



جمهوری اسلامی ایران
Islamic Republic of Iran

سازمان ملی استاندارد ایران

Iranian National Standardization Organization



استاندارد ملی ایران

۱۱۴۹۳-۲

چاپ اول

۱۳۹۴

INSO
11493-2
1st. Edition
2016

فناوری اطلاعات - واسط‌های کاربر - پیشانه
(کنسول) از دور جهانی -
قسمت ۲: توصیف سوکت واسط کاربر

**Information technology — User interfaces
— Universal remote console — Part 2: User
interface socket description**

ICS:35.240.20

سازمان ملی استاندارد ایران

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

صندوق پستی: ۶۱۳۹-۱۴۱۵۵ تهران- ایران

تلفن: ۵-۸۸۸۷۹۴۶۱

دورنگار: ۸۸۸۸۷۰۸۰ و ۸۸۸۸۷۱۰۳

کرج، شهر صنعتی، میدان استاندارد

صندوق پستی: ۱۶۳-۳۱۵۸۵ کرج- ایران

تلفن: ۸-۳۲۸۰۶۰۳۱ (۰۲۶)

دورنگار: ۳۲۸۰۸۱۱۴ (۰۲۶)

رایانامه: standard@isiri.org.ir

وبگاه: <http://www.isiri.org>

Iranian National Standardization Organization (INSO)

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

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

Tel: + 98 (21) 88879461-5

Fax: + 98 (21) 88887080, 88887103

Standard Square, Karaj, Iran

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

Tel: + 98 (26) 32806031-8

Fax: + 98 (26) 32808114

Email: standard@isiri.org.ir

Website: <http://www.isiri.org>

به نام خدا

آشنایی با سازمان ملی استاندارد ایران

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

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

پیش‌نویس استانداردهایی که مؤسسات و سازمان‌های علاقه‌مند و ذی‌صلاح نیز با رعایت ضوابط تعیین شده تهیه می‌کنند در کمیته ملی طرح، بررسی و در صورت تصویب، به عنوان استاندارد ملی ایران چاپ و منتشر می‌شود. بدین ترتیب، استانداردهایی ملی تلقی می‌شود که بر اساس مقررات استاندارد ملی ایران شماره ۵ تدوین و در کمیته 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)^۱، کمیسیون بین‌المللی الکتروتکنیک (IEC)^۲ و سازمان بین‌المللی اندازه‌شناسی قانونی (OIML)^۳ است و به عنوان تنها رابط^۴ کمیسیون کدکس غذایی (CAC)^۵ در کشور فعالیت می‌کند. در تدوین استانداردهای ملی ایران ضمن توجه به شرایط کلی و نیازمندی‌های خاص کشور، از آخرین پیشرفت‌های علمی، فنی و صنعتی جهان و استانداردهای بین‌المللی بهره‌گیری می‌شود.

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

1- International Organization for Standardization

2- International Electrotechnical Commission

3- International Organization for Legal Metrology (Organisation Internationale de Metrologie Legals)

4- Contact point

5- Codex Alimentarius Commission

کمیسیون فنی تدوین استاندارد

«فناوری اطلاعات - واسط‌های کاربر - پیشانه (کنسول) از دور جهانی - قسمت ۲: توصیف

سوکت واسط کاربر»

رئیس:

ترابی، مهنروش

(فوق لیسانس فناوری اطلاعات - تجارت الکترونیک)

سمت و / یا نمایندگی

کارشناس تجزیه و تحلیل سیستم

شرکت برق منطقه‌ای هرمزگان

دبیر:

مشرف، بهنوش

(فوق لیسانس فناوری اطلاعات - شبکه‌های کامپیوتری)

کارشناس پایگاه داده‌ها

شرکت برق منطقه‌ای هرمزگان

اعضاء: (اسامی به ترتیب حروف الفبا)

ابراهیم نژاد، پوریا

(فوق لیسانس مهندسی برق - مخابرات)

کارشناس شبکه‌های بی سیم

شرکت ایرانسل

احمدی، محمد

(فوق لیسانس مهندسی برق - مخابرات)

کارشناس فیبر نوری

شرکت برق منطقه‌ای هرمزگان

اشرفی، رضا

(فوق لیسانس مهندسی برق - ICT)

