌دهد. بنابراین، تنها مراحل ب و ث استاندارد شده هستند. مراحل الف، پ، ت و ج از سرویس‌های عمومی فن‌اوری اطلاعات استفاده می‌کنند.

۶-۳ الگوی کاربردی

الگوی کاربردی یک الگوی مفهومی برای کاربردهایی با الزامات داده مشابه می‌باشد. الگوی کاربردی مبنایی برای تبادل موفق داده بوده و محتوا و ساختار ممکن داده را تعریف می‌کند. همچنین مبنایی برای پیاده-سازی ساختارهای داده خاص- الگو- کاربرد[3] برای ذخیره‌سازی محلی داده می‌باشد.

الگوی کاربردی استفاده شده برای کدگذاری منطبق با این استاندارد باید به زبان الگوی مفهومی UML مطابق با استانداردهای ISO/TS 19103 و ISO 19109 نوشته شود. این استانداردهای بین‌المللی چارچوبی برای چگونگی نوشتن الگوهای کاربردی را تعیین می‌کنند. قواعد، شامل ویژگی‌هایی درباره نحوه استفاده از الگوهای استاندارد شده برای تعریف انواع عارضه‌ها می‌باشند. ضروری است که هردوی فرستنده و دریافت-کننده‌ی داده به الگوی کاربردی دسترسی داشته باشند.

1- Online
2- Offline
3- Application- Schema- Specific

الگوی کاربردی باید به منظور اطمینان از موفقیت نتیجه برای هر دو طرف تبادل داده در دسترس باشد. ضروری است که الگوی کاربردی قبل از تبادل داده، انتقال یابد. به این ترتیب هردوی فرستنده و گیرنده می‌توانند با پیاده‌سازی نگاشت‌ها و ساختارهای داده مطابق الگوی کاربردی سامانه‌های خود را آماده سازند. الگوی کاربردی ممکن است به همراه مجموعه داده منتقل شود یا در یک محل عمومی ذخیره شده و از مجموعه داده به آن ارجاع شود.

الگوی کاربردی ممکن است به روش‌های کاغذی و یا الکترونیکی مبادله شود.

۶-۴ قاعده کدگذاری

۶-۴-۱ مفهوم

قاعده کدگذاری مجموعه‌ی قابل شناسایی از قواعد تبدیل بوده که کدگذاری را برای یک ساختار داده خاص تبیین می‌کنند. قاعده کدگذاری تعیین‌کننده انواع داده که تبدیل می‌شوند، قواعد دستوری، ساختار و الگوهای کدگذاری استفاده شده در ساختار داده‌ی حاصله، می‌باشند. یک قاعده کدگذاری به ساختار داده خاص- الگو- کاربرد ، برای ایجاد ساختارهای داده مستقل از سامانه که مناسب انتقال یا ذخیره است، اعمال می‌شود. به منظور تعریف یک قاعده کدگذاری، ضروری است که سه جنبه مهم این امر تعیین شوند: ساختار داده ورودی، ساختار داده خروجی و قواعد تبدیل بین اجزای ساختار داده ورودی و خروجی. هردو ساختار داده ورودی و خروجی با استفاده از زبان الگوی مفهومی نوشته شده و مفاهیم موجود در این زبان‌ها برای تبیین قاعده کدگذاری استفاده می‌شوند.

۶-۴-۲ ساختار داده ورودی

ساختار داده ورودی یک ساختار داده خاص- الگو- کاربرد می‌باشد. ساختار داده می‌تواند به صورت یک مجموعه از نمونه داده در نظر گرفته شود، یعنی $i = \{i_1, ..., i_p\}$؛ به شکل۱ مراجعه شود. هر نمونه داده، i_k، یک نمونه از مفهوم بوده، I_l، که در یک الگوی کاربردی تعریف شده است. الگوی کاربردی، مجموعه‌ای از مفاهیم تعریف شده در الگوی کاربردی $I = \{I_1, ..., I_m\}$ را تبیین می‌کند.

الگوی کاربردی یک الگوی مفهومی، c، می‌باشد که به زبان الگوی مفهومی، C نوشته شده است. الگوی مفهومی یک مجموعه از مفاهیم $c = \{c_1, ..., c_m\}$ را به وسیله کشف مفاهیم زبان الگوی مفهومی $C=\{C_1,...,C_r\}$ تعریف می‌کند. از آن‌جایی که الگوی کاربردی یک الگو مفهومی می‌باشد، لذا $c = I$.

۶-۴-۳ ساختار داده خروجی

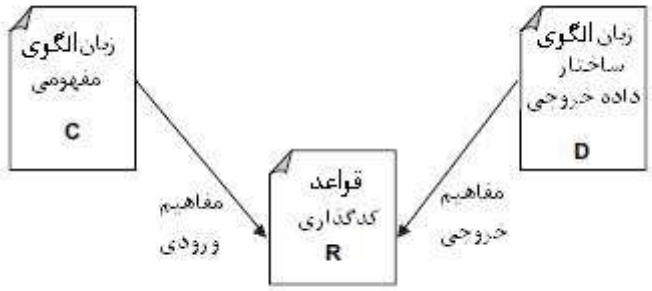
ساختار داده خروجی به وسیله‌ی یک الگو $D = \{D_1, ..., D_s\}$ تعریف شده است. D الگویی برای ساختار خروجی بوده و در شکل۱ نشان داده نشده است. ساختار داده خروجی می‌تواند به عنوان یک مجموعه از

نمونه‌های داده در نظر گرفته شود یعنی $d = \{d_1, ..., d_q\}$ که هر نمونه داده d_k یک نمونه از یک مفهوم D_l می‌باشد.

الگوی D، قواعد دستوری، ساختار و الگوهای کدگذاری ساختار داده خروجی را تعریف می‌کند.

۶-۴-۴ قواعد تبدیل

قاعده تبدیل، چگونگی تبدیل نمونه داده در ساختار داده ورودی را به صفر، یک یا نمونه‌های بیشتر در ساختار داده خروجی تعیین می‌کند. قواعد تبدیل، بر مبنای مفاهیم زبان الگوی مفهومی، C و مفاهیم الگوی ساختار داده خروجی، D تعریف شده است. تعیین یک قاعده تبدیل R_i، برای هر ترکیب قانونی مفاهیم در زبان الگوی مفهومی ضروری است. مجموعه قواعد تبدیل عبارتند از $R=\{R_1, ..., R_n\}$، که R_i قاعده تبدیل iام و C_i ترکیب قانونی iام نمونه‌ها از زبان الگو می‌باشد. می‌توان یک جدول تبدیل برای تمام C_iهای ممکن تهیه نمود به نحوی که هر C_i به نمونه‌ای در ساختار داده‌ی خروجی D نگاشته شود. شکل۲ رابطه بین زبان الگوی مفهومی ورودی و خروجی و قاعده کدگذاری را نشان می‌دهد.



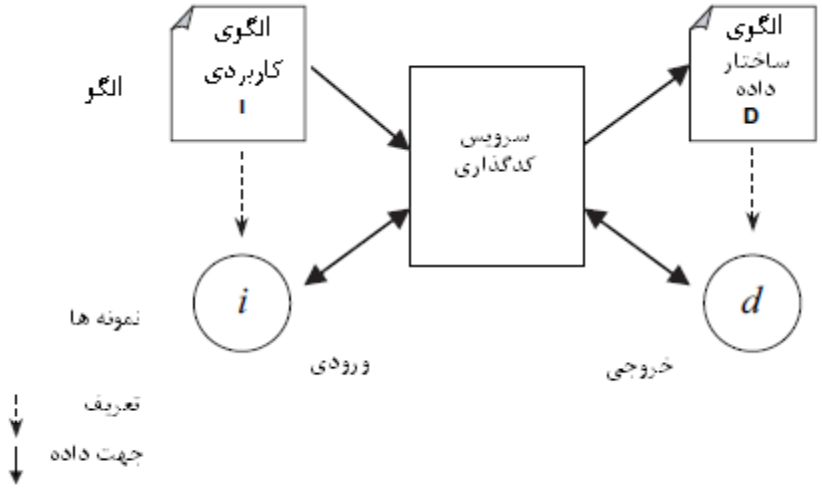
شکل۲- قاعده کدگذاری، قواعد تبدیل از مفاهیم ورودی به مفاهیم خروجی را تعیین می‌کند.

یادآوری- قواعد تبدیل صرفا براساس یک الگوی کاربردی خاص تعریف نشده و براساس دو زبان الگو تعریف شده است. این مسئله یک رویکرد عمومی است که به توسعه‌دهندگان اجازه می‌دهد سرویس‌های کدگذاری مستقل از الگوی کاربردی تهیه کنند که می‌توان آن‌ها را تا زمانی‌که در زبان الگوی مفهومی یکسانی تعریف می‌شوند برای الگوهای کاربردی مختلف، استفاده کرد.

۶-۵ سرویس کدگذاری

سرویس کدگذاری، مولفه‌ای نرم‌افزاری بوده که قاعده کدگذاری را پیاده‌سازی کرده و واسطی برای عملکردهای کدگذاری و کدگشایی فراهم می‌آورد. سرویس کدگذاری بخش یکپارچه‌سازی تبادل داده می‌باشد.

شکل۳ جزئیات یک سرویس کدگذاری و روابط آن با الگوهای ویژگی مهم ارائه می‌کند. سرویس کدگذاری باید قادر به خواندن ساختار داده ورودی و تبدیل نمونه‌ها به ساختار داده خروجی، و برعکس باشد. همچنین باید قادر به خواندن اعلان‌های الگوی کاربردی و نوشتن الگوی ساختار داده خروجی مرتبط باشد. ساختار داده ورودی توسط یک الگوی کاربردی تعریف می‌شود. الگوی کاربردی با استفاده از مفاهیم زبان الگوی مفهومی تعریف می‌شود. ساختار داده خروجی نیز با یک الگو، به نام الگوی ساختار داده، مشخص می‌شود که در آن محتوای ممکن، ساختار و الگوهای کدگذاری ساختار داده خروجی تبیین شده، تشریح می‌شود. الگوی ساختار داده با یک زبان الگو توصیف می‌شود. قاعد کدگذاری، قواعد تبدیل داده را در دو سطح تعیین می‌کند: اول در سطح الگو و دوم در سطح نمونه. در سطح الگو، قواعد تبدیل نگاشتی برای هر مفهوم تعریف شده در الگوی کاربردی را برای مفاهیم مربوطه در الگوی ساختار داده تبیین می‌کنند. در سطح نمونه، قواعد تبدیل، نگاشتی برای هر نمونه در ساختار داده ورودی را به نمونه‌های مربوطه در ساختار داده خروجی تعریف می‌کنند. نمونه‌ی قواعد تبدیل به‌طور عمومی از نتایج قواعد الگوی تبدیل می‌باشند.



شکل ۳- مرور اجمالی فرایند کدگذاری

حداقل عملکرد یک سرویس کدگذاری فراهم کردن واسطهایی برای کدگذاری و کدگشایی می‌باشد. مثال‌هایی از چنین واسطهایی برای کدگذاری d = encode (i, I) و برای کدگشایی i = decode (d, I) می‌باشند. در اینجا، i مرجعی برای یک ساختار داده خاص- الگو- کاربرد، I مرجعی برای الگوی کاربردی و d مرجعی برای ساختار داده‌ی مستقل از سامانه می‌باشد.

۶-۶ سرویس انتقال

سرویس انتقال یک مولفه نرم‌افزاری بوده که دارای یک یا چند پروتکل پیاده‌سازی شده انتقال است که امکان انتقال داده را بین سامانه‌های اطلاعاتی توزیع‌یافته برروی رسانه‌های ارتباطی برخط یا برون‌خط فراهم می‌کند. برای انتقال موفق داده بین دو سامانه، ضروری است که فرستنده و گیرنده در پروتکل انتقال مورد استفاده توافق داشته باشند.

پروتکل‌های انتقال مختلف می‌توانند تعریف شوند. یک مثال پروتکل‌های انتقال برون‌خط بوده، جایی که داده برروی رسانه‌های نوری یا مغناطیسی ذخیره شده و با استفاده از سرویس‌های پستی یا سایر سرویس‌های تحویل اختصاصی تحویل داده می‌شوند. مثال دیگر پروتکل‌های انتقال برخط بوده که داده فشرده شده و به عنوان فایل پیوست یک رایانامه، با استفاده از یک پروتکل انتقال فایل تحویل داده می‌شود یا با به‌کارگیری سایر سرویس‌های فن‌اوری اطلاعات توزیع‌یافته که متکی بر یک سرویس شبکه اصلی باشد، منتقل می‌شوند.

این استاندارد، هیچ برتری در پروتکل‌های انتقال را تعیین نمی‌کند.

۷ مجموعه نویسه

استاندارد ISO/IEC 10646 یک مجموعه تعریف‌شده بین‌المللی از نویسه‌ها به نام مجموعه نویسه بین‌المللی (UCS[1]) و الگوهای کدگذاری- نویسه‌ی آن را تعریف می‌کند. استانداردهای بین‌المللی مجموعه نویسه تبیین شده در استاندارد ISO/IEC 10646 باید در پیاده‌سازی این استاندارد استفاده شوند.

الگوهای کدگذاری- نویسه[2] که می‌توانند توسط رخ‌نمون[3]های بین‌المللی این استاندارد پشتیبانی شوند، به صورت زیر می‌باشند:

۱- قالب انتقال UCS با اندازه متغیر ۸ بیتی UTF-8؛
۲- قالب انتقال UCS با اندازه متغیر ۱۶ بیتی UTF-16؛
۳- مجموعه نویسه بین‌المللی با اندازه ثابت ۱۶ بیتی UCS-2 (از رده خارج)؛
۴- مجموعه نویسه بین‌المللی با اندازه ثابت ۳۲ بیتی UCS-4.

قواعد کدگذاری بین‌المللی که مدعی انطباق با این استاندارد بوده، باید یک یا چند مورد از این الگوهای کدگذاری- نویسه را پشتیبانی کنند. در رخ‌نمون ملی و روش‌های پیاده‌سازی سامانه، ممکن است از الگوهای کدگذاری- نویسه مختلف استفاده شود. الگوهای کدگذاری-نویسه با اندازه ثابت اغلب در پیاده‌سازی پایگاه داده استفاده شده و الگوهای کدگذاری-نویسه با اندازه متغیر بیشتر برای اهداف تبادل داده به‌کار میروند.

1- Universal Character Set
2- Character- encoding
3 - Profile

استاندارد ISO/IEC 10646 فقط مجموعه نویسه‌ها را تعیین کرده و هیچ اشاره‌ای به زبان مورد استفاده نمی‌کند.

یادآوری ۱- در مواردی که تمایز بین زبان‌های مختلف در رشته‌های متنی مهم باشد، می‌توان سازوکارهای خاصی را برای تشخیص زبان مورد استفاده، به کار گرفت.

استاندارد ISO/IEC 10646 سازوکارهایی را برای ایجاد نویسه‌های مرکب تعریف می‌کند. نویسه‌های مرکب از اضافه‌سازی یک یا چند نویسه اضافی برروی نویسه مبنا ایجاد می‌شوند. استاندارد ISO/IEC 10646 مجموعه‌ای از نویسه‌های ازپیش‌ترکیب‌شده و تجزیه آن‌ها را تعریف می‌کند. از آن‌جایی که ترکیب نمودن نویسه‌ها با معادل‌های از پیش ترکیب شده‌ی آن‌ها می‌تواند منجر به مشکلاتی در تفسیر آن‌ها گردد، استفاده از نویسه مرکب در صورت وجود نویسه از پیش ترکیب شده، از رده خارج شده است، یعنی باید همیشه از نویسه ازپیش‌ترکیب‌شده استفاده شود.

به طور خلاصه، یک قاعده کدگذاری باید:

- یک یا چند الگوی کدگذاری- نویسه را پشتیبانی کند؛
- در صورت وجود نویسه‌های ازپیش‌ترکیب‌شده معادل، از نویسه‌های مرکب استفاده نکند.

مثال: نویسه از پیش ترکیب شده ö دارای حالت تجزیه شده‌ی ¨o می‌باشد.

یادآوری ۲- برای یک توصیف بسیار دقیق از عادی‌سازی نویسه، به http://www.unicode.org/reports/tr15/ و http://www.w3.org/TR/charmod-norm/ مراجعه شود.

یادآوری ۳- UTF-16، UCS-2 و UCS-4 اطلاعاتی را برای چگونگی مرتب‌سازی بایت الزام می‌کنند.. به http://www.unicode.org/faq/utf_bom.html مراجعه شود.

۸ مدل نمونه عمومی

۸-۱ مقدمه

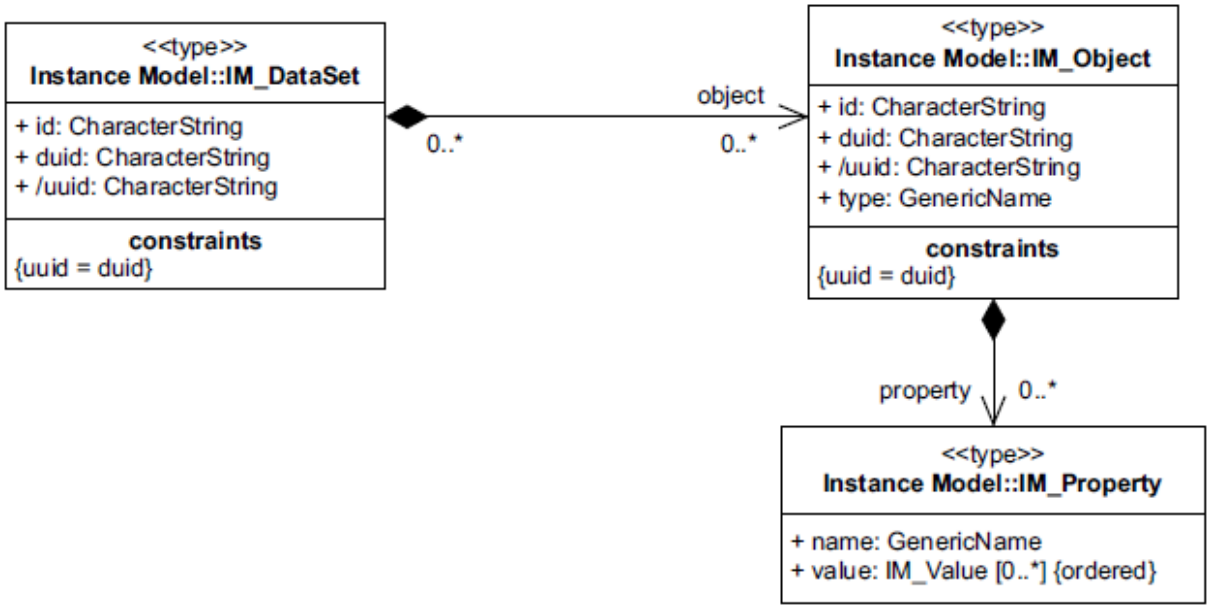
یک مدل نمونه عمومی در بند ۸ تعریف شده است. مدل نمونه یک نمایش رایج و مناسب از داده در هنگام توسعه سرویس‌های کدگذاری می‌باشد. مدل نمونه قادر به نمایش داده‌ی توصیف شده توسط الگوهای کاربردی بیان شده در UML می‌باشد. مدل نمونه ساختار داده خاص- الگو- کاربرد تشریح شده در بند ۶ (ساختارهای داده i_A و i_B در شکل ۱) را نشان می‌دهد. مدل نمونه شامل یک مجموعه داده (IM_Dataset) از یک دنباله از اشیا (IM_Object) است، که یک شی متشکل از یک دنباله از ویژگی‌ها (IM_Property) می‌باشد. ویژگی‌ها در این متن، صفت‌ها یا وابستگی‌ها می‌باشند؛ عملیات در مدل نمونه عمومی گنجانده نمی‌شود. هر ویژگی با توجه به نوع داده آن کدگذاری می‌شود. مدل نمونه در شکل‌های ۴ و ۵ نشان داده شده است.

الگوی کاربردی تعدادی از کلاس‌ها و صفت‌ها و وابستگی‌های آن‌ها را تعریف کرده و مبنایی برای نمایش داده تولیدی می‌باشد. نمایش داده (مجموعه داده) حاوی یک یا چند شی بوده که برطبق تعاریف کلاس آن‌ها ساختاردهی و کدگذاری می‌شوند. بند ۸ اصول چگونگی نمایش اشیا، صفت‌ها و وابستگی‌های بین اشیا را توصیف می‌کند.

واحد اصلی اطلاعات در یک مجموعه داده، شی می‌باشد. یک شی باید یک نمونه از یک کلاس مجزای واقعی باشد. هیچ نمونه‌ای از کلاس‌های انتزاعی و کلاس‌های کلیشه‌ای به عنوان واسط وجود ندارند. بدین‌ترتیب، ویژگی‌های تعریف شده توسط چنین کلاس‌هایی به عنوان بخشی از کلاس‌های واقعی، که از ویژگیهای آن ارث برده یا آن را محقق می‌کنند، کدگذاری می‌شود. هر کلاس باید در الگوی کاربردی دارای یک نام منحصر به فرد باشد. الگوی کاربردی ممکن است به کلاس‌های تعریف شده در الگوهای استاندارد شده یا سایر الگو-های کاربردی ارجاع کرده و یا از آن‌ها استفاده بکند. اعلان این کلاس‌ها یا باید در مدل UML که حاوی الگوی کاربردی بوده، یا با همراهی الگوی کاربردی به عنوان یک فایل مجزا صورت پذیرد.

یک شی باید حاوی یک مجموعه از مقادیر باشد. کلاس شی، ویژگی‌ها را تعریف کرده و می‌تواند این خصوصیات را یا از ابرنوع «class» ارث ببرد یا درون خود کلاس‌ها تعریف شوند. به منظور تمایز بین ویژگی-های مختلف، هر ویژگیی باید دارای یک نام منحصر به فرد در آن کلاس باشد. نوع ویژگی داده، مقادیر ممکن را کنترل کرده و عبارت‌های چندگانه نشان‌دهنده‌ی تعداد نمونه‌های ویژگی در یک شی مورد نظر هستند.

یک شی دارای یک کلاس مرتبط بوده، که در یک الگوی کاربردی یا الگوی استاندارد شده تعریف می‌شود، که صفت‌های ممکن و وابستگی‌های آن را که از ملزومات نمایش آن شی می‌باشد، تعریف می‌کند. یک IM_Object با صفت «class» به کلاس آن اشاره می‌کند، این مورد باید در متن یک مجموعه داده توسط شناسه منحصر به فرد آن «id» شناسایی شده، و ممکن است به طور کلی به صورت منحصر به فرد در یک جهان، دامنه کاربردی یا فضای نام، به وسیله ویژگی «duid» آن شناسایی شود.

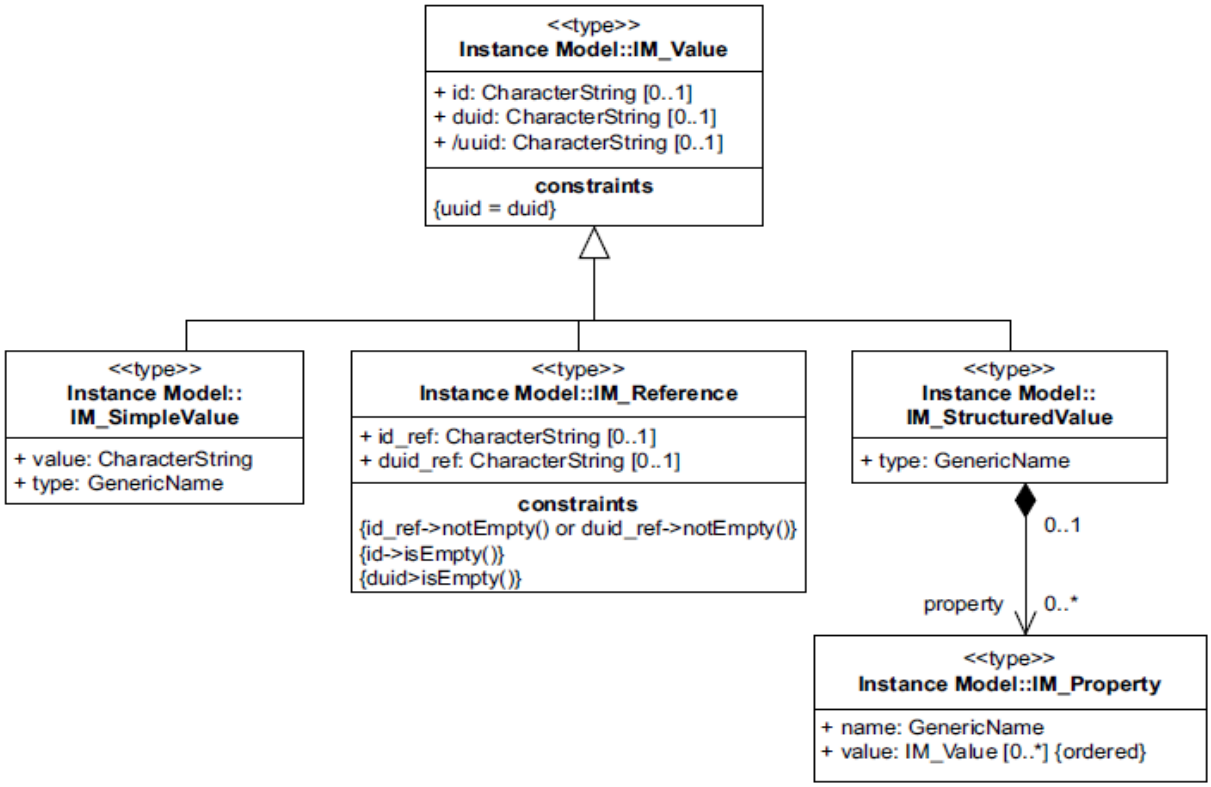


شکل ۴- مدل نمونه- مجموعه داده، شی و ویژگی

مشخصه‌های تعریف شده توسط کلاس و وابستگی‌ها به صورت قابل انتقال از کلاس با نگاشت به یک سری از ویژگی‌ها می‌باشند. یک ویژگی (IM_Property) یک نام با یک مجموعه مرتب‌شده از مقادیر را نمایش می‌دهد. این ویژگی می‌تواند یک صفت یا انتهای یک وابستگی را نشان دهد. نام ویژگی باید مرتبط با نام صفت یا نام نقش هدف یک وابستگی باشد. یک مقدار (IM_Value) مقدار ویژگی را نمایش می‌دهد.

مقادیر پوچ[1] ممکن است به صورت صریح یا ضمنی داده شوند. مقدار پوچ صریح باید توسط یک نمونه مربوط به IM_Property با مقدار nilReason معین نشان داده شود. اگر نمونه IM_Property مربوطه مفقود باشد، به صورت یک مقدار پوچ ضمنی نشان داده می‌شود.