مدیر بخش توسعه

شرکت تامین تله کام

فرزانه عماد

(فوق لیسانس مهندسی کامپیوتر - معماری کامپیوتر)

کارشناس سخت افزار

سازمان فناوری اطلاعات شهرداری سیرجان

مشرف، فاطمه

(لیسانس زبان انگلیسی)

کارشناس روابط بین‌الملل

دانشگاه زابل

مهرشاد، بتول

(دانشجوی دکترا مدیریت فناوری اطلاعات)

رییس بخش انفورماتیک

اداره کل استاندارد خراسان جنوبی

فهرست مندرجات

| صفحه | عنوان |
|------|--|
| ب | آشنایی با سازمان ملی استاندارد |
| ج | کمیسیون فنی تدوین استاندارد |
| ۱ | ۱ هدف و دامنه کاربرد |
| ۱ | ۲ انطباق |
| ۲ | ۳ مراجع الزامی |
| ۲ | ۴ اصطلاحات و تعاریف |
| ۲ | ۵ ارتباط با سایر استانداردها |
| ۱۲ | ۶ ساختار توصیف سوکت تکمیلی |
| ۱۵ | ۷ مجموعه‌ها |
| ۱۷ | ۸ ملاحظات امنیتی و حریم خصوصی |
| ۲۹ | ۹ فرمان |
| ۴۰ | ۱۰ اعلان |
| ۴۶ | ۱۱ تعریف نوع |
| ۵۰ | ۱۲ ملاحظات امنیتی |
| ۵۱ | پیوست الف (اطلاعاتی) اسنادی برای توصیف سوکت واسط کاربر |

پیش‌گفتار

استاندارد « فناوری اطلاعات - واسط کاربر - پیشانه (کنسول) از دور جهانی - قسمت ۲: توصیف سوکت واسط کاربر»، که پیش‌نویس آن در کمیسیون‌های مربوط توسط سازمان ملی استاندارد تهیه و تدوین شده و در سیصد و نود و پنجمین اجلاس هیئت کمیته ملی فناوری اطلاعات مورخ ۱۳۹۴/۱۲/۰۳ مورد تصویب قرار گرفته است، اینک به استناد بند یک ماده ۳ قانون اصلاح قوانین و مقررات مؤسسه استاندارد و تحقیقات صنعتی ایران، مصوب بهمن ماه ۱۳۷۱، به عنوان استاندارد ملی ایران منتشر می‌شود.

برای حفظ همگامی و هماهنگی با تحولات و پیشرفت‌های ملی و جهانی در زمینه صنایع، علوم و خدمات، استانداردهای ملی ایران در مواقع لزوم تجدید نظر خواهد شد و هر پیشنهادی که برای اصلاح و تکمیل این استانداردها ارائه شود، هنگام تجدید نظر در کمیسیون فنی مربوط مورد توجه قرار خواهد گرفت. بنابراین، باید همواره از آخرین تجدیدنظر استانداردهای ملی استفاده کرد.

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

ISO/IEC 24752-2:2014, Information technology – User Interfaces – Universal remote console - Part 2:User interface socket description.

فناوری اطلاعات - واسط کاربر - پیشانه (کنسول) از دور جهانی - قسمت ۲:

توصیف سوکت واسط کاربر

۱ هدف و دامنه کاربرد

هدف از تدوین این استاندارد تعیین تسهیلاتی برای کار با محصولات اطلاعاتی و الکترونیکی از طریق واپایش از راه دور، واسط‌های جایگزین و عوامل هوشمند می‌باشد.

یک سوکت واسط کاربر، یک واسط کاربر انتزاعی است که قابلیت و حالت یک افزاره یا خدمت (هدف) را به شیوه‌ای ماشین-تفسیری^۱ توضیح می‌دهد که مستقل از ظاهر و قابلیت‌های ورودی یک افزاره تعامل با کاربر است. این استاندارد زبان مبتنی بر زبان نشانه‌گر توسعه‌پذیر (XML)^۲ برای توصیف یک سوکت واسط کاربر تعریف می‌کند. هدف از سوکت واسط کاربر ارائه اطلاعات مربوط به یک هدف است به طوری که کاربر بتواند حالتش را متوجه شود و آن را به کار بیندازد. که شامل اطلاعات ارائه شده به کاربر، متغیرهایی که می‌توانند توسط کاربر دستکاری شوند، فرمان‌هایی که کاربر می‌تواند فعال کند، و موارد استثنایی^۳ است که در موردشان به کاربر اطلاع داده می‌شود. ویژگی^۴ سوکت واسط کاربر قابل اعمال به ساخت و سازش^۵ واسط‌های کاربر است.

۲ انطباق

یک پوشه XML مطابق با این استاندارد است (مثال: یک توصیف سوکت واسط کاربر) اگر همه الزامات زیر را برآورده کند:

الف) در صورت امکان یک نوع توسعه نامه اینترنتی چند منظوره (MIME)^۶ داشته باشد همانطور که در زیربند ۶-۱ مشخص شده است؛

ب) در قالب UCS کد شده باشد (به زیربند ۶-۱ مراجعه کنید)؛

پ) عنصر ریشه آن، عنصر `<uis:uiSocket>` باشد (uis نشان‌دهنده فضای نام `http://openurc.org/ns/uisocketdesc-2` می‌باشد)، همانطور که در بند ۶ مشخص شده است؛

ت) شامل تمام عناصر و مشخصه‌های^۷ مورد نیاز همراه با ارزش‌هایشان باشد، همانطور که در بند ۶ مشخص شده است؛

ث) اگر شامل عناصر توصیه‌شده یا اختیاری یا مشخصه‌ها همراه با ارزش‌هایشان باشد، این‌ها همانطور که در بند ۶ مشخص شده است، ارائه می‌شوند.

1-Machine-interpretable
2-Extensible Markup Language
3-Exceptions
4-Specification
5-Adaptation
6-Multipurpose Internet Mail Extension
7-Attributes

یادآوری ۱- انطباق محض زبان (یعنی هیچ عنصر یا مشخصه اضافی، مجاز نباشد) نیاز نیست زیرا نسخه‌های آتی این استاندارد ممکن است عناصر، مشخصه‌ها، و ارزش‌های جدیدی اضافه کنند. بنابراین، سازندگان URC تشویق به اجرای URCsشان می‌شوند به طوری که نشانه گذاری ناشناخته بدون خطا نادیده گرفته می‌شود.

یادآوری ۲- سازندگان هدف که می‌خواهند اطلاعات مخصوص تولیدکننده را به توصیف سوکت فراتر از عناصر، مشخصه‌ها، و ارزش‌های مشخص شده در این استاندارد، اضافه کنند این کار را می‌توانند با ارائه (اختصاصی) توصیف منابعی که به ساختار توصیف هدف اشاره می‌کنند، به صورت خارجی انجام دهند. برای جزئیات بیشتر به استاندارد ISO / IEC 24752-5 مراجعه کنید.

۳ مراجع الزامی

مدارک الزامی زیر شامل مقرراتی است که در متن این استاندارد ملی ایران به آن‌ها ارجاع داده شده است. بدین ترتیب آن مقررات جزئی از این استاندارد ملی ایران محسوب می‌شود. در صورتی که به مدرکی با ذکر تاریخ انتشار ارجاع داده شده باشد، اصلاحیه‌ها و تجدیدنظرهای بعدی آن موردنظر این استاندارد ملی ایران نیست. در مورد مدارکی که بدون ذکر تاریخ انتشار به آن‌ها ارجاع داده شده است، همواره آخرین تجدیدنظر و اصلاحیه‌های بعدی آن‌ها مورد نظر است.

استفاده از مراجع زیر برای این استاندارد الزامی است:

3-1 ISO 15836:2009, Information and documentation — The Dublin Core metadata element set

3-2 ISO/IEC 10646:2011, Information technology — Universal Coded Character Set (UCS)

3-3 ISO/IEC 14977:1996, Information technology — Syntactic metalanguage — Extended BNF

3-4 ISO/IEC 24752-1, Information technology — User interfaces — Universal remote console — Part 1: Framework

3-5 ISO/IEC 24752-4, Information technology — User interfaces — Universal remote console — Part 4: Target description

3-6 W3C Recommendation: XML Path Language (XPath) 2-0 (Second Edition), W3C Recommendation 14 December 2010 (Link errors corrected 3 January 2011)

3-7 W3C Recommendation: XQuery 1.0 and XPath 2.0 Functions and Operators (Second Edition), W3C Recommendation 14 December 2010

3-8 W3C Recommendation: XML Schema Part 1: Structures Second Edition, W3C Recommendation 28 October 2004

3-9 W3C Recommendation: XML Schema Part 2: Datatypes Second Edition, W3C Recommendation 28 October 2004

۴ اصطلاحات و تعاریف

در این استاندارد، علاوه بر اصطلاحات و تعاریف به کار برده شده در استاندارد ISO / IEC 24752-1 و استاندارد ملی ایران به شماره 4-11493، اصطلاحات و تعاریف زیر نیز به کار می‌رود.