1 -Null



شکل ۵- مدل نمونه- انواع مقدار

سه نوع مقدار به صورت زیر تعریف می‌شوند:

- IM_SimpleValue یک مقدار با محتوای ساده را نشان می‌دهد.

مثال: یک عدد صحیح یا یک رشته نویسه

- IM_Reference یک پیوند یا مرجع به شی هدف ارائه می‌کند. شی هدف ممکن است در واحد انتقال مشابه یا یک واحد دیگر واقع شده باشد. یک شناسه منحصربه‌فرد (id_ref) یک شی واقع شده در واحد انتقال مشابه را نشانه‌گذاری می‌کند. یک شناسه منحصربه‌فرد دامنه (duid_ref) یک شی واقع در متن دامنه‌ی کاربردی را نشانه‌گذاری می‌کند.
- IM_StructuredValue یک مقدار نوع داده با محتوای مرکب را نشان می‌دهد [یک دنباله از خصوصیات (IM_Property)].

ممکن است یک شی، به خاطر وابستگی‌هایش، به یک یا چند شی متصل (ارجاع) شود. UML سه نوع رابطه وابستگی مختلف، به نام‌های وابستگی[1]، تجمیع[2] و ترکیب[3]، با قواعد معنایی متفاوت را تعریف می‌کند؛ برای

1- Association
2- Aggregation
3- Composition

جزئیات بیشتر به استاندارد ISO/TS 19103 مراجعه شود. به طور کلی، دو راهبرد برای نمایش وابستگی‌ها وجود دارد: یا به عنوان بخش متصل[1] به عنوان قسمتی از اشیاء یا بخش جدا به‌عنوان وابستگی[2] مجزای اشیاء یا جداول ارتباطی[3].

راهبرد نمایش متصل، یک وابستگی را به ویژگی شی منبع و ویژگی شی هدف تقسیم می‌کند. این دو ویژگی، هر نقطه را به شی دیگر متصل می‌کنند. ویژگی اتصال، حاوی مراجعی برای اشیا هدف خود، یا خود اشیاء هدف در حالت ترکیب می‌باشد. یک ویژگی اتصال توسط نام نقش نزدیک به کلاس هدف شناسایی شده و دارای چندگانگی مربوطه می‌باشد. اگر نام نقش مفقود شده یا وابستگی از شی منبع قابل ردیابی نباشد، در این صورت هیچ ویژگی اتصالی وجود نخواهد داشت. دو مقدار ویژگی اتصال باید مطابق با محدودیت یکپارچگی مرجعی محدود شده باشد. در صورتی که یک شی منبع از طریق یک ویژگی اتصال به شی هدف دیگری اشاره داشته و شی هدف نیز دارای یک وابستگی دو طرفه با شی منبع باشد، شی هدف باید دارای یک ویژگی اتصال مرتبط باشد که به شی منبع قابل ارجاع باشد. اگر یک ویژگی اتصال به عنوان یک مرجع یا یک شی تعبیه شده کدگذاری شده باشد یا به طور کلی کدگذاری نشده باشد، توسط یک قاعده کدگذاری مجزا تعریف می‌شود (همانند پیوست **الف**).

۸-۲ ارتباط بین UML و مدل نمونه

جدول ۳ خلاصه‌ای از ارتباط UML و مدل نمونه را ارائه می‌کند.

اشیائی که برمبنای کلاس‌های دارای اَبَر نوع می‌باشند، باید حاوی تمام ویژگی‌ها و نقاط پایانی وابستگی‌های کلاسها و اَبَرنوع‌هایشان باشند. بدین‌ترتیب، تمام صفت‌ها و نقاط پایانی وابستگی‌ها باید از اَبَر نوع‌ها نسخه-برداری شده و به عنوان بخشی از شی در نظر گرفته شوند. صفت‌ها و اسم نقاط پایانی وابستگی‌ها باید در مسیر دسترسی به مقادیر اطلاعات صفت‌ها بوده و بنابراین درون کلاس‌ها به‌صورت‌منصربه‌فرد باشند.

عملیات و قیدها نباید در مدل نمونه نگاشت شوند.

1- Encapsulated
2- Association
3- Relational tables

جدول ۳- خلاصه‌ای از ارتباط بین UML و مدل نمونه

مفهوم UML	مدل نمونه
Package	N/A[a]
Class Stereotype <<Interface>> <<DataType>> <<Union>> <<Enumeration>> <<CodeList>> <<Type>> <<FeatureType>> None هر نوع کلیشه‌ی دیگری به عنوان تعریف شده توسط رخ‌نمون UML مورد استفاده	 IM_Object IM_SimpleValue یا IM_StructuredValue IM_StructuredValue IM_SimpleValue IM_SimpleValue یا IM_StructuredValue IM_Object IM_Object IM_Object همانطور که توسط ویژگیهای قاعده کدگذاری تعریف شده است
ویژگی توصیفی	با توجه به نوع ویژگی توصیفی IM_Value یا IM_Property یا IM_SimpleValue, IM_Reference یا IM_StructuredValue
وابستگی	IM_Property یا IM_Reference
تجمیع	IM_Property یا IM_Reference
ترکیب	با توجه به نوع هدف یا IM_Value یا IM_Property یا IM_Reference (target type is a class) یا IM_SimpleValue یا IM_StructuredValue (نوع هدف یک نوع داده می‌باشد)
عمومی‌سازی	شی با توجه به زیر کلاس، تمام ویژگی‌های زیر کلاس را که از کلاس بالاتر به ارث برده، دارا می‌باشد
عملیات	N/A
قیود	N/A
a N/A به معنی عدم کاربرد می‌باشد	

۹ قواعد کدگذاری

۹-۱مقدمه

الزامات تعیین قواعد کدگذاری در زیربندهای ۹-۲ تا ۹-۶ تعریف شده‌اند. قاعده کدگذاری، قواعد تبدیل داده از ساختار داده ورودی به ساختار داده خروجی را توصیف می‌کند. الگوها باید برای هردو ساختار داده‌ی ورودی و خروجی تعریف شوند. الگوی ساختار داده ورودی، الگوی کاربردی نامیده می‌شود.

یک قاعده کدگذاری به طور عمومی باید موارد زیر را تعیین کند:

الف- الزامات کدگذاری عمومی (به زیربند ۹-۲ مراجعه شود):

۱- الگوهای کاربردی و زبان الگو
۲- ترتیب بیت‌ها در هر بایت و بایت‌ها در یک کلمه (هرجا که کاربرد داشته باشد)،
۳- مجموعه نویسه و کدگذاری
۴- فراداده تبادل ضروری
۵- قرارداد شناسایی مجموعه داده و شی

ب- ساختار داده ورودی (به زیربند ۹-۳ مراجعه شود):

۱- ساختار داده استفاده شده برای عبور داده مطابق الگوی کاربردی (ساختارهای داده i_A و i_B در شکل۱) به سرویس کدگذاری، مدل نمونه نامیده می‌شود.
۲- چگونه مدل نمونه با الگوی کاربردی مرتبط می‌شود؛

پ- ساختار داده خروجی استفاده شده، که فرمت تبادل (به زیربند ۹-۴ مراجعه شود) نامیده می‌شود؛

ت- قواعد تبدیل، برای تبدیل داده در مدل نمونه به فرمت تبادل (به زیربند ۹-۵ مراجعه شود) که نگاشت نامیده می‌شود:

۱- قواعد تبدیل برای کدگذاری
۲- درصورت لزوم، قواعد تبدیل برای کدگشایی؛

ث- مثال‌های کافی از داده انتزاعی، کاربرد قواعد تبدیل و داده‌ی کدگذاری شده (به زیربند ۹-۶ مراجعه شود).

۹-۲ الزامات عمومی کدگذاری

۹-۲-۱ الگوی کاربردی و زبان الگو

قاعده کدگذاری باید زبان الگوی استفاده شده برای تعریف الگوهای کاربردی را تعیین کرده و نحوه سازماندهی الگوی کاربردی را توصیف کند.

۹-۲-۲ مرتب‌سازی بیت و بایت

اگر قاعده کدگذاری یک کدگذاری دودویی را تعیین کند، باید ترتیب بیت‌ها در هر بایت و ترتیب و تعداد بایت‌ها در هر ساختار چندبایتی (کلمه) را تعیین کند.

یادآوری- اینها قواعد عمومی بوده که به متن و قواعد کدگذاری دودویی اعمال می‌شوند. حتی کدگذاری‌های مبتنی بر متن (همانند UCS-4 ,UCS-2, UTF-16) الزام به تعیین ترتیب بایت می‌باشند. (به http://www.unicode.org/faq/utf_bom.html مراجعه شود).

۹-۲-۳ مجموعه نویسه و کدگذاری

مجموعه نویسه، نویسه‌های مجاز را تعریف می‌کند. مجموعه‌ی نویسه زبان داده را تعریف نمی‌کند. همان مجموعه‌ی نویسه می‌توانند به شیوه‌های مختلف کدگذاری شود.

اگر بیش از یک زبان الزام باشد، الگوی کاربردی یک ویژگی توصیفی را باید به شکلی مدل کند که زبان واقعی متن کدگذاری شده قابل شناسایی باشد.

یادآوری- به کلاس LocalizedCharacterString دراستاندارد ISO/TS 19103 مراجعه شود.

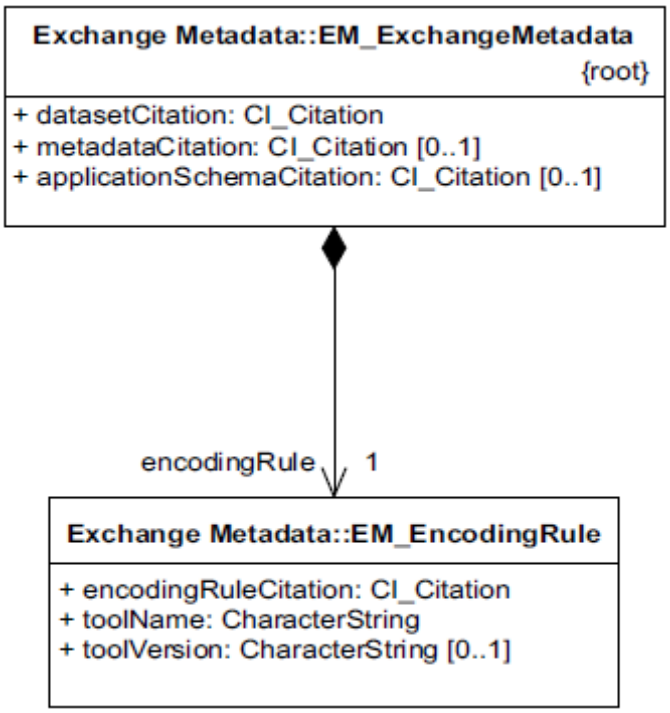
قاعده کدگذاری باید مجموعه نویسه‌ی مورد استفاده و کدگذاری نویسه‌ها را تعیین کند.

۹-۲-۴ فراداده تبادل

فراداده تبادل، فراداده‌ایی درباره ساختار داده کدگذاری‌شده می‌باشند. این اطلاعات ممکن است تولیدکننده مجموعه داده را توصیف کرده، به اطلاعاتی درخصوص مجموعه داده ارجاع داده باشد، به الگوی کاربردی ارجاع داده باشد و اطلاعاتی در مورد قاعده کدگذاری به کار گرفته شده فراهم نمایند.

قاعده کدگذاری باید فراداده تبادل را مشخص کند و چگونگی همراهی آن با ساختار داده کدگذاری شده را تعیین کند.

شکل ۶ مثالی از کلاس فراداده‌تبادل (ExchangeMetadata) و کلاس مربوطه قاعده‌کدگذاری (EncodingRule) آن را نشان می‌دهد. کلاس CI_Citation در منبع [۲۱] کتاب‌نامه تعریف شده است. یک فایل تبادل به طور معمول حاوی یک نمونه از کلاس ExchangeMetadata می‌باشد. صفت «datasetCitation» حاوی اطلاعاتی درباره تولیدکننده مجموعه داده بوده، صفت «metadataCitation» ممکن است حاوی اطلاعاتی درباره محل یافتن فراداده درباره مجموعه داده باشد، و صفت «applicationSchemaCitation» شامل مرجعی به الگوی کاربردی می‌باشد. صفت «encodingRuleCitation» قاعده کدگذاری، قاعده کدگذاری به کار رفته را شناسایی کرده و چگونگی به کارگیری آن در این مورد به‌خصوص را ارائه می‌کند. همچنین ممکن است حاوی اطلاعاتی درباره ابزار سرویس کدگذاری استفاده شده برای کدگذاری مجموعه داده نیز باشد به نحوی که اقلام«toolName» و «toolVersion» نام ابزار سرویس کدگذاری و شماره نسخه آن را نشان می‌دهند.



شکل ۶- مثالی از فراداده تبادل

۹-۲-۵ واحد انتقال

۹-۲-۵-۱ گرانولاریتی[1] و ساختار

تعریف گرانولاریتی و ساختار واحد انتقال به نحوی که به طور موثر کدگذاری و کدگشایی شود، مهم می‌باشد. در اینجا یک شی به عنوان واحد اصلی اطلاعات در نظر گرفته شده و لازم است که انواع مختلف اشیا شناسایی شوند. اشیا در یک واحد انتقال ممکن است به صورت ترتیبی و/یا سلسله مراتبی ساختاردهی شوند. یک شی ممکن است به صورت داخلی به عنوان یک دنباله از صفت‌ها ساختاردهی شود، یا حاوی ارجاع‌هایی به اشیا دیگر بوده و همچنین ممکن است متشکل از سایر اشیا در یک شیوه سلسله مراتبی باشد.

یک شی ممکن است به قطعات مختلف تقسیم شده، یا ویژگی‌های آن ممکن است با اشیا دیگر ادغام شود. قاعده کدگذاری تعیین کننده هر تغییری در ساختار بوده، و تبدیل نمونه باید به روشنی برگشت‌پذیر باشد.

قاعده کدگذاری باید موارد زیر را تعیین کند:

- شی و انواع مختلف شی؛
- ساختار یک شی؛
- ساختار یک مجموعه داده.

1- Granularity

۹-۲-۵-۲ شناسایی شی

می‌توان به اشیا شناسه‌هایی تخصیص داد که اجازه شناسایی منحصر به فرد آن‌ها را در یک زمینه خاص می-دهد. دو زمینه مختلف باید در نظر گرفته شوند:

الف- اولین زمینه، واحد انتقال می‌باشد. در اینجا، شناسه‌های شی در یک واحد خاص انتقال داده منحصربه-فرد می‌باشند. این شناسه‌ها به اشیا اجازه می‌دهند تا به سایر اشیا درون آن واحد انتقال ارجاع دهند. شناسه‌ها ممکن است وقتی که وارد واحد انتقال شده و ماهیت گذرا بگیرند، به اشیا تخصیص داده شوند.

ب- دومین زمینه، دامنه‌ی کاربردی می‌باشد. دامنه‌ی کاربردی یک فضا و قرارداد شناسایی به نام شناسه‌های منحصر به فرد دامنه (DUIDها) را تعریف می‌کند. وقتی که DUID ایجاد شود به شی تخصیص داده می‌شود و در کل طول عمر شی به‌صورت پایدار می‌باشد. DUIDی یک شی حذف شده، نباید مجددا مورد استفاده قرار گیرد. DUID اغلب با هدف پیاده‌سازی به‌کار می‌رود. DUIDها برای مدیریت بلند مدت داده‌ی توزیع‌یافته و برای تحقق سازوکارهای به‌روزرسانی الزامی می‌باشند. این شناسه‌ها، شناسه-های پایدار نیز نامیده می‌شوند. ممکن است از یک مخدوم[1] اسم خاص برای تجزیه و تحلیل شناسه‌های پایدار استفاده شود. شناسه‌ها باید به‌صورت‌منحصربفرد در یک فضای محدود خوب تعریف شده توسط یک دامنه کاربردی، قرار داشته باشند.

دامنه‌ی کاربردی ممکن است از ویژگی‌های اضافی برای شناسایی پدیده‌های جهان واقعی که توسط شی نمایش داده می‌شود (برای مثال شماره بسته[2])، استفاده کند. این ویژگی‌های اضافی به عنوان شناسه‌های شی در حوزه این مشخصه در نظر گرفته نمی‌شوند.

قاعده کدگذاری ممکن است طرح داده را تغییر دهد، برای مثال فاکتورگیری[3] از رشته نویسه‌های مشترک. برای پشتیبانی از این مسئله، مطابق با الگوی کاربردی، انواع بدون شناسایی، می‌تواند شامل یک شناسه‌ی بخش باشد. این شناسه‌های بخش در هر موردی گذرا بوده (تنها در واحد انتقال معتبر می‌باشند) و هرگز یک DUID نمی‌باشند.

قواعد کدگذاری باید شامل موارد زیر باشد:

- سازوکارهای مختلف شناسایی شی استفاده شده؛
- ساختار درونی آن‌ها.

1- Server
2- Parcel
3-Factoring

۹-۳ ساختار داده ورودی

ساختار داده ورودی، مدل نمونه نامیده شده و ابزاری برای استدلال درباره داده کاربردی، تعریف قواعد تبدیل و بیان مثال‌ها می‌باشد. مدل نمونه باید قابلیت نشان دادن داده مطابق با ویژگی‌های الگوی کاربردی را داشته باشد. مدل نمونه ممکن است ویژه یک الگوی کاربردی خاص باشد یا قادر به نمایش داده مطابق با هر الگویی باشد. مدل نمونه ممکن است به صورت خلاصه بوده که اجباری به پیاده سازی آن برای درک سرویس کدگذاری نمی‌باشد.

قاعده کدگذاری باید همانطور که در بند ۸ تعریف شده به مدل نمونه عمومی ارجاع بدهد یا یک مدل نمونه و رابطه‌ی آن با الگوی کاربردی را تعیین کند.

این استاندارد اجباری به پیاده‌سازی مدل نمونه خاص یا چگونگی عبور داده از/ به سرویس کدگذاری ندارد.

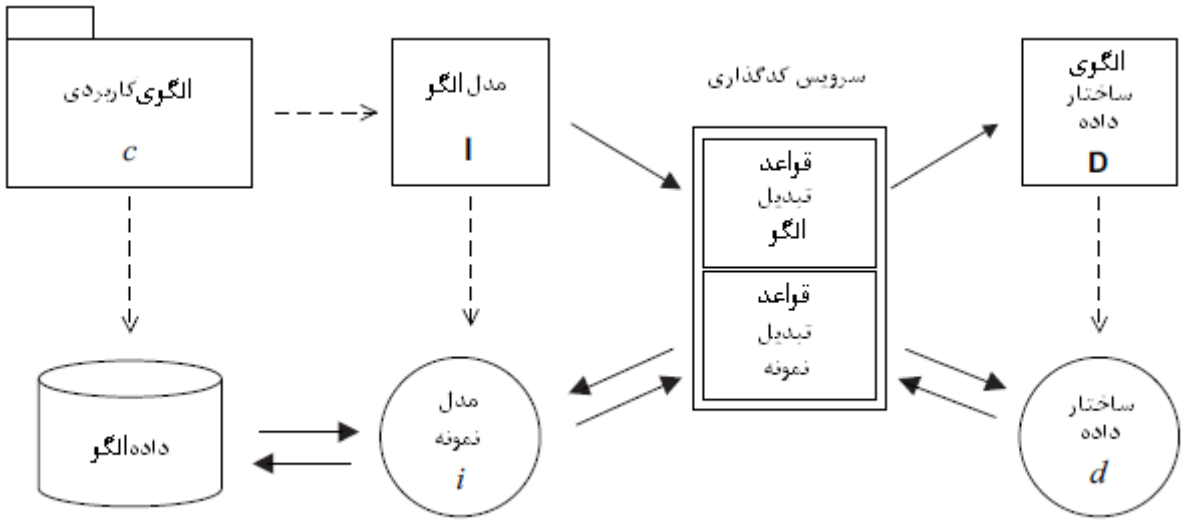
۹-۴ ساختار داده خروجی

ساختار داده خروجی، نحوه ساختاردهی و نمایش داده در یک فایل تبادل را تبیین می‌کند. یک الگو ممکن است همراه با ساختار داده خروجی باشد.

قاعده کدگذاری باید ساختار داده خروجی، و در صورت وجود، الگوی ساختار داده خروجی را تعیین کند.

۹-۵ قواعد تبدیل

یک قاعده تبدیل چگونگی تبدیل نمونه داده در ساختار داده ورودی به نمونه داده در ساختار داده خروجی را تعیین می‌کند. دو مجموعه از قواعد تبدیل ممکن است موجود باشند. مجموعه اول، مجموعه قواعد تبدیل الگو بوده که نگاشتی از الگوی UML به الگوی ساختار داده خروجی را تبیین می‌کند. مجموعه دوم، مجموعه قواعد تبدیل نمونه بوده که نگاشتی از نمونه‌های موجود در مدل نمونه به نمونه‌هایی در مدل داده حاصل را تعریف می‌کند. شکل۷ قواعد تبدیل مختلف را نشان می‌دهد.



شکل ۷- قواعد تبادل

قاعده کدگذاری باید موارد زیر را تعیین کند:

- قواعد تبدیل الگو؛
- قواعد تبدیل نمونه.

۹-۶ مثال‌ها

مثال‌ها برای درک قواعد تبدیل و برای آزمون سرویس‌های کدگذاری مهم هستند.

قاعده کدگذاری باید موارد زیر را فراهم آورد:

- مثال‌هایی که قواعد تبدیل را با شکل نشان می‌دهند؛
- داده‌ی آزمایشی که می‌توانند در پیاده‌سازی سرویس‌های کدگذاری استفاده شوند.

۱۰ سرویس کدگذاری

سرویس کدگذاری یک مولفه نرم‌افزاری بوده که قاعده کدگذاری را پیاده‌سازی کرده و یک واسط برای عملکرد آن سرویس فراهم می‌کند.

یک سرویس کدگذاری باید واسطی برای عملکرد آن سرویس از طریق یک یا چندتا از ویژگی‌های واسط فراهم نماید.

یک سرویس کدگذاری باید یک یا چند مورد از موارد زیر را فراهم کند:

- قابلیت کدگذاری داده مطابق با قواعد تبدیل نمونه؛

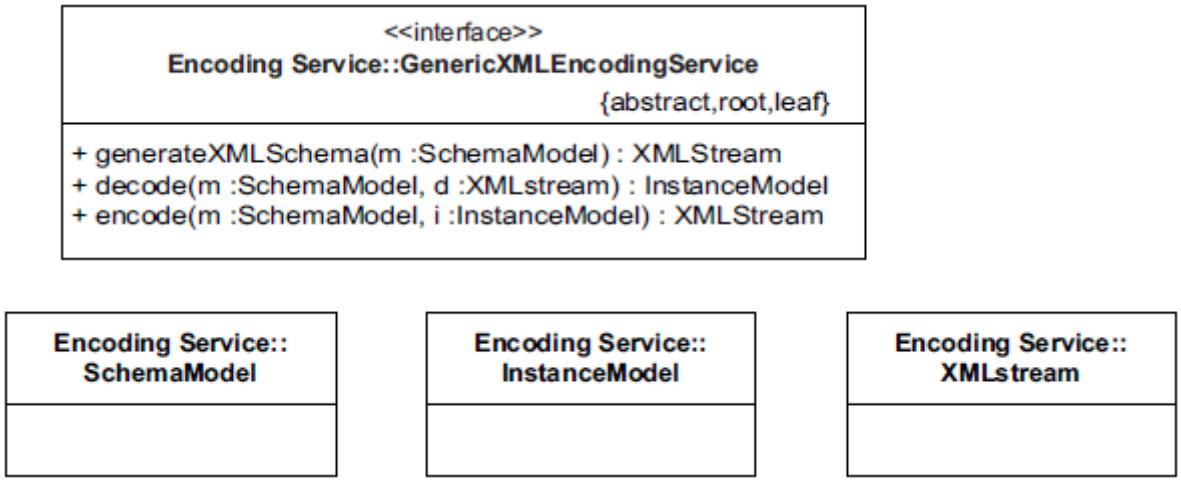
- قابلیت کدگشایی داده مطابق با قواعد تبدیل نمونه؛
- قابلیت ایجاد الگوی ساختار داده خروجی مطابق قواعد تبدیل الگو.

این استاندارد واسط خاصی را برای سرویس کدگذاری تعیین نمی‌کند.

یک سرویس کدگذاری ممکن است تنها قادر به کدگذاری یا کدگشایی باشد.

شکل ۸ مثالی از یک سرویس کدگذاری مبتنی بر XML می‌باشد.

یادآوی- شکل ۸ سرویس کدگذاری با یک واسط مشخص را نشان می‌دهد. این سرویس سه عملیات زیر را پشتیبانی می‌کند: generateXMLSchema (تولید الگوی XML)، کدگذاری و کدگشایی. عملیات «generateXMLSchema» می‌تواند برای تولید یک فایل الگوی XML استفاده شود. این عملیات از یک مدل الگو به عنوان پارامتر ورودی استفاده کرده و یک شی XMLStream را به عنوان نتیجه تولید می‌کند. عملیات کدگذاری «encode» می‌تواند برای تولید یک سند XML استفاده شود. این عملیات یک مدل الگو و یک مدل نمونه را به عنوان پارامترهای ورودی دریافت کرده و یک شی XMLStream را به عنوان خروجی برگشت می‌دهد. عملیات کدگشایی «decode» می‌تواند برای تفسیر یک سند XML استفاده شود. این عملیات یک مدل الگو و یک شی XMLStream را به عنوان ورودی دریافت کرده و یک مدل نمونه را برگشت می‌دهد.



شکل ۸- مثالی از واسط سرویس کدگذاری

پیوست الف

(الزامی)

قاعده کد گذاری بر مبنای XML

الف-۱ کدگذاری XML در این استاندارد باید مطابق با زبان نشانه‌گذاری توسعه پذیر(XML) 1.0 باشد. هیچ مجموعه‌ی ثابتی از الزامات کدگذاری که باعث شود قاعده کدگذاری فایلهای XML مبنا برای تمام الگوها صدق کند، وجود ندارد. در نتیجه، این استاندارد یک قاعده تبدیل الگوی خاصی را تعیین نمی‌کند. هر قاعده کدگذاری بر مبنای XML باید شرایط زیر را دارا باشد:

- ساختار داده ورودی، رخ‌نمون از UML که به طور عمومی توسط قاعده کدگذاری به‌کار رفته، را تعیین و یا شناسایی کند؛
- ساختار داده‌ی خروجی XML را تعیین کند؛
- قوانین تبدیل الگو، بین ساختارهای داده ورودی و خروجی را تعیین کند؛ و
- با تمام الزاماتی که در این پیوست مشخص شده مطابقت داشته باشد.