۱-۴

عنصر زمینه^۱

عنصری که به آن یک وابستگی مربوط می‌شود.

۵ ارتباط با سایر استانداردها

۱-۵ ارتباط با XML

این ویژگی‌ها یک زبان مبتنی بر XML را تعریف می‌کند. نشانه‌گر در XML به حروف کوچک و بزرگ حساس است.

نام‌های برچسب^۱ و نام‌ها و ارزش‌های مشخصه قابل بومی‌سازی نیستند، یعنی آن‌ها برای تمام زبان‌های بین‌المللی یکسان هستند. با این حال، محتوای متن بین برچسب‌ها می‌تواند مخصوص زبان باشد. مثل تمام زبان‌های مبتنی بر XML، نویسه‌های فاصله که بلافاصله اطراف برچسب‌ها را احاطه می‌کنند، قابل توجه نیستند.

این ویژگی باعث می‌شود زمینه فضای نام‌های^۲ XML را قابل استفاده می‌کند به منظور اینکه وارد کردن نام‌های مشخصه و عنصر تعریف شده در جاهای دیگر را ممکن سازد.

همه نام‌های مشخصه و عنصر استفاده شده در این استاندارد بدون پیشوند فضای نام، توسط مجموعه استانداردهای ISIRI 11493 تعریف می‌شوند و قسمتی از فضای نام، با ارجاع <http://openurc.org/ns/> URI استاندارد 2-uisocketdesc هستند. اگر به عنوان فضای نام پیش فرض تعریف نشده باشند، شناسانه^۳ فضای نام: `uis` بهتر است استفاده شود.

در این استاندارد، پیشوندهای فضای نام و شناسانه فضای نام متناظر زیر، برای ارجاع فضای نام‌های خارجی استفاده می‌شود:

- `dc`: فضای نام مجموعه عنصر فراداده هسته دوبلین (<http://purl.org/dc/elements/1.1/>) (مجموعه عنصر تعریف شده توسط استاندارد ISO 15836)؛

- `dcterms`: فضای نام اصطلاحات متاداده DCMI (<http://purl.org/dc/terms>)؛

- `xsd`: فضای نام طرح‌واره XML (<http://www.w3.org/2001/XMLSchema>)؛

- `xsi` : فضای نام نمونه طرح‌واره XML (<http://www.w3.org/2001/XMLSchema-instance>).

برای تعریف طرح‌واره^۴ XML برای واسط کاربر توصیف سوکت، به پیوست الف مراجعه کنید.

۲-۵ عبارات XPath

۱-۲-۵ کلیات

این ویژگی، زبان مسیر XML (XPath)^۵ نسخه 2.0 را برای پرداختن به عناصر درون سوکت استفاده می‌کند. به طور خاص، XPath در توصیف وابستگی بین عناصر سوکت استفاده می‌شود.

قواعد نحوی XPath 2.0 بدون سازگاری با XPath 1.0 استفاده می‌شود.

1-Tag names
2-Namespace
3-Identifier
4-Schema
5-XML Path Language

۵-۲-۲ استفاده از قواعد نحوی و معنایی XPath 2.0

مجموعه استاندارد ISIRI 11493 قواعد نحوی و معنایی XPath 2.0، را با موارد اضافه و استثنای زیر به کار می‌برد:

الف) عبارات XPath باید در UCS کد شوند.

ب) زمینه عبارت ایستا (به بند ۲-۱-۱ در XPath 2.0 مراجعه کنید) باید با مولفه‌های زیر مقداردهی اولیه شده باشد:

۱) "XPath 1.0 compatibility mode" باید نادرست باشد.