الف-۲ موارد استفاده متفاوت منجر به قواعد کدگذاری مختلف می‌شود. در حال حاضر، مجموعه استانداردهای بین‌المللی که با نام مجموعه استانداردهای «ISO 19100» شناخته می‌شوند، دو قاعده کدگذاری بر مبنای XML را تعیین می‌کند:

- پیوست E از منبع [۲۷] کتاب‌نامه یک قاعده کدگذاری بر مبنایXML را برای الگوهای کاربردی مطابق با استاندارد ISO 19109 تعیین کرده که این الگوها می‌توانند با استفاده از یک رخ‌نمون محدود از UML که امکان تبدیل به الگوی XML را ایجاد می‌کند، ارائه شود. قاعده کدگذاری به‌طورکلی به منظور مشخص کردن الگوهای کاربردی انواع عارضه‌ها و ویژگی‌های آن‌ها توسعه یافته است. قاعده کدگذاری از الگوی XML به عنوان الگوی ساختار داده خروجی استفاده می‌کند.

- منبع [۲۸] کتاب‌نامه یک قاعده کدگذاری بر مبنای XMLرا برای الگوهای مفهومی که انواع منابع جغرافیایی مانند فراداده، مطابق با منبع [۲۱] کتاب‌نامه و کاتالوگ‌های عارضه مطابق با منبع [۱۶] کتاب-نامه را تعریف می‌کند، مشخص می‌کند. قاعده کدگذاری از رخ‌نمون UML، همانطور که در مدل‌های UML به‌کار رفته در استانداردهای توسعه یافته توسط کمیسیون فنی استاندارد ISO/TC 211 مورد استفاده واقع شده، پشتیبانی می‌کند. قاعده کدگذاری از الگوی XML برای الگوی ساختار داده خروجی استفاده می‌کند.

الف-۳ قواعد متفاوت کدگذاری بر مبنای XML ممکن است توسط یک جامعه اطلاعاتی الزامی باشند و مشخص شوند. نمونه‌هایی از این الزامات شامل موارد زیر می‌باشند ولی بطور کلی محدود به این موارد نیستند:

- پشتیبانی از قاعده کدگذاری بر مبنای XML که در پیوست **پ** مشخص شده است (این قاعده کدگذاری مورد استفاده جامعه‌های مختلف می‌باشد)؛
- پشتیبانی از یک رخنمون UML متفاوت که تحت پوشش قاعده کدگذاری بر مبنای XML موجود نباشد؛
- پشتیبانی از یک الگوی ساختار داده خروجی، به غیر از الگوی XML (برای مثال Relax NG)؛
- پشتیبانی از فناوری‌های جدید XML یا نسخه‌های جدید فناوری‌های XML موجود؛
- پشتیبانی از تبدیلهای خاص برای بهینه‌سازی استفاده از قابلیت‌های XML؛
- پشتیبانی از سایر الزامات مرتبطXML که در یک اجتماع ایجاد می‌شوند.

پیوست ب

(الزامی)

تسلسل آزمون انتزاعی

ب-۱ موارد آزمون یک قاعده کدگذاری

ب-۱-۱ کلیات

تمام قواعد کدگذاری باید تمامی موارد آزمون مجموعه آزمون انتزاعی که در زیربندهای ب-۱ تا ب-۳، آمـده مطابق با کلاس‌های انطباق معرفی شده در زیربند۲-۲، را بگذرانند.

ب-۱-۲ مستند سازی قواعد تبدیل

الف- هدف آزمون: بررسی این موردکه قواعد کدگذاری قواعد تبدیل الگو یا نمونه را تعریف می‌کنند.

ب- روش آزمون: بازرسی مستندات قاعده کدگذاری.

پ- مرجع: زیربند ۹-۵.

ت- نوع آزمون: اصلی.

ب-۱-۳ نمونه و قواعد تبدیل سازگار

الف- هدف آزمون: بررسی اینکه، اگر قاعده کدگذاری، الگو و قواعد تبدیل نمونه را تبیـین کـرده باشـد، آنهـا سازگار می‌باشند

ب- روش آزمون: بازرسی مستندات قاعده کدگذاری.

پ- مرجع: زیربند ۹-۵.

ت- نوع آزمون: قابلیت.

ب-۱-۴ ترتیب بیت[1] و بایت[2]

الف- هدف آزمون: بررسی اینکه قاعده کدگذاری، ترتیب بیت و بایت را تعیین یا فراداده را برای تعیین آن در زمان اجرا ، تعریف کرده باشد.

1- Bit
2- Byte

ب- روش آزمون: بازرسی مستندات قاعده کدگذاری.

پ- مرجع: زیربندهای ۹-۲-۲ و ۹-۲-۴.

ت- نوع آزمون: قابلیت.

ب-۱-۵ مجموعه نویسه و کدگذاری

الف- هدف آزمون: بررسی اینکه قاعده کدگذاری مجموعه نویسه و کدگذاری آن را مشخص، یا فراداده را برای تعیین آن در زمان اجرا تعریف کرده باشد.

ب- روش آزمون: بازرسی مستندات قواعد کدگذاری.

پ- مرجع: زیربندهای ۹-۲-۳ و ۹-۲-۴.

ت- نوع آزمون: قابلیت.

ب-۱-۶ قرارداد شناسایی

الف- هدف آزمون: بررسی اینکه قاعده کدگذاری یک قرارداد شناسایی را تعیین یا فراداده را برای تعیین آن در زمان اجرا تعریف کرده باشد.

ب- روش آزمون: بازرسی مستندات قواعد کدگذاری.

پ- مرجع: زیربندهای ۹-۲-۴ و ۹-۲-۵-۲.

ت- نوع آزمون: قابلیت.

ب-۱-۷ فرادادهی کدگذاری

الف- هدف آزمون: بررسی اینکه قاعده کدگذاری فرادادهای را برای تعیین جنبههای کدگذاری در زمان اجرا، که مقادیر ثابتی در قاعده کدگذاری نیستند، تعریف کرده باشد.

ب- روش آزمون: بازرسی مستندات قاعده کدگذاری.

پ- مرجع: زیربندهای ۹-۲-۲، ۹-۲-۳، ۹-۲-۴،۹-۲-۵-۲ و زیربند ۹-۵.

ت- نوع آزمون: قابلیت.

ب-۲ موارد آزمون برای قواعد تبدیل نمونه

ب-۲-۱ مستند سازی تبدیل نمونه

الف- هدف آزمون: بررسی اینکه قاعده کدگذاری نحوه نگاشت نمونه‌های مدل نمونه عمومی به فرمت انتقال را تعریف کرده باشد.

ب- روش آزمون: بازرسی مستندات قاعده کدگذاری.

پ- مرجع: زیربند ۹-۵.

ت- نوع آزمون: قابلیت.

ب-۲-۲ کامل بودن تبدیل نمونه

الف- هدف آزمون: بررسی اینکه قاعده کدگذاری قواعد تبدیل نمونه را برای مدل نمونه عمومی کامل تعریف کرده باشد.

ب- روش آزمون: بازرسی مستندات قاعده کدگذاری.

پ- مرجع: بند ۸ و زیربند ۹-۵.

ت- نوع آزمون: قابلیت.

ب-۲-۳ تبدیل نمونه واضح

الف- هدف آزمون: بررسی اینکه قاعده کدگذاری قواعد تبدیل نمونه را طوری تعریف کرده باشد که یک نمونه از مدل نمونه عمومی به فرمت انتقال متناظر نگاشت شود و مجددا برگشت این انتقال بدون از دست دادن اطلاعات انجام پذیرد.

ب- روش آزمون: بازرسی مستندات قاعده کدگذاری.

پ- مرجع: زیربند ۹-۲-۵-۱.

ت- نوع آزمون: قابلیت.

ب-۳ : موارد آزمون برای قواعد تبدیل الگو

ب-۳-۱ مستند سازی تبدیل الگو

الف- هدف آزمون: بررسی اینکه قاعده کدگذاری چگونگی نگاشت الگوی کاربردی به الگوی فرمت انتقال را تعریف کرده باشد.

ب- روش آزمون: بازرسی مستندات قاعده کدگذاری.

پ- مرجع: زیربند ۹-۵.

ت- نوع آزمون: قابلیت.

ب-۳-۲ کامل بودن تبدیل الگو

الف- هدف آزمون: بررسی اینکه قاعده کدگذاری قواعد تبدیل الگو را برای رخ‌نمون UML کامل، که توسط استاندارد ISO/TS 19103 یا یک رخ‌نمون از آن تبیین شده باشد، تعریف کرده باشد.

ب- روش آزمون: بازرسی مستندات قاعده کدگذاری.

پ- مرجع: زیربند ۹-۵.

ت- نوع آزمون: قابلیت.

ب-۳-۳ تبدیل نمونه واضح

الف- هدف آزمون: بررسی اینکه قاعده کدگذاری قواعد تبدیل نمونه را طوری تعریف کرده باشد که یک نمونه از مدل نمونه عمومی به فرمت انتقال متناظر نگاشت شود و مجددا برگشت این انتقال بدون از دست دادن اطلاعات انجام پذیرد.

ب- روش آزمون: بازرسی مستندات قاعده کدگذاری.

پ- مرجع: زیربند ۹-۲-۵-۱.

ت- نوع آزمون: قابلیت.

ب-۴ موارد آزمون برای یک سرویس کدگذاری

ب-۴-۱ مستند سازی یک واسط سرویس

الف- هدف آزمون: بررسی اینکه سرویس کدگذاری یک واسط مستندسازی شده را فراهم کرده باشد.

ب- روش آزمون: بازرسی مستندات سرویس کدگذاری

پ- مرجع: بند ۱۰.

ت- نوع آزمون: اصلی.

ب-۴-۲ ارجاع به قاعده کدگذاری

الف- هدف آزمون: بررسی اینکه مستندسازی سرویس کدگذاری به قاعده کدگذاری که پیاده‌سازی می‌کند، ارجاع داده باشد.

ب- روش آزمون: بازرسی مستندات سرویس کدگذاری.

پ- مرجع: بند ۱۰.

ت- نوع آزمون: اصلی.

ب-۴-۳ پیاده‌سازی قاعده کدگذاری معین

الف- هدف آزمون: بررسی اینکه سرویس کدگذاری، قاعده کدگذاری ارجاع شده را پیاده‌سازی کرده باشد.

ب- روش آزمون: بازرسی پیاده‌سازی سرویس کدگذاری.

پ- مرجع: بند ۱۰.

ت- نوع آزمون: قابلیت.

ب-۵ پشتیبانی از هر الگوی کاربردی

الف- هدف آزمون: بررسی اینکه سرویس کدگذاری از هر الگوی کاربردی، همانطور که توسط رخ‌نمون UML مشخص شده و در قاعده کدگذاری به‌کار گرفته‌شده، پشتیبانی کرده باشد.

ب- روش آزمون: بازرسی واسط سرویس کدگذاری جهت بررسی اینکه آیا از مدل نمونه عمومی پشتیبانی می‌کند.

پ- مرجع: بند ۸.

ت- نوع آزمون: قابلیت.

ب-۶ کدگذاری داده

الف- هدف آزمون: بررسی اینکه سرویس کدگذاری عملکرد نوشتن داده را فراهم کرده باشد.

ب- روش آزمون: بازرسی واسط سرویس کدگذاری جهت بررسی اینکه عملکرد نوشتن داده را فراهم کرده باشد.

پ- مرجع: بند ۱۰.

ت- نوع آزمون: قابلیت.

ب-۷ کدگشایی داده

الف- هدف آزمون: بررسی اینکه سرویس کدگذاری عملکرد خواندن داده را فراهم کرده باشد.

ب- روش آزمون: بازرسی واسط سرویس کدگذاری جهت بررسی اینکه عملکرد خواندن داده را فراهم کرده باشد.

پ- مرجع: بند ۱۰.

ت- نوع آزمون: قابلیت.

ب-۸ تولید الگو

الف- هدف آزمون: بررسی اینکه سرویس کدگذاری عملکرد تولید الگوی فرمت را فراهم کرده باشد.

ب- روش آزمون: بازرسی واسط سرویس کدگذاری جهت بررسی اینکه عملکرد تولید الگوی فرمت را فراهم کرده باشد.

پ- مرجع: بند ۱۰.

ت- نوع آزمون: قابلیت.

پیوست پ

(آگاهی دهنده)

قاعده کدگذاری XML مبنا که مورد استفاده جوامع می‌باشند

پ-۱ مقدمه

این پیوست مثالی از قاعده کدگذاری برمبنای XML را برای تبادل بی‌طرفانه داده در برخی از جوامع معرفی می‌کند. قاعده کدگذاری با زبان مدل‌سازی واحد (UML) سازگار بوده و یک قاعده کدگذاری برمبنای XML را تعریف می‌کند.

یادآوری- پیوست الف این استاندارد، یک قاعده کدگذاری برمبنای XML را مشخص کرده است. این قاعده در بعضی از جامعه‌ها استفاده شده است. این پیوست **پ** در واقع مروری بر این قاعده برای جوامع در مطابقت با این استاندارد می‌باشد.

این پیوست از الزامات بند ۹ پیروی می‌کند و موارد زیر را مشخص می‌کند:

الف- الزامات کدگذاری عمومی در زیربند پ-۲؛

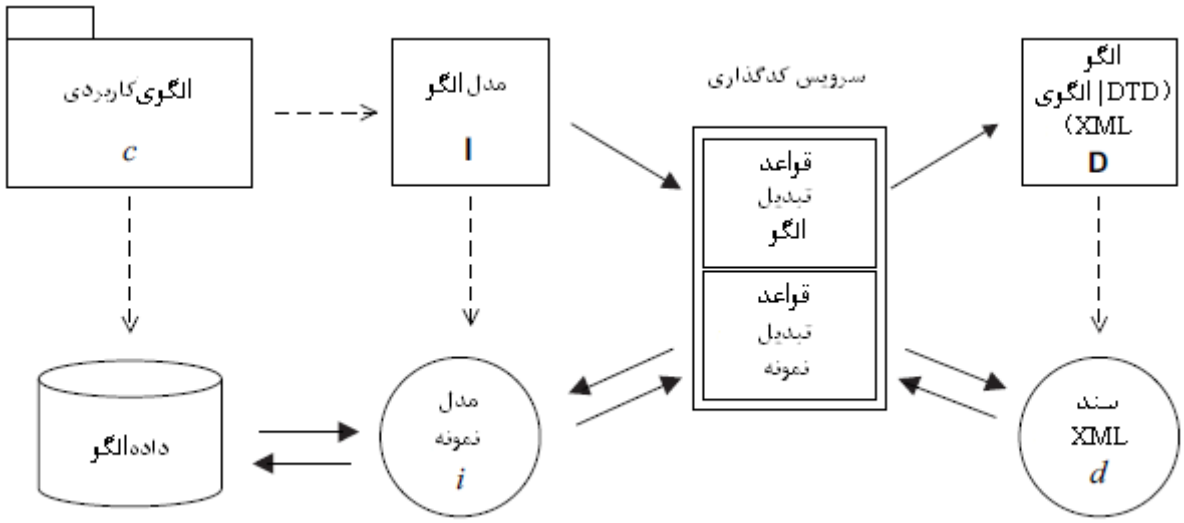
ب- ساختار داده ورودی در زیربند پ-۳؛

پ- ساختار داده خروجی در زیربند پ-۴؛

ت- قواعد تبدیل در زیربندهای پ-۵ و پ-۶.

مثال‌ها در زیربندهای پ-۵-۸ و پ-۶-۴ آورده شده‌اند.

قواعد تبدیل بر مبنای این ایده بوده که تعاریف کلاس در الگوی کاربردی، به نوع اعلان‌ها در الگوی XML نگاشت شده، و اشیا درون مدل نمونه نیز به ساختارهای اجزا متناظر در سند XML نگاشت می‌شوند. شکل پ-۱ دو نوع از قاعده تبدیل را نشان می‌دهد.



شکل پ-۱- قواعد تبدیل برمبنای XML

پ-۲ الزامات عمومی کدگذاری

پ-۲-۱ الگوی کاربردی

پ-۲-۱-۱ مقدمه

الگوی کاربردی باید به زبان الگوی UML مطابق با قواعدی کـه در اسـتانداردهای ISO/TS 19103 و ISO 19109 مشخص شده، بیان شود. الگوی کاربردی شامل مفاهیم تعریف شده کاربردی، که به‌عنوان کلاسـها و وابستگی‌ها بیان شده، می‌باشد. برخی از ایـن کلاسـها ممکـن اسـت از الگـوهـای اسـتاندارد شـده در سـایر استانداردها وارد شده باشند.

پ-۲-۱-۲ کلاس

مفهوم اساسی مدل‌سازی در UML همان کلاس می‌باشد. این کلاس با یک نام شناسایی شده و ممکن است دارای یک کلیشه نیز باشد. کلاس مذکور ممکن است دارای صفت، عملیات و محدودیت باشـد و وابسـتگی-هایی نیز داشته باشد. یک کلاس نوعی را تعریف کرده که می‌تواند به عنوان یک بلوک ساخت، سایر کلاس‌ها را تعریف کند. انواع اشاره شده، همانند بلوک‌های ساخت بوده که فارغ از تمام شکل‌های داده، به عنوان مثال، اعداد، مختصات، رشته‌های متنی، تاریخ‌ها و اشیا، می‌توانند ترکیب و کدگذاری شوند. یک نوع، دامنه مقـادیر قانونی و عملیات مجاز را بر مقادیر آن دامنه تعریف می‌کند.

دو دسته‌ی کلاس اصلی از نوع‌ها، انواع داده ساده و انواع داده مرکب می‌باشند. یک نوع در صورتی یک نوع داده ساده است که یک کدگذاری استاندارد[1] برای آن نوع تعریف شده باشد. این کدگذاری استاندارد ممکن است چگونگی ارائه مقادیر نوع را به‌عنوان بیت‌ها در یک مکان حافظه یا به‌عنوان نویسه‌ها در یک کدگذاری متنی معین کند. اعداد صحیح، اعداد اعشاری و رشته‌ها مثال‌هایی از انواع ساده می‌باشند. یک نوع، هنگامی از نوع داده مرکب است که هیچ کدگذاری استانداردی برای آن نوع تعریف نشده باشد. مثال‌های این نوع عبارت‌اند از: انواع شی، انواع ساختاریافته، رکوردها[2] و مجموعه‌ها[3]. یک نوع مرکب شامل یک مجموعه ساختاریافته از صفت پیچیده و اصلی بوده که می‌تواند با استفاده از ترکیبی از انواع اصلی و ابتکارهای ساختاری خاص کدگذاری شود. دسته‌ی سوم انواع داده خارجی بوده که خارج از مجموعه استانداردهای بین المللی، مجموعه استانداردهای « ISO 19100»، تعریف شده‌اند.

الف- انواع داده ساده- انواع اساسی برای نشان دادن مقادیر:

۱- انواع داده اصلی: رشته نویسه[4]، اعداد صحیح، دودویی، بولی، تاریخ، زمان و غیره.
۲- انواع داده برشمارش و فهرست‌های کد: فهرستی از مقادیر مجاز، که در آن هر مقدار، یک کلمه یا یک کد، با قواعد معنایی مرتبط می‌باشد.

ب- انواع داده مرکب- انواعی برای نشان دادن مجموعه‌های مرکب بیشتری از مقادیر:

۱- انواع داده مجموعه: انواع قالب برای ارائه چندین رخداد از سایر انواع- مجموعه[5]، بسته[6]، دنباله[7]، لغت‌نامه[8] و غیره.
۲- انواع داده ساختار یافته: انواعی که گروه‌های صفت را تعریف می‌کنند.
۳- انواع شی: انواعی که نمونه‌هایش اشیا می‌باشند؛ اغلب در الگوهای کاربردی یا الگوهای استاندارد شده تعریف شده‌اند- GM_Point، ساخت[9] و غیره.
۴- واسطها: انواعی که نمونه‌هایش مولفه سرویس می‌باشند.

پ- انواع داده خارجی- انواع اصلی یا مرکب همراه با یک کدگذاری خوش‌تعریف که در مجموعه استانداردهای ISO 19100 تعریف نشده‌اند؛ مثال‌هایی از این انواع عبارت‌اند از: فرمت‌های تصویری، مانند NASA MODIS، TIFF و غیره. سازوکارهای خاص ارجاع به این داده‌ی خارجی باید برای امکان ارجاعات به این داده تعیین شده، که بطور معمول این موارد در فایل‌های مجزا ذخیره می‌شوند.

1- Canonical
2-Records
3- Collection
4- CharacterString
5- Set
6- Bag
7- Sequence
8- Dictionary
9- Building

استاندارد ISO/TS 19103 مفاهیم به‌کارگرفته شده مدل‌سازی، شامل تعدادی از انواع داده‌ی ساده و مرکب را با یکدیگر و با قواعد معنایی‌شان تعریف می‌کند. توسعه‌دهندگان برنامه کاربردی برای مشخص کردن انواع داده تعریف‌شده توسط کاربر و انواع شی، با استفاده از کلیشه در سازوکار توسعهUML ، آزادی عمل دارند. سایر استانداردهای مجموعه استانداردهای ISO 19100 انواع بیشتری از داده‌ی اختصاصی را تعریف می‌کنند. به طور معمول استفاده از انواع داده خارجی برای توسعه‌دهندگان الگوی کاربردی اجتناب‌ناپذیر است. انواع داده خارجی داده‌ی اصلی یا ساختاریافته بوده که خارج از این مجموعه استانداردها تعریف می‌شوند. ضروری است که آن‌ها دارای یک کدگذاری خوش‌تعریف باشند؛ نمونه‌های آن عبارت‌اند از: فرمت‌های تصویر، نظیرتصاویر NASA MODIS و TIFF. اشیای داده‌ی خارجی می‌توانند در فایلهای مجزا ذخیره شوند و باید سازوکارهای ارجاع خاص به آن‌ها که امکان ارجاع‌دهی به انواع داده خارجی را فراهم کرده، مشخص شوند.

کلیشه‌هایی که در کلاس‌ها مجاز بوده در جدول پ-۱ توضیح داده می‌شوند.

جدول پ-۱- کلیشه‌ها در کلاس‌ها

کلیشه	توصیف
<<Basic Type>>	یک نوع داده اصلی را تعریف کرده که یک کدگذاری استاندارد را تعریف کرده است.
<<Data Type>>	یک نوع داده ساختار یافته را تعریف می‌کند. نمونه‌های آن به‌عنوان اشیا در نظر گرفته نمی‌شوند و از این رو، هیچ هویتی نباید داشته باشند.
<<UNION>>	یک نوع داده اجتماع را تعریف می‌کند.
<<Enumeration>>	یک نوع داده برشمارش را تعریف می‌کند.
<<CodeList>>	یک نوع داده برشمارش توسعه‌پذیر، شامل کدها و مقادیر، را تعریف می‌کند.
<<Interface>>	یک واسط سرویس را تعریف کرده و نباید کدگذاری شود.
<<Type>>	یک نوع شی را تعریف می‌کند. نمونه‌ها باید دارای هویت باشند.
<<Feature Type>>	یک نوع عارضه را تعریف می‌کند. نمونه‌ها باید دارای هویت باشند.
NONE	یک نوع شی را تعریف می‌کند. نمونه‌ها باید دارای هویت باشند.

یک شی به‌عنوان واحد اصلی تبادل در نظر گرفته می‌شود. فقط جنبه‌هایی ضروری می‌باشند که برای بدست آوردن حالت شی در راستای اهداف تبادل داده در نظر گرفته شوند. صفت‌ها و وابستگی‌ها باید کدگذاری شوند. عملیات‌ها و محدودیت‌ها نباید در مراحل جلوتر در نظر گرفته شوند. یک قاعده کدگذاری باید نحوه نمایش نمونه‌های کلاسها، که شامل چگونگی نمایش و ساختاردهی صفت‌ها و وابستگی‌ها بوده، را تعیین کند.

پ-۲-۱-۳ صفت‌ها

یک صفت با یک نام تعریف می‌شود. این صفت می‌تواند چندین عبارت مختلف داشته و همیشه باید دارای یک نوع باشد. عبارت‌های چندگانه صفت باید تعداد رخدادهای مقدارهای مجاز یک صفت خاص را مشخص کند.

ترکیب‌های چندگانگی و استفاده از انواع مجموعه داده، تداخل مقادیر را امکان‌پذیر می‌کند.

باید سازوکاری برای مدیریت مقادیر پوچ وجود داشته باشد. این نوع، باید چگونگی نمایش یک مقدار پوچ را تعریف کند. به استاندارد ISO/TS 19103 برای تعریف انواع داده اصلی و مجموعه داده مراجعه شود.

پ-۲-۱-۴ وابستگی[1]ها

وابستگی‌ها روابط بین کلاس‌هایی را که بین نمونه‌هایشان اتصالاتی باشد، تعریف می‌کنند. معرفی یک وابستگی به وسیله نمونه، لینک[2] نامیده می‌شود. یک لینک شامل فهرست مرتبی از ارجاع‌ها به اشیا می‌باشد. باشد. UML سه نوع مختلف از وابستگی‌ها را معرفی می‌کند که وابستگی، تجمیع و ترکیب نامیده میشود.

- وابستگی، روابط کلی بین کلاسها را تعریف می‌کند؛
- تجمیع، روابط ضعیف جز - کل در بین کلاسها را تعریف می‌کند؛
- ترکیب، روابط قوی جز - کل در بین کلاسها را تعریف می‌کند.

این سه نوع وابستگی قواعد معنایی متفاوتی دارند. استاندارد ISO/TS 19103 جزئیات بیشتری را در این خصوص ارائه می‌دهد.

پایان یک وابستگی با اسم نقش و کلاس هدف آن مشخص می‌شود، و در ادامه با یک بیان چندگانه شرح داده می‌شود. اسم نقش باید برای نمایش یک لینک استفاده شود. اگر اسم نقش موجود نباشد، لینکی هم وجود نخواهد داشت.

پ-۲-۲ ساختار تبادل و فراداده تبادل

ساختار تبادل باید به سه جز تقسیم شود. اولین جز فراداده تبادل است که باید براساس زیربند ۹-۲-۴ توصیف شود. جز دوم در خصوص مجموعه داده است که باید حاوی جزهای XML بوده که به اشیای مستقل مربوط است. جز سوم اختیاری بوده و باید حاوی بخش به‌روزرسانی نسخه‌های اولیه‌ی بروز شده که شرح آن در زیربند پ-۲-۵ آورده شده، باشد.

1- Association
2- Link

پ-۲-۳ مجموعه نویسه و زبان

ساختار اساسی تبادل باید از زبان و مجموعه نویسه پشتیبانی کند:

- زبان (برای برچسب‌گذاری زبان به زیربند پ-۵-۶ مراجعه شود)؛
- مجموعه نویسه (به استاندارد ISO/IEC 10646 مراجعه شود).

پ-۲-۴: مجموعه داده و شناسایی هدف

دو راه شناسایی شی که در زیربند ۹-۲-۵-۲ توضیح داده‌شده، باید به صورت زیر باشد:

- شناسه‌های منحصربه‌فرد مجموعه داده مطابق با سازوکار شناسه‌ی XML؛
- شناسه‌های منحصربه‌فرد بین‌المللی[1] مطابق با تعیین دامنه‌ی کاربردی.