۲) "statically known namespaces" اعلام‌های فضای نامی هستند که در حوزه عنصر XML که شامل عبارت XPath است، می‌باشند.

۳) "default element/type namespace" باید فضای نام تهی^۱ باشد (که اشاره به انواعی دارد که در توصیف سوکت تعریف شده است، به بند ۱۱ مراجعه کنید).

۴) "default function namespace" باید فضای نام کارکرد استاندارد مربوط به XPath 2.0 باشد: <http://www.w3.org/2005/xpath-functions>

۵) "in-scope schema definitions" باید فقط شامل "in-scope schema types" با محتوای زیر باشد: همه انواع فضای نام <http://www.w3.org/2001/XMLSchema>، همانطور که در زیربند ۲-۵-۱ از XPath 2.0 آمده است، و انواع محلی تعریف شده در قسمت `<xsd:schema>` توصیف سوکت.

یادآوری - XPath 2.0 انواع از پیش تعریف شده زیر را به انواع از پیش تعریف شده تعریف طرح‌واره XML قسمت ۲، اضافه می‌کند: `xsd:untypedAtomic`، `xsd:anyAtomicType`، `xsd:dayTimeDuration`، `xsd:yearMonthDuration`.

۶) "in-scope variables" باید خالی باشد.

۷) "function signatures" باید کارکردهایی از فضای نام <http://www.w3.org/2005/xpath-functions> باشند، چنانچه در کارکردها و عملگرهای XQuery 1.0 و XPath 2.0 تعریف شده است، با موارد استثنا که در زیربند ۲-۴-۵ مشخص شده است؛ کارکردهای سازنده برای همه انواع تفکیک ناپذیر در "in-scope schema definitions" و کارکردهای اضافی که در زیربند ۲-۵-۵ تعریف شده است.

یادآوری - مولفه‌های زیر مربوط به زمینه عبارت ایستا XPath 2.0 در این استاندارد استفاده نمی‌شوند: "context item static type"، "statically known collations"، "default collation"، "base URI"، "statically known documents"، "statically known collections"، "statically known default collection type".

پ) زمینه عبارت پویا (به زیربند ۲-۱-۲ در XPath 2.0 مراجعه کنید) باید با مولفه‌های زیر مقداردهی اولیه شود:

۱) "context item" باید مجموعه یا عنصر سوکتی باشد که عبارت XPath به عنوان وابستگی^۲ برایشان مشخص شده است.

۲) "variable values" باید خالی باشد.

1-Null

2-Dependency

۳) “function implementations” باید شامل اجراهای کارکردهای فضای نام <http://www.w3.org/2005/xpath-functions> باشد، همانطور که در کارکردها و عملگرهای XPath 2.0 و XQuery 1.0؛ کارکردهای سازنده برای همه انواع تفکیک ناپذیر در “in-scope schema defenitions”؛ و کارکردهای اضافی بر اساس تعریف زیربند ۵-۲-۵، تعریف شده است.

۴) “current dateTime” باید زمان جاری با منطقه زمانی محلی^۱ URC باشد، که نشان‌دهنده یک ارزش از نوع `xsd:dateTime` می‌باشد.

۵) “implicit timezone” باید منطقه زمانی محلی URC باشد.

یادآوری- مولفه‌های زمینه عبارت پویا XPath 2.0 ای که در ادامه آمده است در این استاندارد استفاده نمی‌شوند: “context item”، “context position”، “context size”، “Available documents”، “Available collections”، “Default collection”.

ت) هیچ مدل داده‌ای (نمونه^۲ XDM) وجود ندارد. عبارات و کارکردهایی که به یک نمونه مدل داده ارجاع می‌دهند نباید در توصیف‌های سوکت استفاده شوند. عبارت عنصر زمینه (به زیربند ۳-۱-۴ در XPath 2.0 مراجعه کنید) نباید استفاده شود. عملیات گره مانند مقایسه گره (به زیربند ۳-۵-۳ در XPath 2.0 مراجعه کنید)، و عملگرهای اجتماع، اشتراک و استثنا (به بند ۳-۳-۳ در XPath 2.0 مراجعه کنید) نباید استفاده شود.