یادآوری- اصطلاح «universal unique identifier» معنای مشابهی با «domain unique identifier» دارد. این پیوست از «UUID» به‌صورت واژه‌ی مخفف با سازگاری برای به‌کارگیری در تمام جوامع استفاده می‌کند. کاربرد UUID در این استاندارد، همانند منبع‌های [۸] و [۱۱] کتاب‌نامه به یک پایه ریاضیاتی خاصی محدود نمی‌باشد. «universal unique identifier» در واقع با دامنه‌های کاربردی ارزیابی نمی‌شود، برخلاف «domain unique identifier» که بادامنه‌های کاربردی ارزیابی می‌شود.

مثال: دامنه کاربردی می‌تواند خواهان به‌کارگیری یک شناسه‌ی دو مولفه‌ای باشد. اولین مؤلفه نام دامنه و مؤلفه بعدی یک نمونه عدد صحیح میباشد. این مؤلفه‌ها می‌توانند توسط یک دونقطه «:» از هم جدا شوند. اعداد نمونه می‌توانند در مبنای ۶ کدگذاری شوند. هیچ محدودیتی در اندازه‌ی اعداد نمونه وجود ندارد. مثالهایی از دو UUID که در یک دامنه «example» نامیده می‌شوند: «example:F23C30» و «example:FFFFFF12345A».

پ-۲-۵: سازوکار به‌روزرسانی

سازوکار به‌روزرسانی به داده‌هایی که از قبل تبادل شده‌اند اجازه می‌دهد تا بدون نیاز به انتشار مجدد یک مجموعه داده جدید به روزرسانی شوند. راهبردها و روند به‌روزرسانی باید توسط یک برنامه کاربردی به‌خصوص تعریف شود. به طورمعمول سه اصل اساسی در امر به‌روزرسانی تعریف می‌شود: اضافه‌کردن، تغییر و حذف کردن. این اصول، در سطح شی کاربرد دارد، اما ممکن است در سطوح صفت‌ها و وابستگی‌های شی نیز عمل کنند. هر شی که قبلا توسط یک UUID انتقال یافته، می‌تواند اصلاح یا حذف گردد. یک مجموعه داده به‌روزسانی‌شده شامل دنباله‌ای ترتیبی از اصول بروزرسانی می‌باشد. اصول اصلی به روزرسانی به شرح زیر هستند.

1- Universal Unique Identifier

۱- *اضافه کردن*: یک شی جدید به مجموعه داده منبع اضافه شده و باید درون مجموعه داده هدف نیز اضافه گردد. اصل اضافه کردن باید شامل اطلاعاتی درخصوص شی جدید اضافه شده و نیز اطلاعاتی درباره محلی که در مجموعه داده هدف قرار می‌گیرد، باشد.

۲- *اصلاح یا تغییر*: اگر یک شی موجود در مجموعه داده منبع تغییر بیابد، باید در مجموعه داده مقصد نیز اصلاح شود. اصل تغییر باید حاوی اطلاعاتی بوده که بتواند شی هدف و تغییرات واقعی را شناسایی کند. مثال‌هایی از اطلاعات تغییر می‌تواند در محدوده‌ای از بروزرسانی یک شی کامل تا صرفا یک صفت باشد.

۳- *حذف کردن*: وقتی یک شی موجود در مجموعه داده منبع حذف شود، باید در مجموعه داده مقصد نیز حذف شود. اصل حذف باید حاوی اطلاعاتی بوده که بتوان شی هدفی را که باید حذف شود را شناسایی کند.

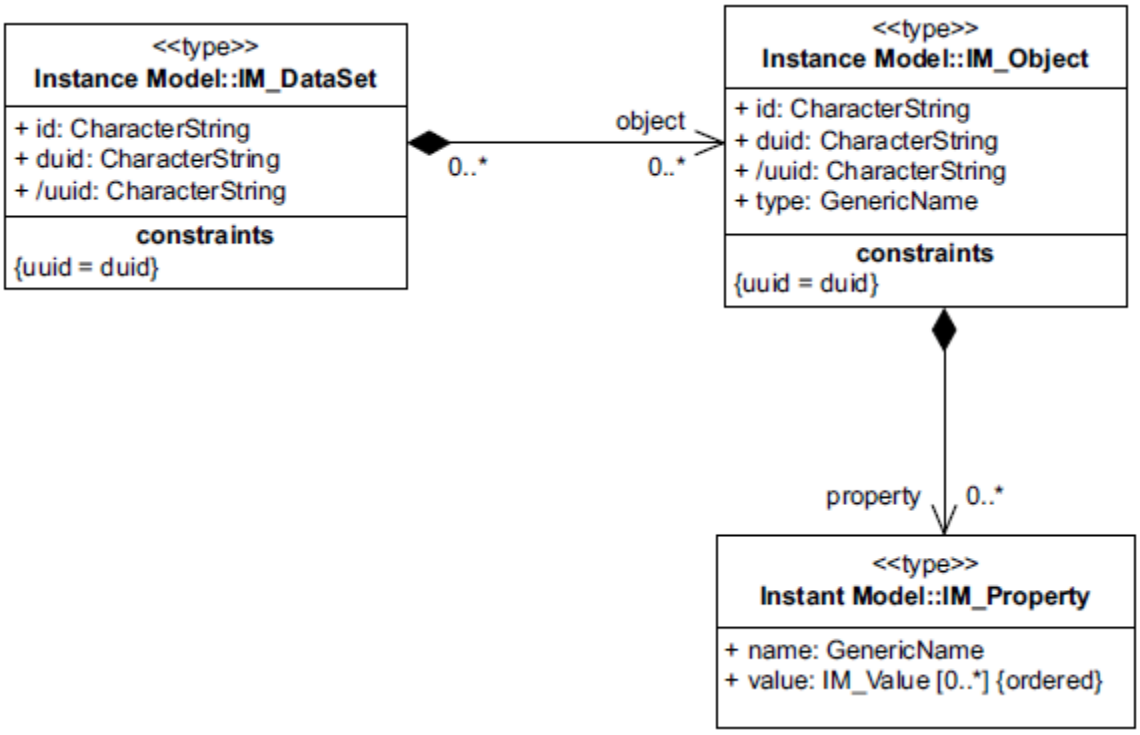
کاربران ممکن است فهرست اصول را گسترش دهند.

پ-۳ ساختار داده ورودی

پ۳-۱ مدل نمونه

مدل نمونه قابلیت نمایش داده‌ی توضیح داده شده توسط الگوهای کاربردی بیان شده در UML، را دارد. همانند ساختار داده ورودی، قاعده کدگذاری نیز از یک مدل نمونه براساس مدل نمونه عمومی که در بند ۸ تعریف شده، استفاده می‌کند. مدل نمونه در شکل‌های پ-۲ و پ-۳ نشان داده شده است.

شکل پ-۲ کلاس‌های IM_Dataset، IM_Object و IM_Property را که در اصل مشابه موارد تعریف شده در زیربند ۸-۱ و شکل ۴ هستند را نشان می‌دهد.



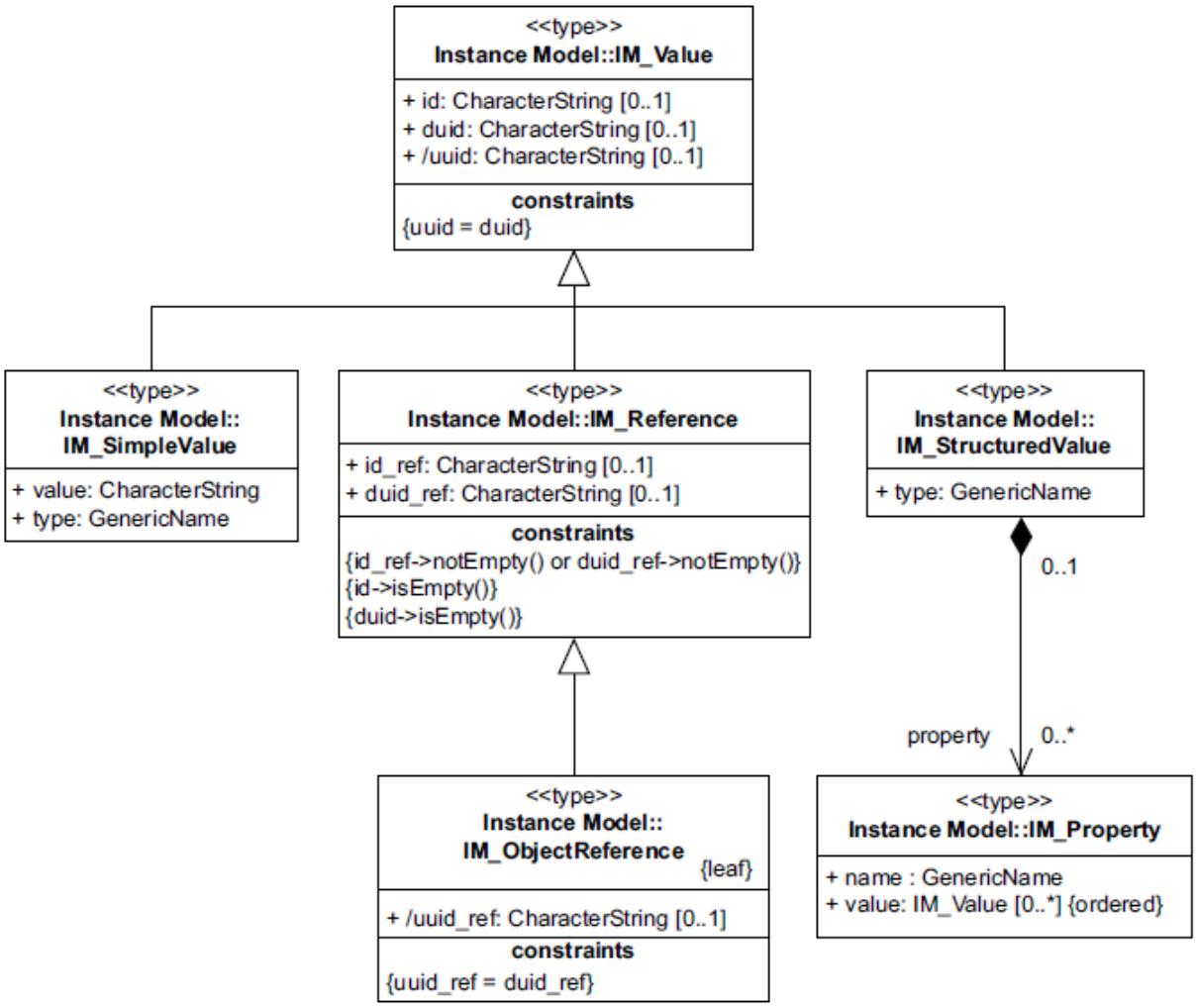
شکل پ-۲- مدل نمونه- مجموعه داده، شی و ویژگی

مدل نمونه شامل مجموعه داده‌ی نشان داده شده توسط IM_Dataset می‌باشد. یک مجموعه داده شامل دنباله‌ای از اشیا است که توسط IM_Object نشان داده می‌شود. یک شی شامل دنباله‌ای از ویژگی‌ها است که توسط IM_Property نشان داده می‌شود.

تعریف دیگری از صفت uuid در IM_Dataset و IM_Object وجود دارد. صفت حاصل از uuid به عنوان نام جایگزین صفت duid تعریف می‌شود.

یادآوری ۱- تعریف صفت uuid جهت حفظ سازگاری در به‌کارگیری در جوامع مختلف می‌باشد.

انواع مقادیر مدل نمونه در شکل پ-۳ که براساس شکل ۵ می‌باشد، تعریف می‌شوند.



شکل پ-۳- مدل نمونه- انواع مقدار

IM_Value یک ابرکلاس[1] شامل چهار نوع مقدار است که به صورت زیر تعریف می‌شود.

- IM_SimpleValue یک مقدار از نوع اصلی را نشان می‌دهد.

مثال: یک عدد صحیح یا یک رشته نویسه.

- IM_Reference یک پیوند یا ارجاع به شی هدف را نشان می‌دهد. شی هدف ممکن است در همان پایگاه داده یا پایگاه داده‌ی دیگر قرار داشته باشد.

- IM_ObjectReference یک زیرنوع از IM_Reference است. یک شناسه منحصربفرد (id_ref) یک شی در محدوده‌ی همان مجموعه داده را مورد هدف قرار می‌دهد. یک شناسه منحصربفرد بین‌المللی (UUID-REF و DUID-REF) یک شی واقع شده در دامنه کاربردی را هدف قرار می‌دهد.

1 - superclass

- IM_StructuredValue مقدار نوع داده با محتوای مرکب را نشان می‌دهد [یک دنباله از ویژگی‌ها (IM_Property)].

تفاوت بین این مدل نمونه و مدل نمونه عمومی در بند ۸ به صورت زیر بیان شده است.

الف- IM_Value حاوی صفت حاصل از uuid به عنوان نام جایگزین صفت duid می‌باشد.

ب- IM_ObjectReference تعریف شده و دارای یک صفت اضافی حاصل از uuid-ref بوده که نام جایگزینی صفت duid-ref تعیین می‌کند.

کلاس‌های دیگر که در شکل پ-۳ تعریف شده‌اند برای موارد تعریف شده در زیربند ۸-۱ و شکل ۵ یکسان هستند.

یادآوری ۲- تعریف کلاس IM_ObjectReference و صفت uuid در IM_Value و IM_ObjectReference جهت حفظ سازگاری در به‌کارگیری در جوامع مختلف می‌باشد.

پ-۳-۲ ارتباط بین UML و مدل نمونه

جدول‌های پ-۲ و پ-۳ خلاصه‌ای از ارتباط بین UML و مدل نمونه را ارائه می‌دهند. بنابراین کلاس انتزاعی نباید معرفی گردد.

اشیا براساس کلاس‌هایی که دارای ابرنوع بوده باید حاوی تمام ویژگی‌ها، وابستگی‌ها و ترکیباتی از کلاسهایشان و انواع ابرنوع‌ها باشند. بنابراین همه صفت و وابستگی‌ها باید از ابرنوع‌ها نسخه‌برداری شده و به عنوان جزئی از شی در نظر گرفته شوند. صفت و نام وابستگی‌ها باید روشی برای دسترسی به مقادیر صفت بوده و این مقادیر باید در کلاسها به‌صورت‌منحصربه‌فرد باشند.

عملیات و قیود نباید به مدل نمونه نگاشت شوند.

جدول پ- ۲- خلاصه‌ای از رابطه بین UML و مدل نمونه

مفهوم UML	مدل نمونه
Package	N/A[a]
Class Stereotype <<Interface>> >>BasicType<< <<DataType>> <<Union>> <<Enumeration>> <<CodeList>> <<Type>> or None Abstract class Sub-class	N/A IM_SimpleValue IM_StructuredValue IM_StructuredValue IM_SimpleValue IM_SimpleValue IM_Object N/A صفت‌ها و وابستگی‌ها باید از اَبَرکلاس‌ها نسخه‌برداری شوند
صفت	IM_Property با IM_Value مطابق با نوع داده (یا IM_SimpleValue یا

مفهوم UML	مدل نمونه
	IM_StructuredValue)
وابستگی	IM_Property با IM_Value از IM_ObjectReference
تجمیع	IM_Property باIM_Value مطابق با IM_StructuredValue یا IM_ObjectReference
ترکیب	IM_Property با IM_Value hc از IM_StructuredValue
عملیات	N/A
محدودیت	N/A
a N/A به معنی عدم کاربرد می‌باشد	

جدول پ- ۳- نگاشت صفت با چندگانگی و نوع مجموعه

مدل نمونه	خصیصه
IM_Property با رخدادهای مقدار چندگانه	a1 [0..*] : Integer مشابه است با a1 : Sequence<Integer>[a]
IM_Property با رخدادهای چندگانه‌ی مقدار نوع T	a2 : Sequence<T>
IM_Property با رخدادهای چندگانه‌ی IM_StructuredValue با دو جز از نوع مقدار T1وT2	a3 : Dictionary<T1, T2>
IM_Property با رخدادهای چندگانه‌ی IM_StructuredValue با جزءهای عدد صحیح	a4 [0..*] : Sequence<Integer>
a برای هر نوع اصلی دیگر نیز معتبر است	

مغایرت نام صفت و وابستگی ممکن است در هنگام استفاده از قواعد وراثت مشکلاتی به وجود آورند. یک راه ساده برای جلوگیری از این مشکل اطمینان از این موضوع است که همه صفت‌ها و وابستگی‌ها باید از قبل با نام کلاس مناسب ثابت شوند؛ روش جایگزین، پرهیز از مغایرت نام‌ها است که به عهده کاربر گذاشته می‌شود.

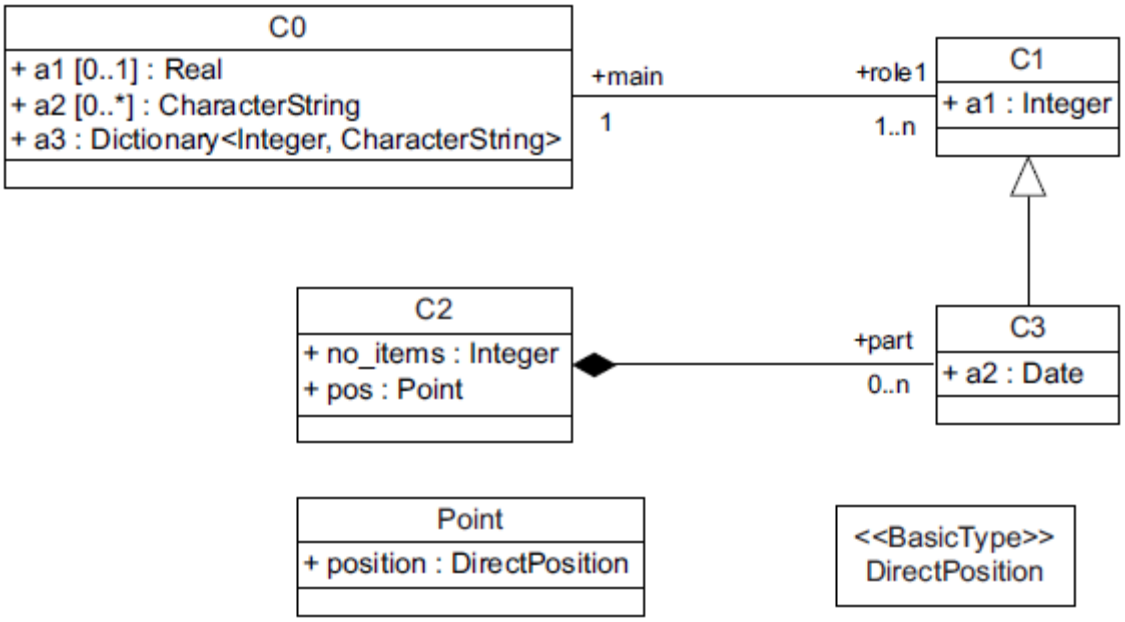
بیان مجدد صفت و وابستگی نیز می‌تواند در استفاده از قواعد وراثت مشکلاتی را فراهم کند. بیان مجدد زمانی اتفاق می‌افتد که یک صفت و یا وابستگی در یک ابرنوع، دوباره در یک زیرنوع با یک نوع جدید بیان می‌شود. بسیاری از زبان‌های برنامه نویسی شی‌گرا قادر به مدیریت مناسب بیان دوباره نبوده و این مشکلات باید به دقت در نظر گرفته شود که در کجا باید از بیان دوباره پرهیز شود.

پ-۳-۳ الگوی کاربردی و مدل نمونه- مثال

پ-۳-۳-۱ الگوی کاربردی

این مثال یک الگوی کاربردی را تعریف می‌کند، اطلاعاتی را ارائه داده و نگاشتی از داده به مدل نمونه نشان می‌دهد.

شکل پ-۴ مثالی از الگوی کاربردی را تعریف می‌کند که چهار کلاس و ارتباطات بین آن‌ها را تعریف می‌کند. در این‌جا کلاس C0 دارای سه صفت است. صفت«a1» دارای چندگانگی صفر و یک است که به معنی اختیاری بودن آن بوده و یک نوع داده اصلی «Real» می‌باشد. صفت «a2» دارای چندگانگی صفر یا چند صفر می‌باشد. صفت «a3» یک نوع داده ساختاری است. وابستگی بین «C0» و «C1» دارای دو نام نقش است. «role1» به C0 تعلق داشته و وابستگی به C1 را در ارتباط با C0 نام‌گذاری می‌کند. کلاس C1 وابسته به C0 است که این کلاس «main» نامیده می‌شود. توجه داشته باشید که کلاس C2 یک صفت به نام «pos» با یک نوع داده کلاس point دارد.

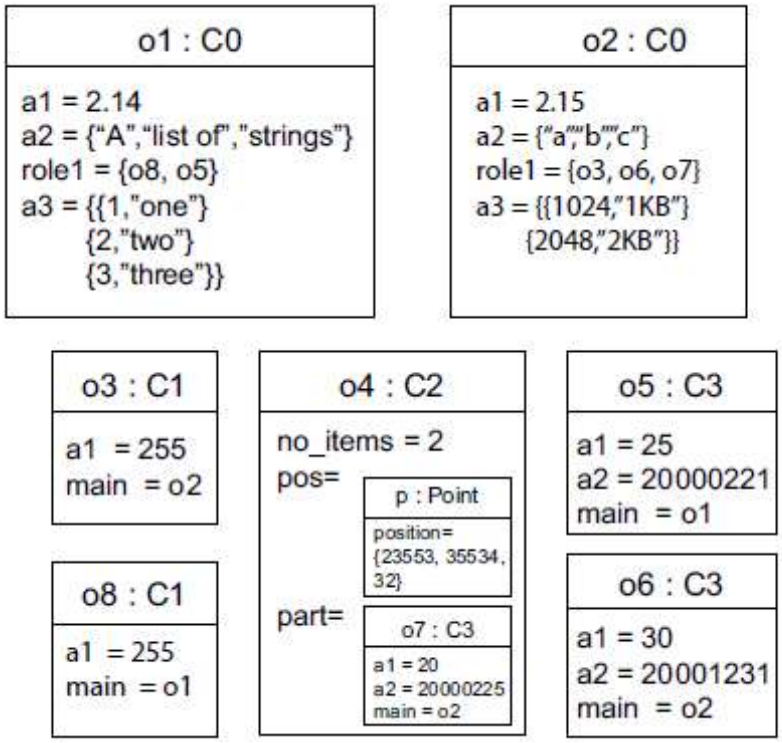


شکل پ-۴- الگوی کاربردی- مثال

پ-۳-۳-۲ داده- مثال

داده‌ی نمونه در شکل پ-۵ در یک نمودار[1] شی نشان داده می‌شود.

1- Diagram



شکل پ-۵- داده- مثال

پ-۴ ساختار داده خروجی

پ-۴-۱ سند XML

این قاعده کدگذاری براساس توصیه XML ۱٫۰[1] می‌باشد. XML یک قالب متنی است و لازم است که مقادیر تمام انواع داده با نویسه کدگذاری شوند. داده باید با استفاد از جزهای XML و براساس قواعد توصیه شده در XML کدگذاری شوند. واحدهای اصلی کدگذاری در XML جزهای XML هستند. یک جز ممکن است دارای صفت‌ها و محتوا باشد. این مسأله یک ساختار سلسله مراتبی و مرکب با امکانات ارتباطی XML در ساختار یک شبکه ایجاد می‌کند. ساختار تبادل باید شامل دنباله‌ای از جزهای مربوط به اشیا در مدل نمونه باشد.

الگوی ساختار داده خروجی که بر ساختار فرمت تبادل حاکم است، باید یک الگوی XML معتبر باشد.

1 -XML Recommendation 1.0

پ-۴-۲ الگوی XML

سند الگوی XML (XSD[1]) تعدادی از انواع مرکب، انواع ساده و جز اعلان‌ها را تعریف می‌کند که ساختار مجاز و نمونه‌های داده از یک سند XML را تبیین می‌کنند. قواعد تبدیل الگوی XML در زیربند پ-۵ تعریف شده است.

پ-۵ قواعد تبدیل الگو

پ-۵-۱ الگوی XML

قواعد تبدیل الگو چگونگی تولید یک سند الگوی XML (XSD) را مطابق با الگوی کاربردی بیان‌شده در UML، تعریف می‌کند. هدف اصلی XSD اطمینان از اعتبار اسناد XML تولیدشده با به‌کارگیری قواعد تبدیل داده می‌باشد.

XSD باید حاوی تعاریف نوع و صفت و اعلان جز باشد که به کلاس‌های تعریف شده در الگوی کاربردی مرتبط باشند. جزها باید در یک ساختار تبادل سازمان‌دهی شوند. XSD باید با الگوی XML در *قسمت ۱*: ساختارها و *قسمت ۲*: انواع داده‌ها مطابقت داشته باشد.

XSD به طور فیزیکی ممکن است در یک سند الگوی منفرد نشان داده شود و یا به چندین سند الگوی (فرعی) جداگانه تقسیم شود. XSD به طور منطقی باید به عنوان یک الگوی منفرد با استفاده از دریافت یا داشتن سازوکارهای الگوی XML، معرفی شود. محدودیتی در استفاده از فضاهای‌نام در یک XSD وجود ندارد.

تعدادی از قواعد کلی در زیربندهای پ-۵-۲ تا پ-۵-۸ تعریف شده است. یک کلاس در حالت کلی می‌تواند براساس زیربند پ-۵-۲ به یک نوع تبدیل شود، کلاس ممکن است به یک اعلان جز مطابق با زیربند پ-۵-۴ تبدیل شود و ممکن است عضوی از ساختار تبدیل باشد.

استثنائات در قواعد کلی تا زمانی که مستندسازی شوند مجاز هستند.

یادآوری- در ادامه فضای اسمی «xs:» به منظور ارجاع به فضای اسمی الگوی XML، «http://www.w3.org/2001/XMLSchema» استفاده می‌شود.

1- XML Schema Document

پ-۵-۲: انواع

پ-۵-۲-۱ <<BasicType>>

پ-۵-۲-۱-۱ قاعده کلی

یک کلیشه کلاس <<BasicType>> باید به یک اعلان ساده در الگوی XML تبدیل شود. هر نوع داده‌ای که در الگوی XML تعریف‌شده باشد، می‌تواند به عنوان بلوک ساخت برای تعریف انواع اصلی که توسط کاربر تعریف می‌شوند، به‌کار گرفته شود. کدگذاری انواع اصلی باید از نمایش استاندارد تعریف شده در الگوی XML جز ۲: انواع داده‌ها پیروی کند.

قواعد مدل‌سازی برای انواع اصلی در استاندارد ISO/TS 19103 تعریف شده است. انواع اصلی تعریف‌شده در استاندارد ISO/TS 19103 با توجه به زیربندهای پ-۵-۲-۱-۲ تا پ-۵-۲-۱-۱۵ تبدیل می‌شوند.

کابران مجاز هستند تا انواع اصلی را با استفاده از سازوکارهای محدودیت تعریف شده در الگوی XML بیشتر محدود کنند.

یادآوری- انواع مختلفی وجود دارد که به صورت واضح نه در استاندارد ISO/TS 19103 و نه در کلیشه مورد استفاده <<BasicType>> تعریف نشده‌اند. بنابراین، اعلان‌هایی که در ادامه آمده

از تعاریف نوع داده در الگوی XML جز ۲: انواع داده‌ها پیروی می‌کنند.

پ-۵-۲-۱-۲ عدد

عدد[1] در استاندارد ISO/TS 19103 به عنوان یک نوع انتزاعی و ابرنوع از انواع اعداد صحیح، حقیقی و اعشاری تعریف شده است. با این حال، این موارد بیش‌تر در الگوهای استاندارد شده استفاده می‌شود. الگوی XML یک نوع عدد انتزاعی را تعریف نمی‌کند اما اعداد اعشاری را به عنوان یک ابرنوع اعداد صحیح تعریف می‌کند.

یادآوری- نوع داده عددی به عنوان یک نوع داده ساده بر مبنای نوع داده اعشاری بیان می‌شود.

```
<xs:simpleType name="Number">
    <xs:restriction base="xs:decimal"/>
</xs:simpleType>
```

پ-۵-۲-۱-۳ عدد صحیح

یک عدد صحیح[2] باید بر مبنای نوع داده عدد صحیح الگوی XML باشد. دامنه مقدار این نوع، ممکن است محدود باشد.

```
<xs:simpleType name="Integer">
    <xs:restriction base="xs:integer"/>
```

1- Number
2- Integer

```
</xs:simpleType>
```

پ-۵-۲-۱-۴ اعداد حقیقی و اعشار

اعداد حقیقی[1] و اعشار[2] هر دو بر مبنای نوع اعشاری الگوی XML می‌باشند.

```
<xs:simpleType name="Decimal">
      <xs:restriction base="xs:decimal"/>
</xs:simpleType>

<xs:simpleType name="Real">
  <xs:restriction base="xs:decimal"/>
</xs:simpleType>
```

یادآوری- استاندارد ISO/TS 19103 یک تفاوت مفهومی بین Decimal وReal را تعریف نمی‌کند.

پ-۵-۲-۱-۵ بردار

بردار[3] به عنوان دنباله‌ای از اعداد تعریف می‌شود. این ساختار فهرست، فهرستی از مقادیر اعشاری را تعریف می‌کند.

```
<xs:simpleType name="Vector">
      <xs:list itemType="xs:decimal"/>
</xs:simpleType>
```

پ-۵-۲-۱-۶ نویسه

نویسه[4] به عنوان یک رشته الگوی XML بیان می‌شود که تنها حاوی یک نویسه می‌باشد.

```
<xs:simpleType name="Character">
      <xs:restriction base="xs:string">
            <xs:length value="1" fixed="true"/>
      </xs:restriction>
</xs:simpleType>
```

پ-۵-۲-۱-۷ رشته نویسه

رشته نویسه[5] بر مبنای رشته الگوی XML است که می‌تواند هر رشته استاندارد ISO/IEC 10646 را نشان دهد.

```
<xs:simpleType name="CharacterString">
      <xs:restriction base="xs:string"/>
</xs:simpleType>
```

1- Decimal
2- Real
3- Vector
4- Character
5- CharacterString

یادآوری- برای شناسایی زبان، به زیربند پ-۵-۶ مراجعه شود.

پ-۵-۲-۱-۸ تاریخ

تاریخ[1] بر مبنای تاریخ الگوی XML است که دارای کدگذاری استاندارد مطابق با استاندارد ISO 8601 است.

```
<xs:simpleType name="Date">
    <xs:restriction base="xs:date"/>
</xs:simpleType>
```

پ-۵-۲-۱-۹ زمان

زمان[2] بر مبنای زمان الگوی XML است که دارای کدگذاری استاندارد مطابق با استاندارد ISO 8601 است.

```
<xs:simpleType name="Time">
    <xs:restriction base="xs:time"/>
</xs:simpleType>
```

پ-۵-۲-۱-۱۰ زمان‌تاریخ

زمان‌تاریخ[3] بر مبنای زمان‌تاریخ الگوی XML است که دارای کدگذاری استاندارد مطابق استاندارد ISO 8601 است.

```
<xs:simpleType name="DateTime">
    <xs:restriction base="xs:dateTime"/>
</xs:simpleType>
```

پ-۵-۲-۱-۱۱ بولی

بولی[4] بر مبنای بولی الگوی XML[5] است. مقادیر «0» یا «false» مقدار نادرستی منطقی[6] را نشان می‌دهد و و «1» و «true»، درستی منطقی[7] را نشان می‌دهند.

```
<xs:simpleType name="Boolean">
    <xs:restriction base="xs:boolean"/>
</xs:simpleType>
```

پ-۵-۲-۱-۱۲ منطقی

منطقی[8] سه مقدار را تعریف می‌کند: درست، شاید و نادرست. این مورد به عنوان اتحادی بین الگوی بولی XML و دو مقدار شمارشی «0.5» و «maybe» نشان داده شده، که نشانگر مقدار شاید می‌باشد.

1- Data
2- Time
3- DateTime
4- Boolean
5- XML Schema boolean
6- Logical false
7- Logical true
8- Logical

```
<xs:simpleType name="Logical">
	<xs:union>
		<xs:simpleType>
			<xs:restriction base="xs:boolean"/>
		</xs:simpleType>
		<xs:simpleType>
			<xs:restriction base="xs:NMTOKEN">
				<xs:enumeration value="maybe"/>
				<xs:enumeration value="0.5"/>
			</xs:restriction>
		</xs:simpleType>
	</xs:union>
</xs:simpleType>
```

پ-۵-۲-۱-۱۳ احتمال

احتمال[1] یک عدد اعشاری بین ۰ و ۱ است.

```
<xs:simpleType name="Probability">
	<xs:restriction base="xs:decimal">
		<xs:minInclusive value="0.0"/>
		<xs:maxInclusive value="1.0"/>
	</xs:restriction>
</xs:simpleType>
```

پ-۵-۲-۱-۱۴ دودویی

الگوی XML دو نوع داده دودوییbinary با نام‌های base64Binary و hexBinary را تعریف می‌کند. این دو نوع باید به عنوان انواع داده دودویی مورد استفاده در UML تعریف شوند: base64Binary و hexBinary. یک نوع انتخاب هنگامی Binary نامیده می‌شود که دارای یکی از انواع base64Binary و hexBinary باشد.

```
<xs:simpleType name="BinaryBase64">
		<xs:restriction base="xs:base64Binary"/>
	</xs:simpleType>

	xs:simpleType name="BinaryHex">
		<xs:restriction base="xs:hexBinary"/>
	</xs:simpleType>

	<xs:complexType name="Binary">
		<xs:choice>
			<xs:element name="BinaryBase64" type="BinaryBase64"/>
			<xs:element name="BinaryHex" type="BinaryHex"/>
```

1- Probability

```
        </xs:choice>
    </xs:complexType>
```

یادآوری- نوع داده binary در استاندارد ISO/TS 19103 تعریف نمی‌شود.

پ-۵-۲-۱-۱۵ عدد صحیح نامحدود

عدد صحیح نامحدود UnlimitedInteger یک نوع اصلی بوده که دارای دامنه‌ی از ۰ تا بی‌نهایت می‌باشد. نماد «*» برای نمایش مقدار بینهایت استفاده می‌شود.

```
<xs:simpleType name="UnlimitedInteger">
    <xs:union>
        <xs:simpleType>
            <xs:restriction base="xs:nonNegativeInteger"/>
        </xs:simpleType>
        <xs:simpleType>
            <xs:restriction base="xs:string">
                <xs:enumeration value="*"/>
            </xs:restriction>
        </xs:simpleType>
    </xs:union>
</xs:simpleType>
```

پ-۵-۲-۲ <<DataType>>

پ-۵-۲-۲-۱ قاعده کلی

یک کلیشه کلاس <<DataType>> باید در الگوی XML به complexType تبدیل شود. صفت و وابستگی-های ممکن باید در صفت XML یا جزهای XML محلی در دنباله‌ای مطابق با زیربند پ-۵-۳ تعریف شوند. ترتیب جزهای ویژگی در تعریف نوع مرکب قید شده است.

انواع داده‌ی تعریف شده در استاندارد ISO/TS 19103 به روش زیربندهای پ-۵-۲-۲-۲ و پ-۵-۲-۲-۳ تبدیل می‌شوند.

یادآوری- برخی از انواع مختلف در استاندارد ISO/TS 19103 تعریف نشده و برخی از کلاسهای انواع داده‌ی ساختار یافته با استفاده از کلیشه <<DataType>> نیز تبیین نشده‌اند. بنابراین مندرجات در زیربندهای پ-۵-۲-۲-۲ و پ-۵-۲-۲-۳ تعاریف انواع داده در استانداردISO/TS 19103 را تفسیر می‌کنند.

پ-۵-۲-۲-۲ چندگانگی

یک کلاس چندگانگی Multiplicity به عنوان محدوده‌ای از حد پایین به حد بالا در استاندارد ISO/TS 19103 تعریف می‌شود. در این‌جا به عنوان <<DataType>> تفسیر شده و به صورت زیر تعریف می‌شود:

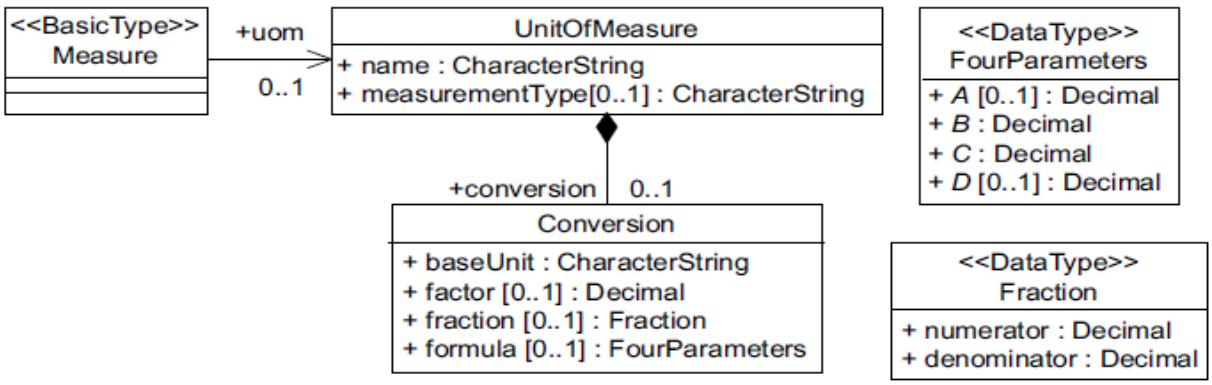
```
<xs:complexType name="Multiplicity">
    <xs:sequence>
        <xs:element name="range" type="MultiplicityRange" minOccurs="1"
    maxOccurs="unbounded"/>
```

```
        </xs:sequence>
</xs:complexType>

<xs:complexType name="MultiplicityRange">
        <xs:sequence>
                <xs:element name="lower" type="xs:nonNegativeInteger"/>
                <xs:element name="upper" type="UnlimitedInteger"/>
        </xs:sequence>
</xs:complexType>
```

پ-۵-۲-۲-۳ واحدهای اندازه‌گیری

هدف انواع واحدهای اندازه‌گیری که در استاندارد ISO/TS 19103 تعریف شده این است که به عنوان تعاریف سیستمهای اندازه‌گیری بین‌المللی و محلی مورد استفاده قرار بگیرند. بنابراین کمیت‌ها مطابق با سیستم واحد اندازه‌گیری می‌شوند که واحدهای آن احتمالا در یک لغت‌نامه واحدهای اندازه‌گیری منتشر شده است. متأسفانه درک مدل‌ها دشوار است، هیچ مثالی ارائه نشده و توضیحات مبهم هستند. پیشنهاد واحدهای توصیه اندازه گیری POSC[1] از واحدهای جایگزین برای استاندارد ISO/TS 19103 می‌باشد. یک نمودار UML که برخی ایده‌های اصلی مشخصات را نشان دهد در شکل پ-۶ نمایش داده می‌شود. نمونه‌ای از UnitOfMeasure یک سیستم اندازه‌گیری را تعریف کرده، آن را نام‌گذاری می‌کند و در صورت لزوم اطلاعاتی درباره چگونگی تبدیل کمیت‌ها با استفاده از یک واحد اصلی با فرمول کلی

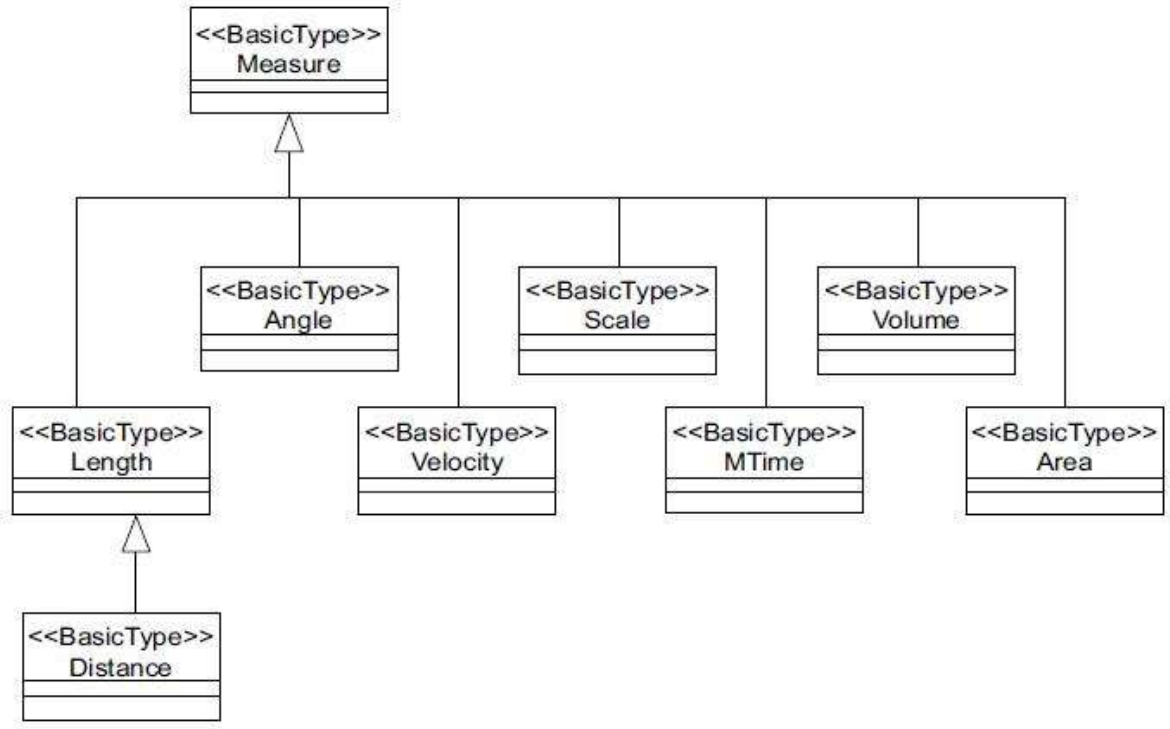
$Y = (A + BX)/(C + DX)$ ارائه می‌دهد، X مقدار واحدی است که باید تبدیل شود و Y واحد اصلی می‌باشد. اگر A=D=0 و C =1 ، بنابراین B عامل تبدیل خواهد بود. اگر A =D =0، عامل تبدیل با استفاده از یک کسر توضیح داده می‌شود. در غیر این‌صورت، با استفاده از چهار پارامتر تعریف می‌شود. یک نمونه از یک Measure مقدار اعشاری با اشاره به واحد اندازه‌گیری می‌باشد.



شکل پ- ۶- واحدهای اندازه‌گیری

1- Petroleum Open Standards Consortium

پیشنهاد می‌شود از تعاریف الگوی XML همانطور که در واحدهای پیشنهاد اندازه‌گیری POSC تعریف شده، استفاده شود.



شکل پ- ۷- انواع Measure

انواع اندازه‌گیری تعریف‌شده توسط استاندارد ISO/TS 19103 (length، angle، velocity، scale، Mtime، area و volume) به صورت زیر است.

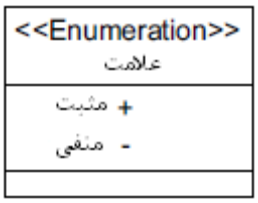
```
<xs:complexType name="Measure">
	<xs:simpleContent>
		<xs:extension base="Decimal">
			<xs:attribute name="uom" type="URI"/>
		</xs:extension>
	</xs:simpleContent>
</xs:complexType>

<xs:complexType name="Length">
	<xs:simpleContent>
		<xs:extension base="Measure"/>
	</xs:simpleContent>
</xs:complexType>

<xs:complexType name="Angle">
	<xs:simpleContent>
		<xs:extension base="Decimal"/>
```

```
        </xs:simpleContent>
</xs:complexType>

<xs:complexType name="Scale">
        <xs:simpleContent>
                <xs:extension base="Measure"/>
        </xs:simpleContent>
</xs:complexType>

<xs:complexType name="Area">
        <xs:simpleContent>
                <xs:extension base="Measure"/>
        </xs:simpleContent>
</xs:complexType>
<xs:complexType name="Velocity">
        <xs:simpleContent>
                <xs:extension base="Measure"/>
        </xs:simpleContent>
</xs:complexType>

<xs:complexType name="Mtime">
        <xs:simpleContent>
                <xs:extension base="Measure"/>
        </xs:simpleContent>
</xs:complexType>

<xs:complexType name="Distance">
        <xs:simpleContent>
                <xs:extension base="Measure"/>
        </xs:simpleContent>
</xs:complexType>

<xs:complexType name="Volume">
        <xs:simpleContent>
                <xs:extension base="Measure"/>
        </xs:simpleContent>
</xs:complexType>
```

پ-۵-۲-۳ <<Enumeration>>

یک کلاس کلیشه با نام برشمارش <<Enumeration>> باید به یک نوع ساده که یک رشته متن را به تعدادی از مقادیر برشمارش محدود می‌کند، تبدیل شود.

مثال: شمارش علائم که بر مبنای یک رشته بوده و مقدرا آن را به مقدار «+»، «positive»، «-» یا «negative» محدود می‌کند.



شکل پ- ۸- مثالی از <<Enumeration>>

```
<xs:simpleType name="Sign">
	<xs:restriction base="xs:string">
		<xs:enumeration value="+"/>
		<xs:enumeration value="-"/>
		<xs:enumeration value="positive"/>
		<xs:enumeration value="negative"/>
	</xs:restriction>
</xs:simpleType>
```

پ-۵-۲-۴ <<CodeList>>

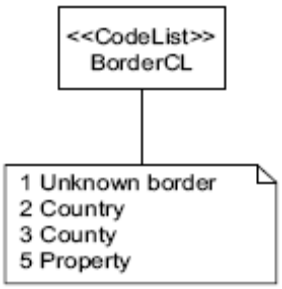
کلاس کلیشه فهرست‌کد <<CodeList>> نباید به الگوی خروجی تبدیل شود اما ممکن است در عوض به یک لغت‌نامه که کد و جفت‌های مقداری تعریف‌شده را در فهرست‌کد ذخیره می‌کند، نگاشت شود. این لغت-نامه باید در دسترس عموم باشد و آدرس وب آن باید به صورت یک شناسه منبع یکسان (URI) [1] ارائه شود.

صفت نوع فهرست‌کد باید به عنوان یک مقدار رشته‌ای کدگذاری شود.

هم‌چنین به شکل پ-۲۲ برای کلاس CodeListExtraction و شکل پ-۲۷ برای نمایش فهرست‌کد مراجعه شود.

مثال: شکل پ-۹ یک فهرست‌کد به نام BorderCL را نشان می‌دهد که به یک لغت‌نامه در XML نگاشت می‌شود. به این نکته توجه شود که به جای فهرست‌بندی جفت‌های مقدار کد به عنوان صفت مربوط به BorderCL، از صندوق توضیح [2] استفاده شده است.

1 -Uniform Resource Indicator
2 -Comment box



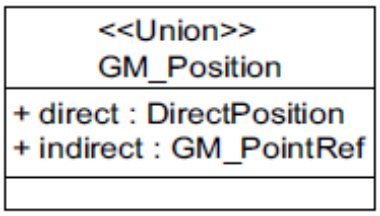
شکل پ- ۹- مثالی از <<CodeList>>

```
<codelist name="BorderCL">
	<codevalue code="1" value="Unknown border"/>
	<codevalue code="2" value="Country"/>
	<codevalue code="3" value="County"/>
	<codevalue code="5" value="Property"/>
</codelist>
```

پ-۵-۲-۵ <<Union>>

یک کلیشه کلاس <<Union>> چند صفت را فهرست می‌کند و مفهوم آن، این است که در هر زمان فقط یکی از آن صفات را ارائه می‌کند. این نوع باید به یک تعریف نوع مرکب تبدیل شود که در آن صفات به عنوان اجزای ساختار انتخابی می‌باشند

مثال: اتحاد GM_Position از منبع [۱۴] کتاب‌نامه گرفته می‌شود و به نوع مرکب با اعلان‌های جز محلی مناسب در ساختار انتخابی نگاشت می‌گردد.



شکل پ- ۱۰- مثالی از <<Union>>

```
<xs:complexType name="GM_Position">
	<xs:choice>
		<xs:element name="direct" type="DirectPosition"/>
		<xs:element name="indirect" type="GM_PointRef"/>
	</xs:choice>
</xs:complexType>
```

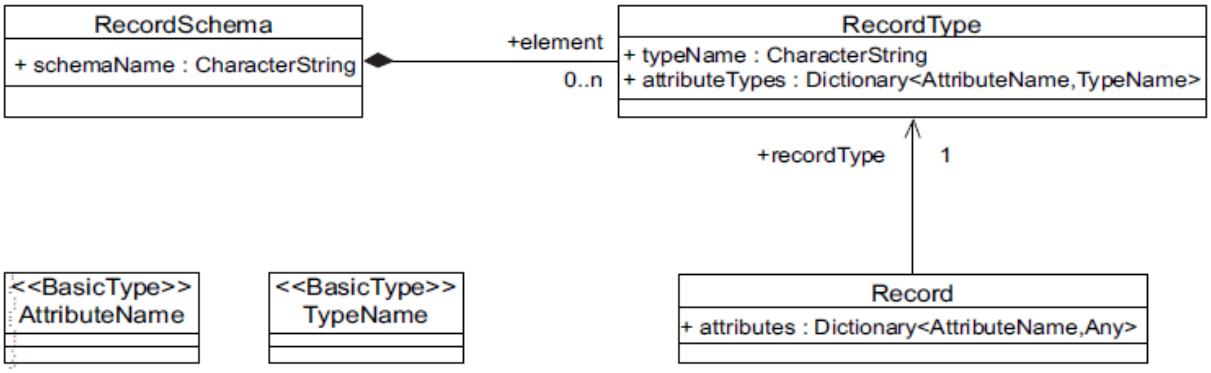
پ-۵-۲-۶ انواع شی

پ-۵-۲-۶-۱ قاعده کلی

کلاسی که هیچ کلیشه ندارد یا کلیشه‌ای از نوع <<Type>> باشد، باید به یک نوع مرکب با نامی مشابه نام کلاس نگاشت شود. تعریف نوع مرکب باید شامل صفت‌های شناسایی باشد، که یا از IM_Object به ارث برده و یا از گروه صفت IM_ObjectIdentificaton ارجاع داده شده است.

پ-۵-۲-۶-۲ انواع رکورد

تعدادی از انواع رکورد در استاندارد ISO/TS 19103 تعریف شده است. این‌ها دوباره در شکل پ-۱۱ مدل-سازی شده و تفسیر می‌شوند. AttributeName و TypeName به عنوان انواع اصلی براساس رشته حروف مدل‌سازی می‌شوند.



شکل پ- ۱۱- انواع رکورد

```
<xs:complexType name="RecordSchema">
	<xs:sequence>
		<xs:element name="schemaName" type="CharacterString"/>
		<xs:element name="element" type="RecordType" minOccurs="0"
	maxOccurs="unbounded"/	>
	</xs:sequence>
	 <xs:attributeGroup ref="IM_ObjectIdentification"/>
</xs:complexType>
<xs:complexType name="RecordType">


<xs:sequence>
	<xs:element name="typeName" type="CharacterString"/>
		<xs:element name="attributeTypes"
	type="Dictionary_AttributeName_TypeName_"/>
		</xs:sequence>
	<xs:attributeGroup ref="IM_ObjectIdentification"/>
	</xs:complexType>
<xs:complexType name="Record">
```

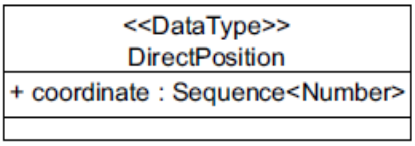
```
<xs:sequence>
  <xs:element name="attributes" type="Dictionary_AttributeName_Any_"/>
            <xs:element name="recordType" type="ref_RecordType"/>
          </xs:sequence>
        <xs:attributeGroup ref="IM_ObjectIdentification"/>
</xs:complexType>
```

پ- ۵-۲-۷ نوع قالب حدی[1]

نوع قالب حدی نوعی است که پارامترها به مقادیر آرگومان[2] واقعی محدود می‌شوند. استاندارد ISO/TS 19103 پنج نوع قالب مختلف را تعریف می‌کند: Set<T>، Bag<T>، Sequence<T>، CircularSequence<T> و Dictionary<K,V>. چهار مورد اول دارای یک پارامتر هستند درحالی‌که Dictionary دو پارامتر به خود می‌گیرد. این موارد به طور معمول به اعلان‌های صفت محدود می‌شوند؛ به صفت «attributeTypes» از نوع رکورد که در شکل پ- ۱۱ توصیف شده مراجعه کنید.

- نوع قالب حدی باید به یک نوع تعریف مرکب تبدیل شود که به نوع قالب مربوط است. اعلان نوع باید نام‌گذاری شود. این نام باید از به هم پیوستن نام قالب با نام‌های آرگومان ساخته شود که حروف آن با زیرخط «_» از هم جدا می‌شوند. نویسه‌های کوچک‌تر از «<»، ویرگول «،» و بزرگ‌تر از «>» را نمی‌توان در نام‌ها استفاده کرد.
- یک مجموعه Set، Bag، Sequence یا نوع قالب CircularSequence باید به یک تعریف نوع مرکب که شامل ساختار دنباله ای از چندگانگی نامحدود که حاوی جزئی با نام و نوعی منطبق با نوع پارامتر منفرد بوده، نگاشت شود. اگر نوع پارامتر از انواع اصلی باشد، بنابراین ممکن است از تعریف نوع ساده با به‌کارگیری فهرستی از ساختار الگوی XML استفاده شود.
- لغت‌نامه نوع قالب حدی باید به یک تعریف نوع مرکب متشکل از یک ساختار دنباله‌ای از چندگانگی نامحدود با دو جز دارای نام و نوع مطابق با ویژگی دو پارامتری، نگاشت شود.