ث) ارزیابی عبارات منطقی (AND / OR) باید حتماً از چپ به راست باشد، و عملوند سمت راست نباید ارزیابی شود اگر نتیجه قبلاً توسط عملوند سمت چپ تعیین شده باشد. مثال: در عبارت “A and B”، B نباید ارزیابی شود اگر A نادرست باشد؛ و در مورد “A or B”، B نباید ارزیابی شود اگر A درست باشد. علاوه بر این، عملیات بولی باید ارزش “undefined” را در نظر بگیرند (به زیربند ۳-۲-۵ مراجعه کنید).

ج) اجرای XPath 2.0 باید بر اساس XML 1.0 و فضاهای نامی در XML باشد.

چ) اجرای XPath 2.0 مجاز است محور فضای نام را پشتیبانی کند.

۳-۲-۵ ارزش تعریف نشده

مجموعه استانداردهای ISIRI 11493 ارزش “undefined” را به عنوان ارزش ویژه‌ای برای همه انواع XPath 2.0 (به زیربند ۲-۲-۵ مراجعه کنید) و انواع تعریف شده به صورت محلی (به بند ۱۱ مراجعه کنید) می‌افزاید.

اگر هر بخشی از یک عبارت XPath تعریف نشده باشد، کل عبارت باید تعریف نشده باشد. اگر بر اساس منطق ارزیابی، نتیجه یک عبارت، بدون ارزیابی هر بخش تعریف نشده‌ای از آن تعیین شود، این قانون نباید اعمال شود.

مثال: عبارت زیر هرگز یک نتیجه تعریف نشده نخواهد داشت. در صورتی که عنصر با شناسانه ‘myvar’ موجود باشد عبارت صحیح است و ارزش ۴ می‌گیرد، در غیر این صورت عبارت نادرست است.

```
uis:hasDefinedValue('myvar') and uis:value('myvar') eq 4
```

1-Local timezone
2-Instance

یادآوری- اجرا مجاز است تا هر زمان که اثر توصیف شده، تضمین شود ادامه یابد برای مثال یک خطای استثنا می تواند به صورت داخلی اجرا شود تا سیگنالی دهد مبنی بر این که عبارت XPath، "undefined" است.

۴-۲-۵ کارکردهای XPath

- کارکردهای XPath زیر مجاز است مورد استفاده قرار گیرد:
- کارکردهای فضای نام <http://www.w3-org/2005/xpath-functions>، همانطور که در کارکردها و عملگرهای XPath 2.0 و XQuery 1.0 تعریف شده است.
- کارکردهای سازنده برای همه انواع تفکیک ناپذیر در "in-scope schema definitions" با استثنای زیر:
- کارکرد string() تنها باید با یک آرگومان استفاده شود.
- کارکرد resolve-uri() نباید استفاده شود.
- کارکردهای مربوط به QName (قسمت ۱۱ از کارکردها و عملگرهای XPath 2.0 و XQuery 1.0)، عملگرها در NOTATION (قسمت ۱۳) و کارکردها و عملگرها در گرهها (قسمت ۱۴) نباید استفاده شود.
- کارکردهای زمینه ای زیر (قسمت ۱۳ از کارکردها و عملگرهای XPath 2.0 و XQuery 1.0) نباید استفاده شود: position, last, default-collation, tatic-base-uri.
- قوانین مخصوص XPath 2.0 برای تبدیل ضمنی بین انواع، اعمال می شود.

۵-۲-۵ کارکردهای اضافی

۱-۵-۲-۵ کلیات

مجموعه استانداردهای ISIRI 11493 کارکردهای اضافی زیر که مجاز است در عبارت توصیف سوکت استفاده شود را تعریف می کند.

یادآوری- این کارکردها در فضای نام "http://openurc.org/ns/uisocketdesc-2" تعریف شده اند. یک پیشوند برای این فضای نام (برای مثال "uis") نیاز است تا روی هر یک از عناصر XML که حاوی عبارت XPath هستند، اعلام شود. توجه داشته باشید که پیشوند "http://openurc.org/ns/uisocketdesc-2" همیشه باید برای این کارکردها استفاده شود زیرا فضای نام کارکرد پیش فرض، <http://www.w3-org/2005/xpath-functions> است (فضای نام کارکرد XPath). استفاده از مشخصه 'xmlns' برای اعلام یک فضای نام پیش فرض XML، فضای نام کارکرد پیش فرض را تغییر نمی دهد.