مثال ۱: مکان مستقیم نوع داده در منبع [۱۴] کتاب‌نامه تعریف شده که در شکل پ-۱۲ نشان داده می‌شود.



مثال پ- ۱۲- مثالی از نوع قالب محدود

دنباله <Number> یک نوع قالب حدی را تعریف می‌کند و مطابق با قاعده کلی به شکل زیر نگاشت می‌شود:

1-Bound template type
2- Argument

```
<xs:complexType name="Sequence_Number_">
	<xs:sequence maxOccurs="unbounded">
		<xs:element name="Number" type="Number"/>
	</xs:sequence>
</xs:complexType>
```

از آن‌جایی که نوع آرگومان یک نوع اصلی بوده، می توان آن را نیز به روش زیر نگاشت کرد :

```
<xs:simpleType name="Sequence_Number_">
	<xs:list itemType="Number"/>
</xs:simpleType>
```

مثال ۲: انواع قالب حدی که در شکل پ-۱۱ تعریف شده به صورت زیر نگاشت می‌شود.

```
<xs:complexType name="Dictionary_AttributeName_Any_">
	<xs:sequence maxOccurs="unbounded">
		<xs:element name="AttributeName" type="CharacterString"/>
		<xs:element name="Any" type="xs:anyType"/>
	</xs:sequence>
</xs:complexType>

<xs:complexType name="Dictionary_AttributeName_TypeName_">
	<xs:sequence maxOccurs="unbounded">
		<xs:element name="AttributeName" type="CharacterString"/>
		<xs:element name="TypeName" type="CharacterString"/>
	</xs:sequence>
</xs:complexType>
```

پ-۵-۲-۸ وراثت

پ-۵-۲-۸-۱ کلیات

سازوکار وراثت در UML اجازه می‌دهد که یک زیرنوع، صفت‌ها و وابستگی‌های نوع بالاتر خود ارث ببـرد. در وراثت منفرد، یک نوع می‌تواند تنها از یک ابرنوع ارث ببرد، در حالی‌که در وراثت چندگانه یک نوع می‌تواند از بیش از یک ابرنوع ارث ببرد. در UML هر دو وراثت ساده و چندگانـه مجـاز مـی‌باشـد. الگـوی XML فقـط وراثت منفرد را پشتیبانی می‌کند. لذا ضروری است در XML وراثت چندگانه شبیه‌سازی گردد.

وراثت با یکی از موارد زیر تحقق می‌یابد:

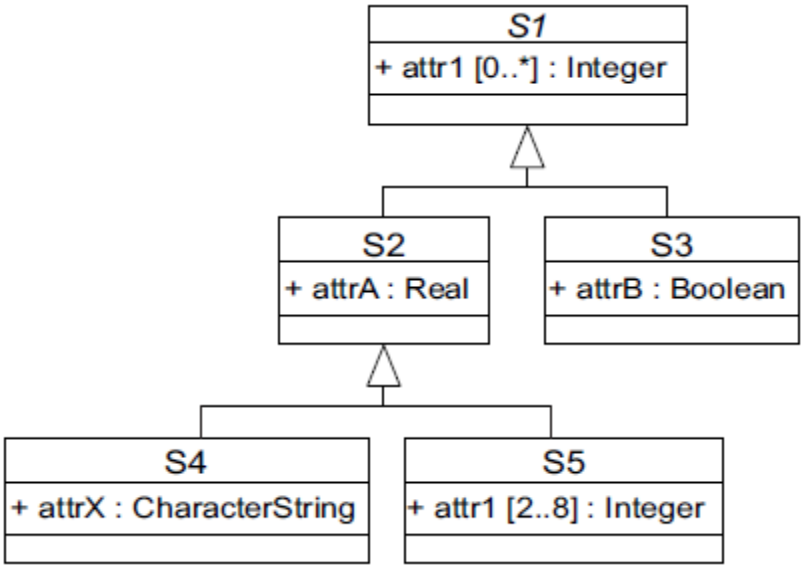
- توسط سازوکار گسترش یا محدودیت الگوی XML به نام *وراثت منفرد* (پ-۵-۲-۸-۲)، یا
- توسط نسخه‌برداری صفت‌ها و وابستگی‌ها از ابرنوع به نـوع هـدف کـه *وراثـت چندگانـه* (پ-۵-۲-۸-۲) نامیده می‌شود.

در حالت وراثت چندگانه، صفت‌ها و وابستگی‌ها باید درون نوع هدف نسخه‌برداری شوند.

پ-۵-۲-۸-۲ وراثت منفرد

قاعده‌ی عمومی باید سازوکار گسترش الگوی XML را برای انواع مرکب مورد استفاده قرار بدهد. اما اگر یک صفت یا وابستگی دوباره تعریف شود، باید سازوکار محدودیت مورد استفاده قرار گیرد.

مثال: شکل پ-۱۳ ابرنوع S1 با زیرنوع[1]های S2 و S3 را نشان می‌دهد. S1 یک کلاس انتزاعی است. S4 و S5 زیرنوع‌های S2 هستند. توجه شود که S5، «aatr1» را دوباره تعریف می‌کند و بنابراین استفاده از سازوکار محدودیت ضروری می‌باشد.



شکل پ- ۱۳- مثالی از وراثت منفرد

```
<xs:complexType name=”S1” abstract=”true”>
    <xs:complexContent>
        <xs:extension base="IM_Object">
            <xs:sequence>
                <xs:element name=”attr1" type="Integer" minOccurs="0"
            maxOccurs="unbounded"/>
            </xs:sequence>
        </xs:extension>
    </xs:complexContent>
</xs:complexType>

<xs:complexType name=”S2”>
    <xs:complexContent>
        <xs:extension base="S1">
            <xs:sequence>
                <xs:element name=”attrA" type="Real"/>
```

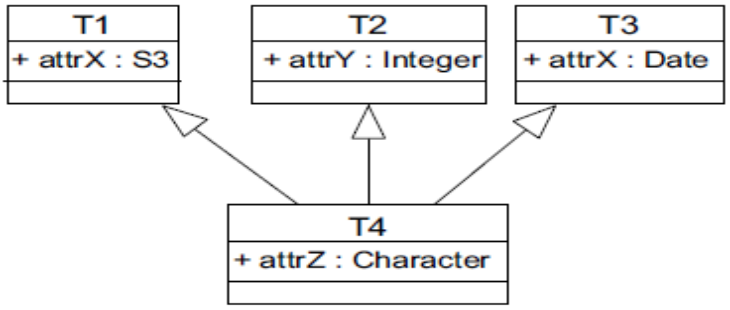
1 - subtype

```
                </xs:sequence>
            </xs:extension>
        </xs:complexContent>
</xs:complexType>

<xs:complexType name=”S3”>
        <xs:complexContent>
            <xs:extension base="S1">
                <xs:sequence>
                    <xs:element name=”attrB" type="Boolean"/>
                </xs:sequence>
            </xs:extension>
        </xs:complexContent>
</xs:complexType>

<xs:complexType name=”S4”>
        <xs:complexContent>
            <xs:extension base="S2">
                <xs:sequence>
                    <xs:element name=”attrX" type="CharacterString"/>
                </xs:sequence>
            </xs:extension>
        </xs:complexContent>
</xs:complexType>

<xs:complexType name=”S5”>
        <xs:complexContent>
            <xs:restriction base="S2">
                <xs:sequence>
                    <xs:element name=”attr1" type="Integer" minOccurs="2"
                maxOccurs="8"/>
                    <xs:element name="attrA" type="Real"/>
                </xs:sequence>
            </xs:restricton>
        </xs:complexContent>
</xs:complexType>
```

پ-۵-۲-۸-۳ وراثت چندگانه

روند نسخه‌برداری صفت‌ها/وابستگی‌ها با ابرنوع سمت چپ آغاز شده و صفت و وابستگی‌های آن را نسخه-برداری می‌کند، سپس با ابرنوع بعدی در سمت راست ادامه می‌دهد تا به آخرین ابرنوع در سمت راست برسد. در نهایت صفت‌های زیرنوع اضافه می‌شوند. در صورتی که یک ابرنوع یا زیرنوع یک صفت یا وابستگی را با نام قبلی نسخه‌برداری کند یک تناقض اتفاق می‌افتد. در حالت تناقض نام، آخرین صفت یا وابستگی باید آخرین اولویت را بگیرد و با مورد نسخه‌برداری شده قبلی جایگزین شود.

مثال: شکل پ-۱۴ چهار نوع T3، T2،T1 و T4 را توصیف می‌کند. T4 زیرنوع T2،T1 و T3 می‌باشد.



شکل پ- ۱۴- مثالی از وراثت چند گانه

```
<xs:complexType name=”T1”>
	<xs:sequence>
		<xs:element name=”attrX" type="S3"/>
	</xs:sequence>
	<xs:attributeGroup ref="IM_ObjectIdentification"/>
</xs:complexType>

<xs:complexType name=”T2”>
	<xs:sequence>
		<xs:element name=”attrY" type="Real"/>
	</xs:sequence>
	<xs:attributeGroup ref="IM_ObjectIdentification"/>
</xs:complexType>

<xs:complexType name=”T3”>
	<xs:sequence>
		<xs:element name=”attrX" type="Date"/>
	</xs:sequence>
	<xs:attributeGroup ref="IM_ObjectIdentification"/>
</xs:complexType>

<xs:complexType name=”T4”>
	<xs:sequence>
		<xs:element name=”attrX" type="Date"/>
		<xs:element name=”attrY" type="Integer"/>
		<xs:element name=”attrZ" type="Character"/>
	</xs:sequence>
	<xs:attributeGroup ref="IM_ObjectIdentification"/>
</xs:complexType>
```

پ-۵-۲-۹ انواع جایگزین

استفاده از یک ابرنوع به عنوان یک صفت به این معنی است که یک نمونه از صفت می‌تواند یکی از زیرنوعهای ثابت باشد که توسط سلسله‌مراتب وراثت ابرنوع تعریف می‌شود. الگوی XML از این سازوکار نوع پویا به طور مستقیم پشتیانی نمی‌کند.

سه رویکرد جایگزین می‌تواند مورد استفاده قرار بگیرد:

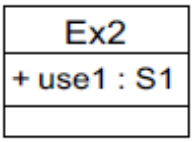
الف- یک اعلان جز استاندارد با نوع مرتبط با ابرنوع اعلان شود. در فایل تبادل، صفت xsi:type باید برای نشان دادن نوع الزامیمورد نیاز استفاده شود. به مثال ۱ مراجعه شود.

ب- جزهای سراسری با یک گروه جایگزینی که با سلسله‌مراتب وراثتی ابرنوع سازگاری دارد، تعریف شود. جز سراسری باید در یک اعلان جز ارجاع‌دهی شود. به مثال ۲ مراجعه شود.

پ- گروههای انتخاب برای هر ابرنوع که شامل انتخاب اعلان‌های جز برای انواع اصلی در سلسله‌مراتب وراثتی ابرنوع بوده، تعریف شود. گروه انتخابی باید به اعلان جز اشاره کند. به مثال ۳ مراجعه شود.

اگر سازوکار نسخه‌برداری تعریف‌شده در زیربند پ-۵-۲-۸-۳ مورد استفاده قرار بگیرد، تنها سازوکار پ باید استفاده شود.

مثال ۱: رویکرد الف: کلاس Ex2 یک صفت از نوع S1 را تعریف می‌کند که در شکل پ-۱۵ دیده می‌شود. S1 ابرنوع انتزاعی با سلسله مراتب وراثتی نشان داده شده در شکل پ-۱۳ می‌باشد.



شکل پ- ۱۵- مثالی از صفت یک ابرنوع

```
<xs:complexType name="Ex2">
    <xs:sequence>
        <xs:element name="use1" type="S1"/>
    </xs:sequence>
</xs:complexType>
```

استفاده از xsi:type در فابل تبادل:

```
<Ex1>
    <use1 xsi:type="S3">
        <attr1>42</attr1><attr1>43</attr1><attr1>44</attr1>
        <attrB>true</attrB>
    </use1>
</Ex1>
```

مثال ۲: رویکرد ب: استفاده از جزهای جایگزین سراسری.

```
<xs:element name="S1" type="S1" abstract="true"/>
<xs:element name="S2" type="S2" substitutionGroup="S1"/>
<xs:element name="S3" type="S3" substitutionGroup="S1"/>
<xs:element name="S4" type="S4" substitutionGroup="S2"/>
<xs:element name="S5" type="S5" substitutionGroup="S2"/>
<xs:complexType name="Ex2">
```

```
        <xs:sequence>
            <xs:element name="use1">
                <xs:complexType>
                    <xs:sequence>
                        <xs:element ref="S1"/>
                    </xs:sequence>
                </xs:complexType>
            </xs:element>
        </xs:sequence>
</xs:complexType>
<Ex1>
        <use1>
            <S3>
                <attr1>42</attr1><attr1>43</attr1><attr1>44</attr1>
                <attrB>true</attrB>
            </S3>
        </use1>
</Ex1>
```

مثال ۳: رویکرد پ: استفاده از گروه‌های انتخاب

```
<xs:group name="S1">
        <xs:choice>
            <xs:element name="S2" type="S2"/>
            <xs:element name="S3" type="S3"/>
            <xs:element name="S4" type="S4"/>
            <xs:element name="S5" type="S5"/>
        </xs:choice>
</xs:group>
<xs:complexType name="Ex2">
        <xs:sequence>
            <xs:element name="use1">
                <xs:complexType>
                    <xs:group ref="S1"/>
                </xs:complexType>
            </xs:element>
        </xs:sequence>
</xs:complexType>
```

نمونه به طور دقیق مشابه نمونه‌ی مثال ۲ می‌باشد. مزیت این رویکرد این است که هیچ جز سراسری وجـود ندارد.

پ- ۵-۲-۱۰ <<Abstract>> یا کلاس انتزاعی

اگر از سازوکار نسخه‌برداری استفاده شود یک کلاس انتزاعی نمی‌تواند به عنوان اعلان نوع مرکب ارائه شـود. اگر وراثت منفرد مورد استفاده قرار بگیرد، یک کلاس انتزاعی باید به تعریف نوع مرکب مطابق با زیربنـد پ-۵-۲-۶ که دارای صفت «abstract» بوده که به درستی تنظیم شده، تبدیل شود.

پ-۵-۲-۱۱ <<ExternalType>>

همه کلیشه‌های کلاس <<ExternalType>> یا باید به نوع ساده‌ای که نوع الگوی XML را با هر URI محدود می‌کند، یا به یک نوع NOTATION. نگاشت شود

مثال:

```
<xs:simpleType name="Modis">
    <xs:restriction base="xs:anyURI"/>
</xs:simpleType>
```

پ-۵-۳ اعلان‌های جز ویژگی

پ-۵-۳-۱ صفت

یک صفت، ویژگی‌های یک کلاس را تعریف می‌کند. یک صفت باید به یک *اعلان جز* یا *اعلان صفت* در اعلان نوع مرکب کلاس تبدیل شود. قاعده پیش‌فرض *اعلان جز* می‌باشد. اگر یک صفت نوع اصلی دارای چندگانگی صفر یا یک باشد، ممکن است به یک *اعلان صفت* تبدیل شود. تمام صفت‌های دیگر باید به یک *اعلان جز* تبدیل شوند.

قاعده پیش‌فرض، تبدیل تمام صفت‌ها و صفت‌های به دست‌آمده می‌باشد.

اعلان صفت، یک صفت را با یک نام و یک نوع داده اصلی تعریف می‌کند. چندگانگی یک صفت باید مطابق با جدول پ-۴ باشد. چندگانگی پیش‌فرض یک اعلان صفت، در الگوی XML اختیاری است.

جدول پ-۴- نگاشت چندگانگی صفت

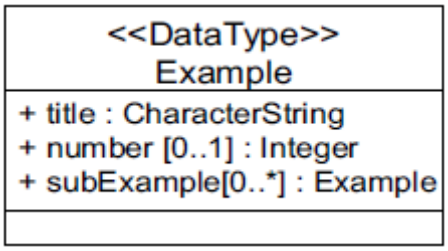
UML	اختیاری	اعلان صفت ضروری
۱ (پیش‌فرض)	نادرستی	اختیاری =«false»
۱..۰	درست (پیش‌فرض)	-

اعلان جز یک جز را با یک نام و یک نوع تعریف می‌کند. چندگانگی باید مطابق با جدول پ-۵ باشد. مقادیر پیش‌فرض در جدول نشان داده می‌شوند و اعلان آن‌ها لزومی ندارد. در عوض هر دو مقدار کمینه و بیشینه ممکن است ارائه شوند.

جدول پ-۵- نگاشت چندگانگی برای جزهای محتوا

UML	minOccurs	maxOccurs	بیان مشخصه‌های ضروری
۱ (پیش‌فرض)	۱ (پیش‌فرض)	۱ (پیش‌فرض)	-
۱..۰	۰	۱ (پیش‌فرض)	minOccurs=”0”
*..۰	۰	نامحدود	minOccurs=”0” maxOccurs=”unbounded”
*..۱	۱ (پیش‌فرض)	نامحدود	maxOccurs=”unbounded”
۸..۲	۲	۸	minOccurs=”2” maxOccurs=”8”

مثال: یک مثال نوع داده در UML بیان شده است. که دارای سه صفت «title» و «number» و «subExample» است. به شکل پ-۱۶ مراجعه شود. هر دو صفت «title» و «number» می‌تواند به اعلان صفت تبدیل شود، درحالی‌که «subExample» یک نوع مرکب بوده و ضروری است که به یک اعلان جز نگاشت شود.



شکل پ- ۱۶- صفت مثال

اعلان پیش‌فرض جز موارد زیر را ارائه می‌دهد:

```
<xs:complexType name=”Example” >
    <xs:sequence>
        <xs:element name="title" type="CharacterString"/>
        <xs:element name="number" type="Integer" minOccurs="0"/>
        <xs:element name="subExample" type="Example" minOccurs="0"
    maxOccurs="unbounded"/>
    </xs:sequence>
</xs:complexType>
```

در این‌جا «title» و «number» مطابق با قاعده‌ی اعلان صفت، تبدیل می‌شود.

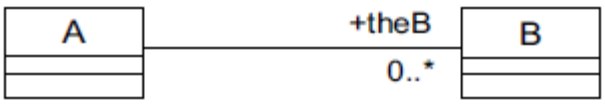
```
<xs:complexType name=”Example” >
    <xs:sequence>
        <xs:element name="subExample" type="Example" minOccurs="0"
    maxOccurs="unbounded"/>
    </xs:sequence>
    <xs:attribute name="title" type="CharacterString" optional="false"/>
    <xs:attribute name="number" type="Integer"/>
</xs:complexType>
```

پ- ۵-۳-۲ وابستگی

وابستگی، ارتباط کلی بین دو کلاس را تعریف می‌کند. در ادامه، یکی از کلاس‌ها، کلاس منبع و دیگری کلاس هدف نامیده می‌شود. اشیا منبع، مراجع را روی اشیای هدف و بالعکس ذخیره می‌کنند.

- در صورتی‌که وابستگی قابل هدایت بوده و کلاس هدف با یک نام نقش شناسایی شود، نوع مرکب مطابق با کلاس منبع باید حاوی یک اعلان جز باشد. نام جز، باید نام نقش شناسایی کلاس هدف باشد، و نوع آن یا باید *IM_ObjectReference* باشد، یا نوعی بر مبنای صفت تعریف شده در *IM_ObjectReference*. چندگانگی اعلان جز باید بر اساس جدول پ-۵ باشد.
- در صورتی‌که وابستگی قابل هدایت بوده و کلاس منبع با یک نام نقش شناسایی شود، نوع مرکب مطابق با کلاس هدف باید حاوی یک اعلان جز باشد. نام جز، باید نام نقش شناسایی کلاس منبع باشد، و نوع آن یا باید *IM_ObjectReference* باشد، یا نوعی بر مبنای صفت تعریف شده در *IM_ObjectReference*. چندگانگی اعلان جز باید بر اساس جدول پ-۵ باشد.

مثال: یک وابستگی بین کلاس A و B تعریف شده است، به شکل پ-۱۷ مراجعه شود. از آن‌جایی که فقط یک نام نقش تعریف شده است، فقط کلاس A کلاس B را می‌شناسد.



شکل پ- ۱۷- مثال وابستگی

اعلان‌های الگوی XML به صورت زیر می‌باشد:

```
<xs:complexType name="A">
    <xs:sequence>
        <xs:element name=”theB” type=”ref_B” minOccurs=”0”
    maxOccurs=”unbounded”/>
    </xs:sequence>>
</xs:complexType>

<xs:complexType name="ref_B">
    <xs:attributeGroup ref="IM_ObjectReference"/>
</xs:complexType>
```

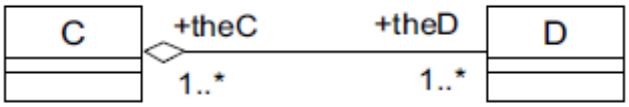
پ- ۵-۳-۳ تجمیع

تجمیع یک ارتباط ضعیف رابطه کل- جز بین یک تجمیع (کلی) و یک جز جزئی را تعریف می‌کند. مالکیت در جزهایی که به طور هم‌زمان می‌توانند عضو بیش از یک تجمیع باشند ضعیف است. بنابراین شی جز ممکن است توسط بیش از یک شی تجمیع به اشتراک گذاشته شود. بنابراین یک تجمیع می‌تواند به‌طور

عمومی مراجع را فقط برای قسمت‌های خود ذخیره کند، اما در حالت مالکیت کامل شامل قسمت‌های مرتبط می‌باشد.

- نوع مرکب مربوط به کلاس تجمیع باید دارای یک اعلان جز باشد که نام آن مطابق با نام نقش شناسایی کلاس جز باشد. چندگانگی این جز باید مطابق با جدول پ-۵ باشد. نوع جز باید براساس IM_ObjectReference بوده و ممکن است حاوی صفر یا یک جز از نوعی باشد که به کلاس جز مرتبط است.
- درصورتی‌که وابستگی قابل هدایت بوده و کلاس هدف با یک نام نقش تعریف شود، نوع مرکب مطابق با کلاس جز باید حاوی اعلان جز باشد. نام جز باید با نام نقش شناسایی کلاس تجمیع مطابقت داشته و نوع آن باید بر مبنای یک قلم IM_ObjectReference باشد. چندگانگی اعلان جز باید بر اساس جدول پ-۵ باشد.

مثال: تجمیع بین کلاس تجمیع C و کلاس بخش D تعریف شده است، به شکل پ-۱۸ مراجعه شود. کلاس C، کلاس D را با نام نقش «theD» شناسایی کرده و کلاس D، کلاس C را با نام نقش «theC» شناسایی می‌کند.



شکل پ- ۱۸- مثال تجمیع

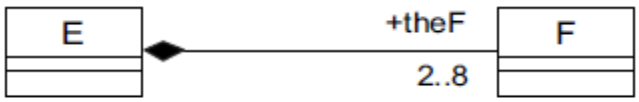
```
<xs:complexType name="C">
    <xs:sequence>
        <xs:element name="theD" minOccurs="1" maxOccurs="unbounded">
            <xs:complexType>
                <xs:sequence>
                    <xs:element name="D" type="D" minOccurs="0"
                maxOccurs="1"/>
                </xs:sequence>
                <xs:attributeGroup ref="IM_ObjectReference"/>
            </xs:complexType>
        </xs:element>
    </xs:sequence>
</xs:complexType>

<xs:complexType name="D">
    <xs:sequence>
        <xs:element name="theC" type="IM_ObjectReference" minOccurs="1"
    maxOccurs="unbounded"/>
    </xs:sequence>
</xs:complexType>
```

پ- ۵-۳-۴- ترکیب

ترکیب یک ارتباط قوی کل- بخش بین یک ترکیب (کل) و یک بخش جز تعریف می‌کند. مالکیت در بخشی که می‌تواند به طوردقیق عضوی از یک شی ترکیب بوده، قوی می‌باشد. بنابراین یک ترکیب باید حاوی بخش‌های مرتبط باشد.

- نوع مرکب که به کلاس ترکیب مربوط است باید حاوی یک اعلان جز باشد که نام آن به نام نقش شناسایی کلاس بخش مربوط بوده و نوع آن نیز به نوع کلاس بخش مرتبط باشد. چندگانگی اعلان جز باید بر اساس جدول پ-۵ باشد.
- نوع مرکب مربوط به کلاس بخش نباید حاوی هیچ اعلان جز باشد، حتی اگر یک نام نقش کلاس ترکیب را تعریف کند. این مسأله ضمنی است زیرا یک بخش همیشه در داخل یک کلاس ترکیب قرار میگیرد.

مثال: ترکیب بین کلاس‌های E و F تعریف شده‌است، به شکل پ-۱۹ مراجعه کنید. کلاس E، کلاس F را با یک نام نقش هدف «theF» شناسایی می‌کند.



شکل پ- ۱۹- مثال ترکیب

```
<xs:complexType name="E">
    <xs:sequence>
        <xs:element name="theE" type="F" minOccurs="2" maxOccurs="8"/>
    </xs:sequence>
</xs:complexType>
```

پ-۵-۴ اعلان جز

پ- ۵-۴-۱ ساختار سند

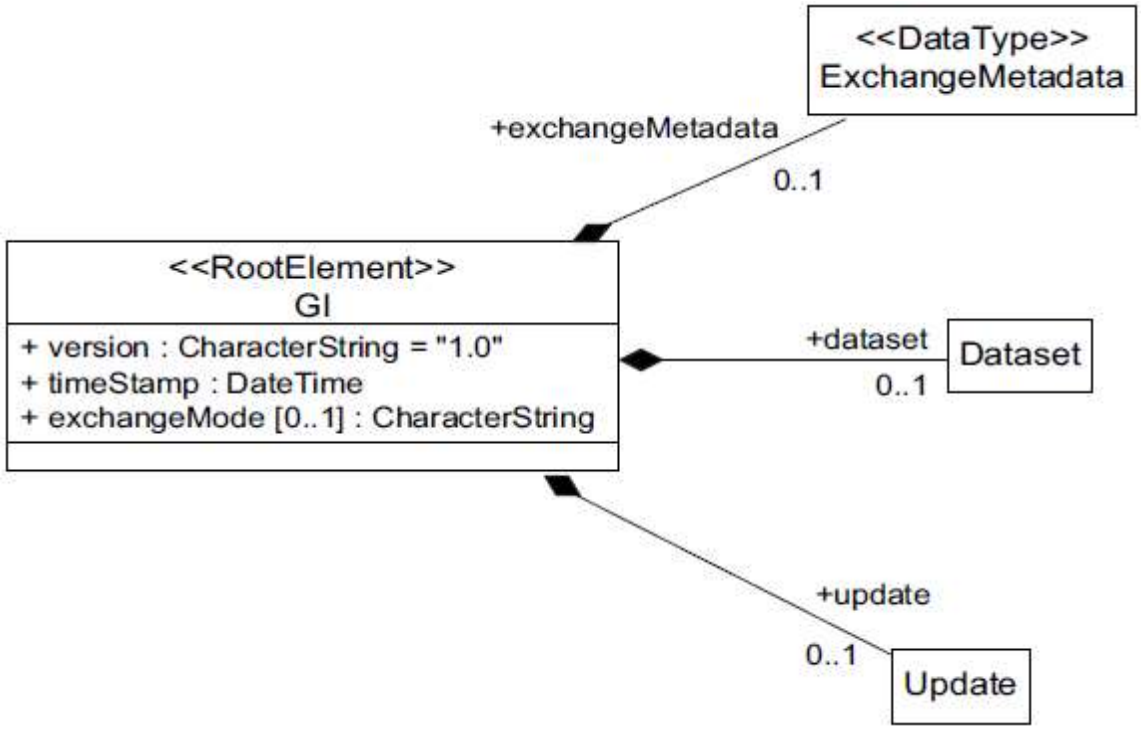
جز GI باید جز ریشه‌ی فایل تبادل باشد، که دارای سه جز است: exchangeMetadata، dataset و update؛ به شکل پ-۲۰ مراجعه شود. جزهای مجموعه داده و به‌روزرسانی ممکن است دارای شناسه باشند، درحالی‌که فراداده تبادل این‌گونه نیست.

صفت‌های جز GI به صورت زیر می‌باشد:

- version: CharacterString = "1.0;"
- timeStamp: DateTime;
- exchangeMode [0..1] : CharacterString.

صفت version لازم است روی «۱/۰» تنظیم شود. این مسأله نشان می‌دهد که فایل تبادل با این نسخه از استاندارد مطابقت دارد. نسخه‌های ویرایش‌شده این استاندارد دارای عدد دیگری است که به آن‌ها مربوط

می‌شود. صفت timestamp تاریخ و زمانی را که داده کدگذاری شده است را نشان می دهد. این اقلام مرتبط به زمانی است که فایل تبادل تولید می‌شود. صفت exchangeMode توسط کاربر تعیین می‌شود و مقدار آن ممکن است محتوا یا نوع فایل تبادل را نشان دهد.



شکل پ- ۲۰- ساختار سند

اعلان جز GI به شرح زیر است:

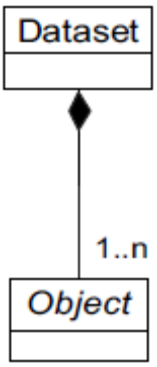
```
<xs:element name="GI">
    <xs:complexType>
        <xs:sequence>
            <xs:element name="exchangeMetadata" type="ExchangeMetadata"
        minOccurs="0"/>
            <xs:element name="dataset" type="Dataset" minOccurs="0"/>
            <xs:element name="update" type="Update" minOccurs="0"/>
        </xs:sequence>
        <xs:attribute name="version" type="CharacterString" use="required"
    fixed="1.0"/>
        <xs:attribute name="timeStamp" type="DateTime" use="required"/>
        <xs:attribute name="exchangeMode" type="CharacterString"/>
    </xs:complexType>
</xs:element>
```

پ-۵-۴-۲ جزهای مجموعه داده و شی

یک مجموعه داده از یک یا تعداد بیش‌تری جز تشکیل شده که اشیا را کدگذاری کرده، و جزهای شی نامیده می‌شوند، که باید اعلان شود.

- تمامی تعاریف انواع مرکب با صفت شناسایی، گزینه‌هایی برای اعلان جز شی محسوب می‌شوند. برخی از این‌ها نباید به عنوان اشیای مستقل در نظر گرفته شوند، و بنابراین تعریف نمی‌شوند.
- نام جز باید با نام نوع یکی باشد یا یک برچسب نام تعریف شده برای این نوع باشد. یک جز شی ممکن است به عنوان یک جز محلی یا به عنوان یک جز سراسری اعلان شود. حالت پیش‌فرض جز محلی است.
- تمام جزهای شی باید در یک گروه انتخاب که ممکن است به نام **Object** باشد،گروه‌بندی شوند. بنابراین گروه انتخاب یا باید به جزهای اعلان شده‌ی سراسری اشاره کند یا جزهای شی را به صورت محلی اعلان نماید. این گروه انتخاب باید برای محدود کردن اشیای مجاز در یک مجموعه داده مورد استفاده قرار بگیرد.

مثال: گروه شی، جزهای شی مثال در زیربند پ-۵-۲-۸-۳ را تعریف می‌کند. نوع مجموعه‌داده به یک دنباله نامحدود از گروه شی اشاره دارد (شکل پ-۲۱). این بدین معنی است که مجموعه‌داده می‌تواند تنها حاوی چهار جز T1, T2, T3 و T4 باشند.



شکل پ- ۲۱- مجموعه داده شامل اشیا

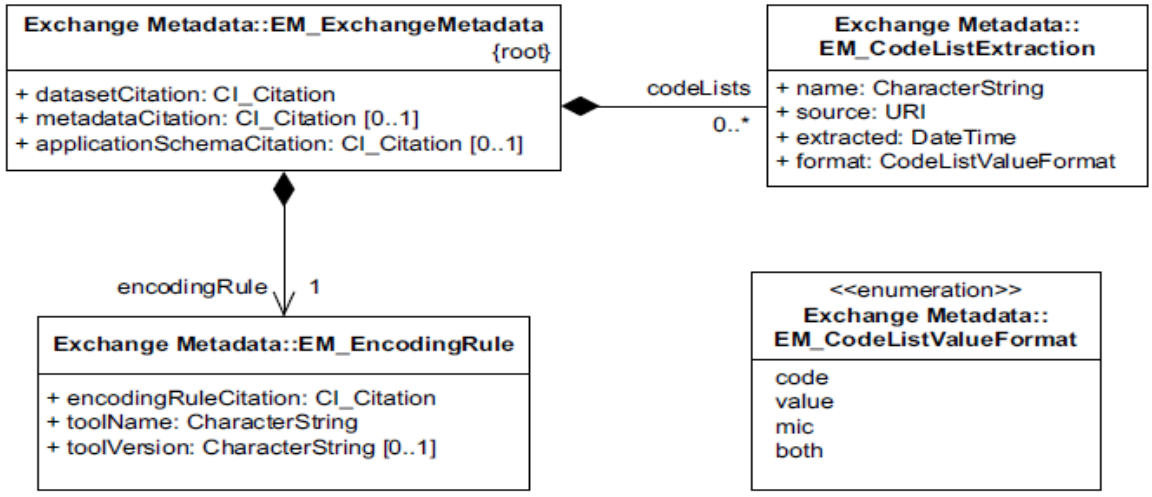
```
<xs:complexType name="Dataset">
      <xs:sequence maxOccurs="unbounded">
            <xs:group ref="Object"/>
      </xs:sequence>
      <xs:attributeGroup ref="IM_ObjectIdentification"/>
</xs:complexType>

<xs:group name="Object">
      <xs:choice>
            <xs:element name=”T1” type=”T1”/>
            <xs:element name=”T2” type=”T2”/>
            <xs:element name=”T3” type=”T3"/>
            <xs:element name=”T4” type=”T4"/>
      </xs:choice>
```

</xs:group>

پ-۵-۴-۳ فراداده‌ی تبادل

انواع فراداده‌ی تبادل در شکل پ-۲۲ تعریف می‌شوند. نوع CI_Citation از منبع [۲۱] کتاب‌نامه وارد شده و دوباره استفاده می‌شود.

یک فراداده‌ی تبادل ExchangeMetadata باید حاوی اطلاعاتی باشد که مجموعه داده را توضیح می‌دهد. صفت «datasetCitation» تولیدکننده‌ی مجموعه داده را توضیح می‌دهد. «metadataCitation» به فراداده مربوط به مجموعه داده اشاره دارد، «applicationSchemaCitation» به الگوی کاربردی مورد استفاده اشاره دارد و «configFileCitation» فایل پیکربندی مورد استفاده را توضیح می‌دهد. ترکیب «encodingRule» قاعده کدگذاری مورد استفاده برای تولید مجموعه داده را توضیح می‌دهد. اگر مجموعه‌داده شامل صفت فهرست‌های کد باشد، باید نشان‌دهنده‌ی فهرست‌های‌کد به کار برده شده و اعتبار آن‌ها نیز باشد. این مسأله توسط ترکیب «codeLists» انجام می‌شود.



شکل پ- ۲۲- فراداده تبادل

EncodingRule شامل اطلاعاتی درباره قاعده کدگذاری به‌کار رفته در تولید مجموعه‌داده می‌باشد. «encodingRuleCitation» قاعده کدگذاری مورد استفاده را توضیح می‌دهد. «toolName» نام ابزار سرویس کدگذاری است که برای تولید مجموعه داده به کار می‌رود و «toolVersion» شماره نسخه آن را نشان می‌دهد.

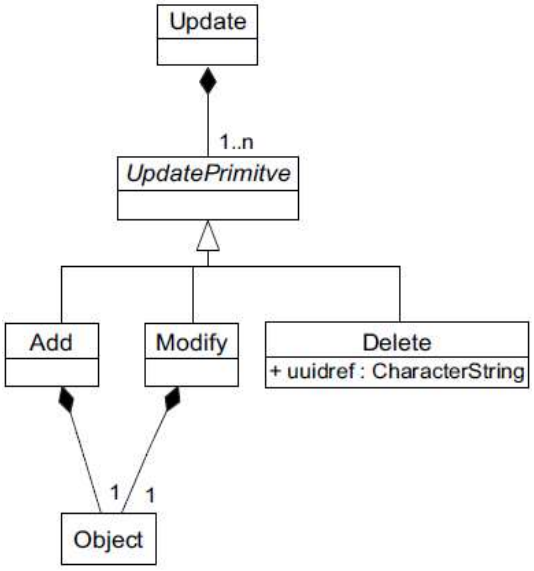
CodeListExtraction کدلیست را با استفاده از صفت‌های «name» و «source» شناسایی می‌کند. هم‌چنین مشخص کننده‌ی زمانی است که کدها مورد استفاده برای فهرست کد خاص «extracted» شده‌اند، و بنابراین، اعتبار کدها را تعیین می‌کند. صفت «format» نشان می‌دهد که آیا صفت از کد یا جز مقدار از

فهرست کد استفاده می‌کنند یا نه. آن‌ها هم‌چنین ممکن است کد و مقدار را با هم مخلوط کرده و با هم استفاده کنند.

```
<xs:complexType name="ExchangeMetadata">
	<xs:sequence>
		<xs:element name="datasetCitation" type="CI_Citation"/>
		<xs:element name="metadataCitation" type="CI_Citation" minOccurs="0"/>
		<xs:element name="applicationSchemaCitation" type="CI_Citation"
	minOccurs="0"/>
		<xs:element name="configFileCitation" type="CI_Citation" minOccurs="0"/>
		<xs:element name="encoding" type="EncodingRule"/>
		<xs:element name="codelist" type="CodeListExtraction" minOccurs="0"
	maxOccurs="unbounded"/>
	</xs:sequence>
</xs:complexType>

	<xs:complexType name="EncodingRule">
		<xs:sequence>
			<xs:element name="ruleCitation" type="CI_Citation"/>
			<xs:element name="toolName" type="CharacterString"/>
			<xs:element name="toolVersion" type="CharacterString"/>
	</xs:sequence>
</xs:complexType>

	<xs:complexType name="CodeListExtraction">
		<xs:sequence>
			<xs:element name="name" type="CharacterString"/>
			<xs:element name="source" type="URI"/>
			<xs:element name="timeStamp" type="DateTime"/>
		<xs:element name="format" type="CodeListValueFormat"/>
	</xs:sequence>
</xs:complexType>

<xs:simpleType name="CodeListValueFormat">
	<xs:restriction base="CharacterString">
		<xs:enumeration value="code"/>
		<xs:enumeration value="value"/>
		<xs:enumeration value="mix"/>
	</xs:restriction>
</xs:simpleType>
```

پ- ۵-۴-۴ به روزرسانی

سه روش به‌روزرسانی در شکل پ-۲۳ تعریف شده است.



شکل پ- ۲۳- روش‌های به‌روزرسانی

```
<xs:complexType name="Add">
	<xs:sequence>
		<xs:group ref="Object"/>
	</xs:sequence>
</xs:complexType>

<xs:complexType name="Modify">
	<xs:sequence>
		<xs:group ref="Object"/>
	</xs:sequence>
</xs:complexType>

<xs:complexType name="Delete">
	<xs:attribute name="uuidref" type="CharacterString" use="required"/>
</xs:complexType>

<xs:group name="UpdatePrimitive">
	<xs:choice>
		<xs:element name="add" type="Add"/>
		<xs:element name="modify" type="Modify"/>
		<xs:element name="delete" type="Delete"/>
	</xs:choice>
</xs:group>

<xs:complexType name="Update">
	<xs:choice maxOccurs="unbounded">
		<xs:group ref="UpdatePrimitive"/>
	</xs:choice>
</xs:complexType>
```

پ- ۵-۵ انواع هسته

پ-۵-۵-۱ شناسایی شی

همه انواع شی باید از نوع IM_Object که به کلاس IM_Object مربوط بوده و در مدل نمونه تعریف شده است، گرفته شوند. این نوع دو صفت را تعریف می‌کند که برای شناسایی شی مورد استفاده قرار می‌گیرد. صفت «id» در محدوده سند XML باید منحصربفرد باشد و از قواعد معنایی نوع داده«ID» XML پیروی کند. صفت «uuid» ممکن است برای ایجاد یک شناسه منحصربفرد درون یک مجموعه کلی تعریف شده در یک دامنه کاربردی استفاده شود.

```
<xs:attributeGroup name="IM_ObjectIdentification">
      <xs:attribute name="id" type="xs:ID"/>
      <xs:attribute name="uuid" type="xs:string"/>
</xs:attributeGroup>

<complexType name=”IM_Object”>
      <attributeGroup ref="IM_ObjectIdentification"/>
</complexType>
```

پ- ۵-۵-۲ مرجع شی

نوع IM_ObjectReference فهرستی از سه صفت را تعریف می‌کند که برای ارجاع‌دهی به یک شی باید استفاده شود. صفت «idref» باید برای ارجاع به یک شی ارائه شده توسط یک جز XML در همان مجموعه داده، استفاده شود. صفت «uuidref» باید برای ارجاع به یک شی در مجموعه کلی از یک دامنه‌ی کاربردی استفاده شود. یک سرور نام ممکن است برای دربرداشتن مقادیر «uuid» استفاده شود. صفت «uuref» باید برای ارجاع به اشیایی که در سایر مجموعه داده واقع شده، و با استفاده از قاعده دستوری شناسه منبع یکسان (URI[1]) که توسط منبع [۵] کتاب‌نامه تعریف شده، استفاده گردد.

یک نمونه از نوع IM_ObjectReference صرفا باید دارای مقادیری برای نمایش یکی از سه صفت حاضر باشد.

```
<xs:attributeGroup name="IM_ObjectReference">
            <xs:attribute name="idref" type="xs:IDREF"/>
            <xs:attribute name="uuidref" type="xs:string"/>
            <xs:attribute name="uriref" type="URI"/>
      </xs:attributeGroup>

      xs:complexType name="IM_ObjectReference">
            <xs:attributeGroup ref="IM_ObjectReference"/>
   </xs:complexType>
```

1 -Uniform Resource Identifier

مثال ۱: جزی شی e دارای مقادیر زیر برای صفت‌های «id» و «uuid» است:

<e id="i05" uuid="dce:F6A120B3"> … </e>

مثال ۲: جز l می‌تواند با استفاده از صفت «idref»به e اشاره کند:

<l idref="i05"/>

مثال ۳: این جز می‌تواند با استفاده از صفت «uuidref»به e اشاره کند:

<l uuidref="dce:F6A120B3"/>

مثال ۴: یا جز می‌تواند به e با استفاده از یک لینک که ارجاع به یک سند XML خارجی در ترکیب با شناسایی قطعه با به‌کارگیری یک Xpointer بوده، اشاره کند. در اینجا «i05|» کوتاه شده «id(i05)|» است.

<l uriref="http://www.example.org/data.xml|i05"/>

پ-۵-۶ سازوکارهای برچسب‌گذاری زبان

دو سازوکار برچسب‌گذاری زبان پشتیبانی می‌شود.

- اولین سازوکار استفاده از صفت ذخیره شده «xml:lang» است. صفت ذخیره شده «xml:lang» باید به عنوان صفت جزهایی که به برچسب زبانی نیاز دارند، بیان شود. سازوکار «xml:lamg» نشان می‌دهد که صفت و محتوای یک جز باید به یک زبان خاص باشند.
- سازوکار دوم استفاده از سازوکار برچسب‌گذاری زبان استاندارد ISO/IEC 10646:2011 plane 14 است. اگر بیش از یک زبان در یک رشته‌ی نویسه مورد استفاده قرار گیرد، استاندارد ISO/IEC 10646:2011 plane 14 باید به کار گرفته شود.

استفاده از نویسه‌های برچسب زبان به شدت مأیوس‌کننده است. این نویسه‌ها فقط برای استفاده محدودی توسط پروتکل‌های به‌خصوص که برای ایجاد برچسب‌گذاری زبان برای رشته‌های کوتاه ضروری بوده، بدون استفاده از سازوکارهای نشانه‌گذاری پیشرفته، کدگذاری می‌شوند. بیش‌تر کاربرانی که به متن برچسب با شناسایی زبان نیاز دارند باید از سازوکارهای نشانه‌گذاری استانداردی مانند HTML، XML و سایر سازوکار-های دیگر با متن غنی[1] استفاه کنند. درون پایگاه داده، زبان به طور کلی باید به جای برچسب‌های زبانی تعبیه‌شده یا نشانه‌گذاری، توسط فیلدهای مناسب داده بیان شود.

مثال: در این‌جا یک نوع مرکب با نام CharacterStringWithLanguageTag تعریف می‌شود. توجه داشته باشید که تعریف xml: namespace ضروری می‌باشد.

```
<xs:complexType name="CharacterStringWithLanguageTag">
    <xs:simpleContent>
        <xs:extension base="xs:string">
```

1- Rich text

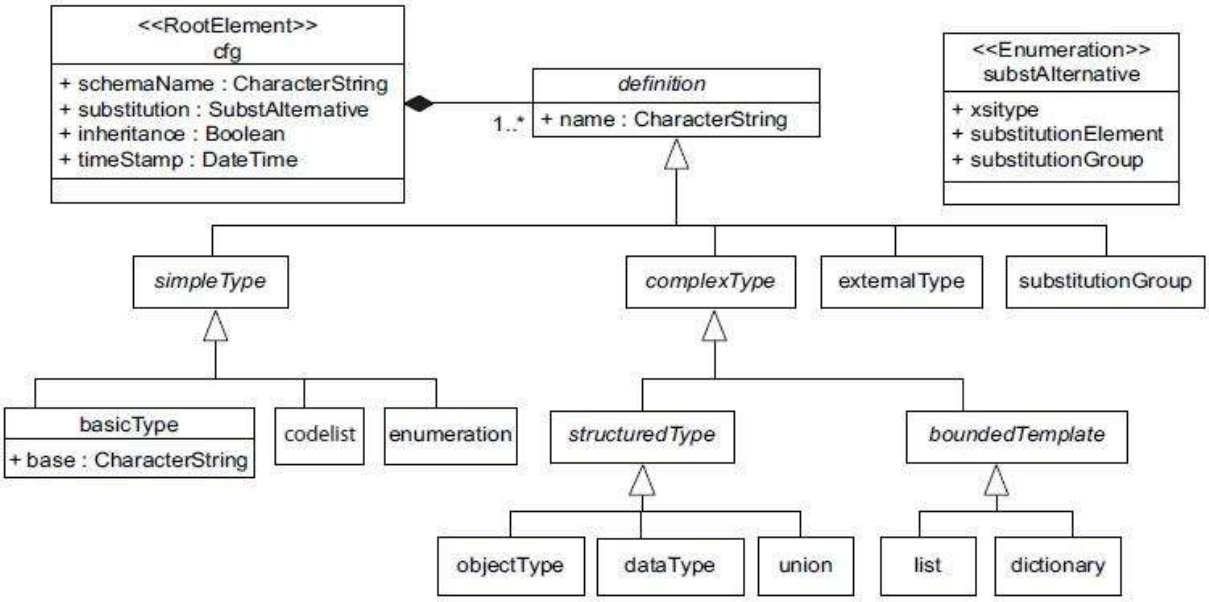
```
            <xs:attribute ref="xml:lang"/>
        </xs:extension>
    </xs:simpleContent>
</xs:complexType>
```

پ-۵-۷ انتخاب‌های کدگذاری

تاکنون تعدادی انتخاب در قواعد تبدیل الگو توضیح داده شده است. این انتخاب‌ها باید مستندسازی شوند و زیربند پ-۵-۷ آن‌ها را فهرست کرده و توضیح می‌دهد که این انتخاب‌ها چگونه باید در یک فایل پیکربندی، ذخیره شوند.

- راهبرد نام‌گذاری. راهبرد نام‌گذاری پیش‌فرض، بدین صورت است که وقتی نام‌های جز در XSD اعلان می‌شوند، نام الگوی کاربردی استفاده می‌شود. قواعد تعریف کلاس، صفت‌ها و نام‌های نقش در UML در استاندارد ISO/TS19103 ارائه شده‌اند. یک راهبرد نام‌گذاری متفاوت با تعریف یک مجموعه از نام‌های جز جایگزین، مجاز می‌باشد.
- اشیا مستقل. پیش‌فرض، بیان همه کلاس‌های بدون کلیشه یا کلیشه <<Type>> به عنوان جزهای شی می‌باشد. اینها ممکن است است با تعریف اشیایی که مستقل یا غیرمستقل بوده، لغو شوند.
- سازوکار وراثت الگوی XML یا نسخه‌برداری. اگر وراثت منفرد مورد استفاده قرار گیرد، بنابراین ممکن است از وراثت الگوی XML استفاده شود. در غیر این صورت باید از سازوکار نسخه‌برداری استفاده شود. سازوکار استفاده‌شده باید تشریح شود.
- سازوکار جایگزینی. فایل پیکربندی باید شکل سازوکار جایگزینی مورد استفاده را نشان دهد.
- استثناء در قواعد. اگر برخی کلاس‌ها به طور متفاوت از قاعده عمومی نگاشت شوند، این موضوع باید تشریح شود.

فایل پیکربندی با یک جز برای هر کلاس در یک سند XML ذخیره می‌شود. فایل پیکربندی باید برای تعیین دقیق کدگذاری کلاس‌های مختلف و ویژگی‌های آن‌ها استفاده شود. فایل پیکربندی هم‌چنین سازوکار تغییر نام‌های جز پیش‌فرض را ارائه می‌دهد. جزهای مهم پیکربندی در شکل پ-۲۴ توضیح داده شده‌اند.



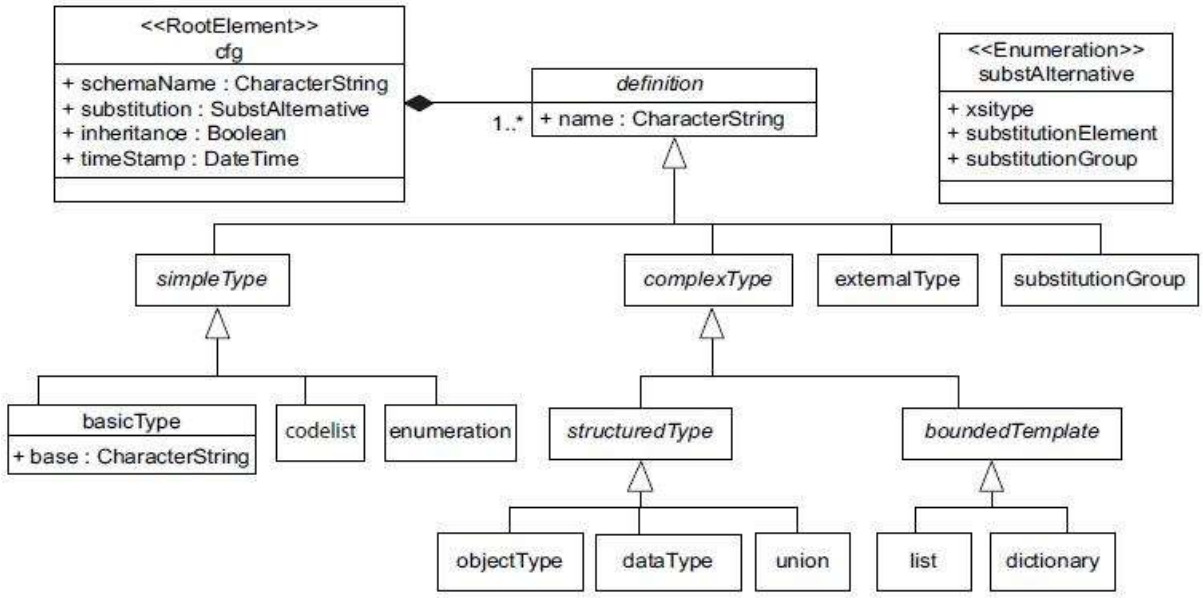
شکل پ- ۲۴- فایل پیکر بندی: جزهای مهم

جزء **cfg** تعداد جزهای تعریف را نشان می‌دهد. این جز دارای صفت بوده که حاوی اطلاعاتی درباره نام الگو، سازوکار جایگزینی مورد استفاده، در صورتی که وراثت به‌کار رفته باشد و تاریخ چاپ فایل (زمانی کـه فایـل تولید شده)، می‌باشد. جزهای تعریف مختلف به صورت زیر است.

- basicType: کلیشه کلاس را به عنوان <<BasicType>> نشان می‌دهد؛
- codelist: کلیشه کلاس را به عنوان <<CodeList>> نشان می‌دهد؛
- enumeration: کلیشه کلاس را به عنوان <<Enumeration>> نشان می‌دهد؛
- objectType: یک نوع شی را نشان می‌دهد (یک کلاس بدون کلیشه یـا بـا کلیشـه <<Type>>). بایـد دارای نامی مشابه کلاس الگوی کاربردی و فهرستی از ویژگی‌های مرتبط با صفت کلاس الگوی کاربردی و وابستگی‌ها باشد. این صفت مستقل مشخص می‌کند که آیا این شی می‌تواند مستقل باشد و در نتیجـه باید دارای اعلان جز جداگانه باشد. صفت برچسب دارای نام XML برای جـز و نـوع مـی‌باشـد و مقـدار پیش‌فرض، نام خود کلاس است؛
- dataType: کلیشه‌ی یک کلاس را به عنوان <<DataType>> نشان می‌دهد و بایـد دارای نـامی مشـابه کلاس الگوی کاربردی و فهرستی از ویژگی‌ها باشد؛
- union: یک کلیشه کلاس را به عنوان <<Union>> نشان می‌دهد و باید دارای نامی مشابه کلاس الگوی کاربردی و فهرستی از ویژگی‌ها باشد؛
- list: نوع قالب محدوده‌ای از مجموعه، دنباله، بسته یا نوع CircularSequence را نشان می‌دهد؛

- dictionary: نوع قالب محدوده‌ای ازنوعDictionary را نشان می‌دهد؛
- externalType: یک کلیشه کلاس را به عنوان <<ExternalType>> نشان می‌دهد؛
- substitutionGroup: یک گروه جایگزین را نشان می‌دهد. یک ابرنوع با زیرنوع‌های اصلی قابل اثبات است.