۲-۵-۲-۵ xsd:anyType uis:value(string path)

این کارکردی است که آرگومانش را از مسیر یک متغیر، فرمان، اعلان سوکت، یا مولفه نمایه شده سوکت، می گیرد و ارزش فعلی آن متغیر، فرمان، یا اعلان (مولفه) برمی گرداند.

یادآوری ۱- نوع ارزش بازگشتی از کارکرد uis:value() همان نوع عنصر سوکتی است که توسط آرگومان 'path' ارجاع داده می شود توجه کنید که، اگر چه نوع بازگشتی ایستا xsd:anyType است، اما نوع پویا ارزش بازگشتی، همیشه خاص تر (مشتق شده از نوع ایستا) است. که توسط عملگر بولی 'instance of' می تواند پرسمان شود.

قواعد نحوی زیر برای مسیر تعریف می‌شود (در Extended BNF¹، به استاندارد ISO / IEC 14977 مراجعه کنید):

- path = absolutePath | relativePath | shortcutPath؛
- absolutePath = “/”, { setPath, “/” }, elementPath؛
- shortcutPath = elementPath؛
- setPath = setId, { “[“, setIndex “]” }؛
- elementPath = normalElementPath | basicCommandPath | timedCommandPath | notifyPath
- normalElementPath = elementId, { “[“, elementIndex, “]” }؛
- basicCommandPath = elementId, { “[“, elementIndex, “]” }؛
- timedCommandPath = elementId, { “[“, elementIndex, “]” }, [“[ttc]”]؛
- notifyPath = elementId, { “[“, elementIndex, “]” } [“[ttc]”]؛

یک مسیر باید یک مسیر مطلق، نسبی یا یک مسیر میان‌بر باشد. یک مسیر مطلق باید فهرستی از شناسانه‌های مجموعه و شناسانه عنصر که با نشان ممیز از هم جدا شده‌اند باشد که مسیری از عنصر ریشه <uiSocket> (به زیربند ۶-۱ مراجعه کنید) به سمت پایین، عنصر سوکت است، همراه با نمایه‌هایی در پرانتز مربعی در هر جا که یک مجموعه یا عنصر بُعددار^۲ شده است. یک مسیر مطلق باید با یک نویسه ممیز (“/”) که نمایانگر عنصر ریشه است، شروع شود. یک مسیر نسبی باید به عنوان عنصر زمینه (گره)، عنصر یا مجموعه سوکت داشته باشد که وابستگی بر روی آن‌ها تعریف شود. مسیرهای نسبی که با “/” شروع می‌شوند عنصر زمینه خود را به عنوان نقطه شروع برای مسیر بعدی قرار می‌دهند. مسیرهای نسبی که با “./” شروع می‌شوند به عنوان اولین آرگومان در مسیر، به <set> والد اشاره دارند. هر “./” بعدی، در مسیر، به والد بعدی در جهت ریشه اشاره دارد. هیچ نمایه‌ای نباید برای هر مجموعه یا عنصر دارای بُعد تعریف شود که توسط “./” یا “/” به آن‌ها اشاره می‌شود. در زمان اجرا، نمایه‌های از دست رفته (با شروع از ریشه)، از مجموعه / عنصر دارای وابستگی، گرفته می‌شود، اگر هیچ چیز دیگری در مسیر مشخص نشده باشد. این بدان معناست که مسیرهای نسبی روی مجموعه‌ها یا عناصر دارای بُعد، ذاتاً به عناصری با زیرمجموعه یا مجموعه نمایه‌های یکسان اشاره دارند (یعنی به مولفه “slice” یکسان اشاره می‌کنند).