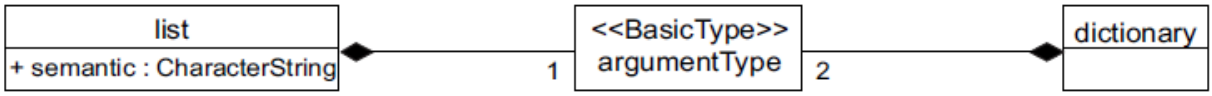
نوع ساختاریافته، یک ابرنوع انتزاعی از objectType، dataType و union است و در شکل پ-۲۵ توضیح داده شده است.



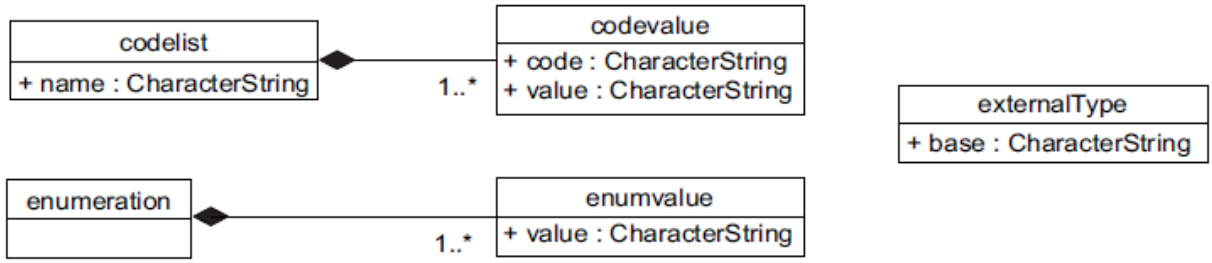
شکل پ- ۲۵- فایل پیکربندی: انواع ساختار یافته

جز ویژگی، یک صفت یا وابستگی یک کلاس را نشان می‌دهد و باید دارای نامی مشابه صفت مربوط در الگوی کاربردی یا نام نقش وابستگی باشد. صفت چندگانگی باید دارای چندین صفت باشد. صفت rep می‌تواند دارای سه مقدار باشد: «value» به این معنی که یک نمونه باید از دامنه‌ی مقادیر نوع ارائه‌شده در dts گرفته شود؛ «reference» به این معنی که یک نمونه از صفت باید به یک شی که شناسه خود را به عنوان یک مقدار در نظر می‌گیرد اشاره کند؛ و «both» به این معنی که می‌تواند یک مرجع یا یک مقدار باشد. صفت dts باید دارای نام نوعی باشد که به نوع داده هدف در الگوی کاربردی مربوط است. اگر rep نشان‌دهنده «reference» باشد، در نتیجه انواع داده در dts باید یک نوع ساختار یافته به همراه شناسه باشند.

نوع اصلی در شکل پ-۲۶ توضیح داده شده و فهرست‌کدها، شمارش‌ها و انواع خارجی در شکل پ-۲۷ تشریح می‌شوند.



شکل پ- ۲۶- فایل پیکر بندی: انواع قالب محدود



شکل پ- ۲۷- فایل پیکر بندی: فهرست کد، شمارش و نوع خارجی

پ-۵-۸ فایل پیکربندی و XSD-مثال

پ-۵-۸-۱ کلیات

زیربندهای پ-۵-۸-۲ و پ-۵-۸-۲ به ترتیب فایل پیکربندی و فایل XSD را برای نمونه الگوی کاربردی ارائه‌شده در زیربند پ-۳-۳ نشان می‌دهند.

پ-۵-۸-۲ فایل پیکربندی

فایل پیکربندی با توجه به الگوی شکل پ-۲۴

```
<?xml version="1.0" ?>
<!DOCTYPE cfg SYSTEM "cfg.dtd">

<cfg schemaName="Ex" inheritance="false" substitution="substitutionGroup"
timeStamp="2001-05-31T15:13">

<objectType id="i1" name="C0" independent="true" tag="C0">
      <property id="i2" name="a1" multiplicity="0..1" rep="value" dts="Real"
attribute="false" tag="a1"/>
      <property id="i3" name="a2" multiplicity="0..*" rep="value" dts="CharacterString"
attribute="false" tag="a2"/>
      <property id="i4" name="a3" rep="value" dts="Dictionary_Integer_CharacterString_"
attribute="false" tag="a3"/>
      <property id="i5" name="role1" multiplicity="1..*" rep="reference" dts="C1"
attribute="false" tag="role1"/>
</objectType>

<objectType id="i7" name="C1" independent="true" tag="C1">
```

```
        <property id="i8" name="a1" rep="value" dts="Integer" attribute="false" tag="a1"/>
        <property id="i9" name="main" multiplicity="1" rep="reference" dts="C0"
attribute="false" tag="main"/>
</objectType>

<objectType id="i10" name="C2" independent="true" tag="C2">
        <property id="i11" name="no_items" rep="value" dts="Integer" attribute="false"
tag="no_items"/>
        <property id="i12" name="pos" rep="value" dts="Point" attribute="false" tag="pos"/>
        <property id="i13" name="part" multiplicity="0..*" rep="value" dts="C3"
attribute="false" tag="part"/>
</objectType>

<objectType id="i14" name="C3" independent="true" tag="C3">
        <property id="i15" name="a1" rep="value" dts="Integer" attribute="false" tag="a1"/>
        <property id="i16" name="main" multiplicity="1" rep="reference" dts="C0"
attribute="false" tag="main"/>
        <property id="i17" name="a2" rep="value" dts="Date" attribute="false" tag="a2"/>
</objectType>

<objectType id="i18" name="Point" independent="true" tag="Point">
        <property id="i19" name="position" rep="value" dts="DirectPosition" attribute="false"
tag="position"/>
</objectType>

<basicType id="i20" name="DirectPosition" base="CharacterString"/>

<dictionary id="i21" name="Dictionary_Integer_CharacterString_">
        <argumentType name="Integer"/>
        <argumentType name="CharacterString"/>
</dictionary>

<substitutionGroup id="i22" name="C1">
        <type name="C1"/><type name="C3"/>
</substitutionGroup>
</cfg>
```

پ-۵-۸-۳ XSD

ابتدا یک فایل Ex.xsd وجود دارد..

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema version="1.0" xmlns:xs="http://www.w3.org/2001/XMLSchema">

<!–PART 1 – Objects with identity→
        <xs:group name="Object">
                <xs:choice>
                        <xs:element name="C0" type="C0"/>
```

```
            <xs:element name="C1" type="C1"/>
            <xs:element name="C2" type="C2"/>
            <xs:element name="C3" type="C3"/>
            <xs:element name="Point" type="Point"/>
        </xs:choice>
</xs:group>–!--PART–2 - Type declarations-->
<xs:complexType name="C0">
        <xs:sequence>
            <xs:element name="a1" type="Real" minOccurs="0"/>
            <xs:element name="a2" type="CharacterString" minOccurs="0"
        maxOccurs="unbounded"/>
            <xs:element name="a3" type="Dictionary_Integer_CharacterString_"/>
            <xs:element name="role1" type="ref_C1" maxOccurs="unbounded"/>
            <xs:element name="theC2" minOccurs="0" maxOccurs="unbounded">
                <xs:complexType>
                    <xs:sequence minOccurs="0">
                        <xs:element name="C2" type="C2"/>
                    </xs:sequence>
                    <xs:attributeGroup ref="IM_ObjectReference"/>
                </xs:complexType>
            </xs:element>
        </xs:sequence>
        <xs:attributeGroup ref="IM_ObjectIdentification"/>
</xs:complexType>

<xs:complexType name="C1">
        <xs:sequence>
            <xs:element name="a1" type="Integer"/>
            <xs:element name="main" type="ref_C0"/>
        </xs:sequence>
        <xs:attributeGroup ref="IM_ObjectIdentification"/>
</xs:complexType>

xs:complexType name="C2">
        <xs:sequence>
            <xs:element name="no_items" type="Integer"/>
            <xs:element name="pos" type="Point"/>
            <xs:element name="part" type="C3" minOccurs="0"
        maxOccurs="unbounded"/>
        </xs:sequence>
        <xs:attributeGroup ref="IM_ObjectIdentification"/>
</xs:complexType>

<xs:complexType name="C3">
        <xs:sequence>
            <xs:element name="a1" type="Integer"/>
            <xs:element name="main" type="ref_C0"/>
            <xs:element name="a2" type="Date"/>
```

```
        </xs:sequence>
        <xs:attributeGroup ref="IM_ObjectIdentification"/>
    </xs:complexType>

    <xs:complexType name="Point">
        <xs:sequence>
            <xs:element name="position" type="DirectPosition"/>
        </xs:sequence>
        <xs:attributeGroup ref="IM_ObjectIdentification"/>
    </xs:complexType>

    <xs:simpleType name="DirectPosition">
        <xs:restriction base="CharacterString"/>
    </xs:simpleType>
    <xs:complexType name="Dictionary_Integer_CharacterString_">
        <xs:sequence maxOccurs="unbounded">
            <xs:element name="Integer" type="Integer"/>
            <xs:element name="CharacterString" type="CharacterString"/>
        </xs:sequence>
    </xs:complexType>

    <xs:group name="C1">
        <xs:choice>
            <xs:element name="C1" type="C1"/>
            <xs:element name="C3" type="C3"/>
        </xs:choice>
    </xs:group>–!--PART–3 - Object Reference types-->
    <xs:complexType name="ref_C1">
        <xs:attributeGroup ref="IM_ObjectReference"/>
    </xs:complexType>
    <xs:complexType name="ref_C0">
        <xs:attributeGroup ref="IM_ObjectReference"/>
    </xs:complexType>
</xs:schema>
```

سپس یک الگوی کاربردی به نام schema: Ex_AS.xsd وجود دارد که تمام فایل‌ها را به یکدیگر پیوند می‌دهد.

```
<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema
    targetNamespace="Ex"
    xmlns="Ex"
    xmlns:xsd="http://www.w3.org/2001/XMLSchema"
    version="1.0">

    <xsd:include schemaLocation="19103.xsd"/>
    <xsd:include schemaLocation="19115.xsd"/>
    <xsd:include schemaLocation="Ex.xsd"/>
    <xsd:include schemaLocation="19118.xsd"/>
</xsd:schema>
```

پ-۶ قواعد تبدیل نمونه

پ-۶-۱ ساختار سند XML

قواعد تبدیل نمونه در پ-۶ ارائه شده‌اند. این قواعد توضیح می‌دهند که اشیا چگونه در مدل نمونه به جزهای XML مرتبط تبدیل می‌شوند. نتیجه یک تبدیل نمونه یک سند XML است که باید یک سند معتبر XML باشد. این فایل باید حاوی موارد زیر باشد:

- دستورالعمل پردازش نسخهXML و اعلان کدگذاری اختیاری؛

مثال ۱: <?xml version="1.0" encoding="UCS-2" ?>

- جز ریشه با ارجاع به یک فضای نام که به یک فایل XSD مرتبط است؛
- جزهایی که به اعلان‌هایی در فایل XSD مرتبط هستند.

ایده کلی این است که اشیا در مدل نمونه، مطابق اعلان‌های فایل XSD، به جزهایی در فایل XML تبدیل شوند.

مثال ۲: سند XML به طور معمول دارای ساختار زیر است. توجه شود که فایل Ex.xsd فضای نام هدف را تعریف نمی‌کند.

```
<?xml version="1.0"?>
<GI version="1.0"
        timeStamp="2001-06-07T23:59"
        xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
        xsi:noNamespaceSchemaLocation="Ex.xsd">
    <exchangeMetadata> ... </exchangeMetadata>
    <dataset>
        <objectElement id="o1"> … </objectElement>
            ...
        ...
    </dataset>
    <update>
        <add> … </add>
        <modify> … </modify>
        <delete uuid="dcd:FD3243"/>
    </update>
</GI>
```

پ-۶-۲ شی

یک شی در مدل نمونه باید براساس کلاس و اعلان جز مرتبط به یک جز شی تبدیل شود. محتوای جز باید برابر جزهای محتوای آن باشد. یک شی مستقل باید دارای مجموعه صفت id خود باشد.

رویکرد کلی، کدگذاری هر شی به عنوان یک جز و اجازه به کدگذاری ویژگی‌های شی به عنوان جز محتوای شی می‌باشد. در اصل، جز باید دارای نامی مشابه نام کلاس و جز محتوا نیز دارای نامی مشابه نام ویژگی‌های خود باشند.

پ- ۶-۳ کدگذاری نویسه

فایل‌های XSD و اسناد XML باید از الگوهای کدگذاری- نویسه که توسط این استاندارد تعیین می‌شود پیروی کنند. XML تمام الگوهای مجاز کدگذاری- نویسه «UTF-8»، «UTF-16»، «ISO/IEC-10646-UCS-2»و «ISO/IEC-10646-UCS-4» را پشتیبانی می‌کند. کدگذاری نویسه باید در اولین بیان فرایند فایل XML تعریف شود. اگر هیچ‌گونه کدگذاری تعیین نشود، پیش‌فرض مطابق XML، «UTF-8» خواهد بود.

مثال:

```
<?xml version="1.0" encoding="UTF-8" ?>
```

پ-۶-۴ کدگذاری XML-مثال

در ادامه مثالی از یک کدگذاری XML تعریف شده توسط الگوی کاربردی و مدل نمونه که در زیربند پ-۳-۳ و فایل پیکربندی و فایل XSD که در زیربند پ-۵-۸ آورده شده، ارائه می‌شود. فایل XSD مطابق با الگوی کاربردی یک فضای نام هدف به نام «Ex» را تعریف می‌کند.

```
<?xml version="1.0" encoding="UTF-8"?>
<ex:GI
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xmlns:ex="Ex"
    xsi:schemaLocation="Ex Ex_AS.xsd"
    timeStamp="1999-05-31T13:20-05:00"
    version="1.0">
<dataset>
    <C0 id="o1">
        <a1>2.14</a1>
        <a2>A</a2><a2>list</a2><a2>of</a2><a2>strings</a2>
        <a3><Integer>1</Integer><CharacterString>one</CharacterString>
            <Integer>2</Integer><CharacterString>two</CharacterString>
            <Integer>3</Integer><CharacterString>three</CharacterString>
        </a3>
        <role1 idref="o8"/><role1 idref="o5"/>
    </C0>
    <C0 id="o2">
        <a2>a</a2><a2>b</a2><a2>c</a2>
        <a3>
            <Integer>1024</Integer><CharacterString>1KB</CharacterString>
            <Integer>2048</Integer><CharacterString>2KB</CharacterString>
```

```
        </a3>
        <role1 idref="o3"/><role1 idref="o6"/><role1 idref="o7"/>
    </C0>
    <C1 id="o3">
        <a1>255</a1>
    <main idref="o2"/>
    </C1>
    <C1 id="o8">
        <a1>255</a1>
        <main idref="o1"/>
    </C1>
    <C2 id="o4">
        <no_items>1</no_items>
        <pos id="p">
            <position>235553 3553234 32</position>
        </pos>
        <part id="o7">
            <a1>20</a1>
            <main idref="o2"/>
            <a2>2000-02-25</a2>
        </part>
    </C2>
    <C3 id="o5">
        <a1>20</a1>
        <main idref="o1"/>
        <a2>2000-02-21</a2>
    </C3>
    <C3 id="o6">
        <a1>30</a1>
        <main idref="o2"/>
        <a2>2000-12-31</a2>
    </C3>
</dataset>
</ex:GI>
```

پ-۷ مجموعه آزمون انتزاعی

پیاده‌سازی کدگذاری مطابق با قواعد کدگذاری در این پیوست نشان می‌دهد که مجموعه‌های آزمون انتزاعی در زیربند پ-۸ محقق شده و هم‌چنین مجموعه‌های آزمون انتزاعی در پیوست **ب** تعریف شده‌اند.

یک قاعده کدگذاری قابلیت‌های سرویس کدگذاری را توضیح می‌دهد.

دو نوع سرویس کدگذاری وجود دارد. نوع ویژه، یک سرویس کدگذاری است که از یک یا چند الگوی کاربردی به‌خصوص پشتیبانی می‌کند. سرویس کدگذاری عمومی از هر الگوی کاربردی پشتیبانی می‌کند. سرویس کدگذاری باید مطابق با انطباق سطح ۱ بوده و می‌تواند مطابق با انطباق سطح ۲ باشد.

انطباق سطح ۱ به طور کامل در پیوست **ب** توضیح داده می‌شود.

یک سرویس کدگذاری که انطباق سطح ۱ را برآورده می‌کند در پیوست **ب** توضیح داده شده، که دارای یک قاعده کدگذاری بوده که تمام الزامات زیربند پ-۸ را برآورده می‌کند. یک سرویس کدگذاری که انطباق سطح ۲ را برآورده می‌کند دارای یک قاعده کدگذاری بوده که به طور مستقیم بر مبنای قاعده کدگذاری برمبنای XML است و در این پیوست توضیح داده می‌شود. برای برآورده کردن انطباق سطح ۲، لازم است که سرویس‌های کدگذاری الزامات توضیح‌داده شده در پیوست **ب** و هم‌چنین زیربند پ-۸ را محقق کنند.

پ-۸ انطباق سطح ۲

پ-۸-۱ الگوی کاربردی

الف- هدف آزمون: تأیید اینکه سرویس کدگذاری می‌تواند توسط یک یا چند الگوی کاربردی که در UML نوشته شده پیکربندی شود و می‌تواند براساس این پیوست الگوهای XML معتبر را بنویسد.

ب- روش آزمون: شناسایی الگوی کاربردی که باید آزمون شود و بازرسی سرویس کدگذاری برای این که معلوم شود که آیا این الگوی کاربردی را پشتیبانی می‌کند یا نه. بازرسی الگوی XML تولید شده توسط سرویس کدگذاری و بررسی اینکه آیا از قواعد تبدیل پیروی می‌کند یا نه و بتواند الگوی XML معتبر را تولید نماید.

پ- مرجع: بندهای ۶، ۷، ۸، ۹ و پیوست **پ**.

ت- نوع آزمون: آزمون قابلیت.

پ-۸-۲ ساختار داده

الف- هدف آزمون: تأیید این که سرویس کدگذاری می‌تواند اسناد XML معتبر را مطابق این استاندارد بخواند و بنویسد.

ب- روش آزمون: مواردی از قبیل داده کاربردی و چگونگی نگاشت آن به یک ساختار داده خاص- الگو- کاربرد، مدل نمونه و اینکه آیا قواعد تبدیل به درستی اعمال شده باشند، را بازرسی می‌کند. این کار با بازرسی بخشی از سند XML و استفاده از یک پردازشگر XML عمومی برای خواندن سند XML تولیدشده، انجام می‌گیرد.

پ- مرجع: پیوست **پ**.

ت- نوع آزمون: آزمون اصلی.

پ-۸-۳ داده‌های آزمون

الف- هدف آزمون: تأیید این که سرویس کدگذاری می‌تواند مجموعه داده آزمون همراه با الگوی کاربردی را بخواند و بنویسد. یک الگوی کاربردی باید دارای مجموعه داده انتزاعی و داده‌های آزمون مرتبط باشد که شامل یک سند XML صحیح بوده که بر مبنای مجموعه داده انتزاعی تولید شده، باشد.

ب- روش آزمون: مدل نمونه‌ای را با مجموعه داده انتزاعی جمع کرده و یک سند XML تولید می‌شود. این سند را مقایسه کرده تا مشخص شود که آیا از نظر منطقی معادل داده آزمون الگوی کاربردی می‌باشد. داده‌های آزمون را خوانده و آن‌ها را با مدل نمونه حاصله مقایسه می‌کنند. در صورت وجود ناسازگاری-هایی بین دو نوع داده، آزمون رد می‌شود. باید کنترل شود که داده نویسه در تبدیل‌ها از بین نروند.

پ- مرجع: پیوست **پ**.

ت- نوع آزمون: آزمون ساختگی[1].

1- Falsification test

کتاب‌نامه

[۱] استاندارد ملی ایران شماره ۱-۶۴۱۷: سال ۱۳۸۸، فناوری اطلاعات- واژه نامه -قسمت۱- واژه‌های پایه

[۲] استاندارد ملی ایران شماره ۸۲۱۱: سال ۱۳۸۹، فناوری اطلاعات - ویژگی یک فایل توصیفی داده برای تبادل اطلاعات

[۳] استاندارد ملی ایران شماره ۱۶۳۶۴: سال ۱۳۹۱، اطلاعات مکانی -انطباق و آزمون

[۴] استاندارد ملی ایران شماره ۱۶۳۶۵: سال ۱۳۹۱، اطلاعات مکانی –پروفایل‌ها

[5] ISO/IEC 2022:1994, Information technology — Character code structure and extension techniques

[6] ISO/IEC 8859 (all parts), Information technology — 8-bit single-byte coded graphic character sets

یادآوری- مجموعه استاندارد ملی ایران شماره ۸۸۵۹، فناوری اطلاعات- مجموعه‌نویسه‌های شکلی کدشده تک‌بایتی ۸ بیتی با استفاده از برخی قسمت‌های مجموعه استاندارد ISO/IEC 8859 تدوین شده است.

[7] ISO 8879:1986, Information processing — Text and office systems — Standard Generalized Markup Language (SGML)

[8] ISO/IEC 9834-8, Information technology — Open Systems Interconnection — Procedures for the operation of OSI Registration Authorities: Generation and registration of Universally Unique Identifiers (UUIDs) and their use as ASN.1 Object Identifier components

یادآوری- استاندارد ملی ایران شماره ۸-۹۸۳۴: سال ۱۳۹۰، فناوری اطلاعات -اتصال متقابل سامانه‌های باز- رویه‌های عملیات مراجع ثبت اتصال متقابل سامانه‌های باز (OSI) تولید و ثبت شناسانه‌های منحصر به فرد بین‌المللی (UUIDS) و استفاده ی آن‌ها به عنوان مولفه‌های شناسانه شی نشانه گذاری قاعده نحوی انتزاعی یک(ASN.1) با استفاده از استاندارد ISO/IEC 9834-8 تدوین شده است.

[9] ISO 10303-21, Industrial automation systems and integration — Product data representation and exchange — Part 21: Implementation methods: Clear text encoding of the exchange structure

[10] ISO/IEC 11179-3:2003, Information technology — Metadata registries (MDR) — Part 3: Registry metamodel and basic attributes

[11] ISO/IEC 11578, Information technology — Open Systems Interconnection — Remote Procedure Call (RPC)

یادآوری- استاندارد ملی ایران شماره ۱۱۵۷۸: سال ۱۳۹۱،فناوری اطلاعات -اتصال متقابل سامانه‌های بازبرخوانی رویه‌ی راه دور (RPC) با استفاده از استاندارد ISO/IEC 11578 تدوین شده است.

[12] ISO 19101:2002, Geographic information — Reference model

[13] ISO/TS 19104:2008, Geographic information — Terminology

[14] ISO 19107:2003, Geographic information — Spatial schema

[15] ISO 19108:2002, Geographic information — Temporal schema

[16] ISO 19110:2005, Geographic information — Methodology for feature cataloguing

[17] ISO 19111:2007, Geographic information — Spatial referencing by coordinates

[18] ISO 19112:2003, Geographic information — Spatial referencing by geographic identifiers

[19] ISO 19113:2002, Geographic information — Quality principles

[20] ISO 19114:2003, Geographic information — Quality evaluation procedures

[21] ISO 19115:2003, Geographic information — Metadata

[22] ISO 19116:2004, Geographic information — Positioning services

[23] ISO 19117:2005, Geographic information — Portrayal

[24] ISO 19119:2005, Geographic information — Services

[25] ISO/TR 19120:2001, Geographic information — Functional standards

[26] ISO 19135:2005, Geographic information — Procedures for item registration

[27] ISO 19136:2007, Geographic information — Geography Markup Language (GML)

[28] ISO/TS 19139:2007, Geographic information — Metadata — XML schema implementation

[29] ISO/IEC 19501, Information technology — Open Distributed Processing — Unified Modeling Language (UML) Version 1.4.2

[30] Document Object Model (DOM) Level 1 Specification, Version 1.0, W3C Recommendation. October 1998. Available at <http://www.w3.org/TR/REC-DOM-Level-1>

[31] IETF STD 1:1998, Internet Official Protocol Standards

[32] KUHN, M. UTF-8 and Unicode FAQ. Available at <http://www.cl.cam.ac.uk/~mgk25/unicode.html>

[33] Namespaces in XML, W3C Recommendation. Available at <http://www.w3.org/TR/REC-xml-names>

[34] Resource Description Framework (RDF) Model and Syntax Specification, W3C Recommendation. February 1999. Available at <http://www.w3.org/TR/1999/REC-rdf-syntax-19990222/>

[35] Resource Description Framework (RDF) Schema Specification 1.0, W3C Candidate Recommendation. March 2000. Available at <http://www.w3.org/TR/2000/CR-rdf-schema-20000327/>

[36] RFC 2396, Uniform Resource Identifiers (URI): Generic Syntax. Available at http://www.ietf.org/rfc/rfc2396.txt

[37] Scalable Vector Graphics (SVG) 1.1, W3C Working Draft May 2011. Available at <http://www.w3.org/TR/SVG/>

[38] The Open Group, CAE Specification, DCE 1.1: Remote Procedure Call, Document Number C706. Available at http://www.opengroup.org/publications/catalog/c706.htm

[39] www.w3.org/TR/charmod-norm/

[40] XML Linking Language (XLink) Version 1.0, W3C Recommendation. Available at <http://www.w3.org/TR/xlink/>

[41] XML Metadata Interchange (XMI), Version 2.0, Object Management Group. Available at <http://www.omg.org/cgi-bin/doc?formal/2003-05-02>

[42] XML Path Language (XPath), Version 1.0, W3C Recommendation. November 1999. Available at <http://www.w3.org/TR/xpath>

[43] XML Pointer Language (XPointer), W3C Working Draft. Available at <http://www.w3.org/TR/xptr>

[44] XSL Transformations (XSLT), Version 1.0, W3C Recommendation. November 1999. Available at <http://www.w3.org/TR/xslt>