یک مسیر میانبر باید شناسانه‌های مجموعه را حذف کند، و فقط باید یک شناسانه و نمایه عنصر، در صورت وجود، استفاده کند. نباید یک نویسه ممیز (“/”) داشته باشد. یک مسیر میانبر نباید استفاده شود اگر مسیر شامل مجموعه دارای بُعد باشد.

setId جایگاهی برای مشخصه “id” از یک عنصر <set> است (که یک جد عنصر سوکت درخواست شده است). برای عناصر <set> دارای بُعد (برای مثال عناصر <set> با یک مشخصه “dim” غیر خالی)، setId

1-Extended Backus-Naur Form

مجموعه‌ای از نمادها که به صورت رسمی قواعد نحوی را توضیح می‌دهد

2-Dimension

یک جایگاه برای ارزش نمایه عنصر <set> است، با ارزش نمایه‌ای که با نوع نمایه مرتبط سازگار شده است. عناصر <set> بدون بُعد نباید setIndex داشته باشند.

elementId یک جایگاه برای مشخصه "id" از یک عنصر <variable> (به زیربند ۸-۱ مراجعه کنید)، <command> (به زیربند ۱-۱ مراجعه کنید)، یا <notify> است. برای عناصر دارای بُعد، یک elementIndex باید به عنوان جایگاهی برای یک ارزش نمایه عنصر باشد با ارزش نمایه‌ای که با نوع نمایه مرتبط سازگار شده است. عناصر بدون بُعد نباید elementId داشته باشند.

- برای عناصر <command> از نوع uis:timedCommand انتخابگر "[ttc]" باید در انتهای مسیر اضافه شود.

- برای عناصر <notify> با یک مشخصه 'timeout'، انتخابگر "[ttc]" باید در انتهای مسیر اضافه شود. آرگومان مسیر باید یک رشته باشد. نویسه‌های زیر باید نادیده گرفته شوند اگر به عنوان بخشی از setIndex یا elementIndex استفاده شوند، چنانچه در زیر آمده است. '^' باید به عنوان نویسه گریز استفاده شود.

- "[^]" باید به عنوان "[^]" کد شود.

- "[^]" باید به عنوان "[^]" کد شود.

- "[^^]" باید به عنوان "[^^]" کد شود.

یادآوری ۲- برای مسیرهای سخت-رمز شده (مسیری که در زمان نوشتن شناخته می‌شود)، نویسنده می‌تواند یک رشته استفاده کند که در نقل قول تکی (') یا دوتایی (") محصور شده است. برای مسیرهای که در زمان اجرا محاسبه می‌شوند (مثال: زمان ارجاع متغیر را به عنوان ارزش نمایه)، نویسنده می‌تواند عملیات‌های رشته XPath معتبر (به بند ۵-۲ مراجعه کنید) برای الحاق یک رشته مسیر بادوام استفاده کند. مثال زیر را ببینید.

ارزش بازگشتی باید ارزش فعلی عنصر سوکت مشخص شده، باشد (یا مولفه آن، اگر یک عنصر دارای بُعد است یا مجموعه دارای بُعد به عنوان جدا دارد). برای انواع فرمان که اطلاعات حالت ندارند (uis:voidCommand)، یک رشته خالی بازگردانده خواهد شد. برای یک فرمان از نوع uis:basicCommand یا uis:timedCommand و بدون elementIndex، حالت (به عنوان رشته) فرمان یا مولفه‌هایش باید بازگردانده شود (به زیربند ۹-۳ مراجعه کنید). برای فرمان‌ها از نوع uis:timedCommand و elementIndex از "ttc"، زمان تکمیل (به عنوان رشته در قالب xsd:duration) فرمان یا مولفه‌هایش باید بازگردانده شود اگر در حال حاضر تعریف شده باشد، در غیر این صورت یک رشته خالی بازگردانده می‌شود. برای یک اعلان و بدون elementIndex مشخص شده، حالتش (به عنوان رشته) یا حالت مولفه‌اش باید بازگردانده شود (مقادیر بازگشتی معتبر "active"، "inactive" و "stacked" هستند). برای یک اعلان و یک elementIndex از "ttc"، یک ارزش نشان‌دهنده زمان باقی مانده از مهلت اتمام (در واحد ثانیه) که با رسیدن به مهلت اتمام، مولفه‌اش باید بازگردانده شود.

uis:value(string path) باید به یک نتیجه تعریف نشده برای عناصر سوکت (یا مولفه‌هایشان) که یک ارزش/حالت تعریف نشده دارند، برسد. در این مورد کل عبارت (که uis:value(path) بخشی از آن است) باید یک ارزش تعریف نشده داشته باشد.

مثال ۱: یک متغیر از نوع `xsd:string` با شناسانه `"var"` داخل دو مجموعه با شناسانه‌های `"outerSet"` و `"innerSet"` تودرتو قرار دارد. نه متغیر نه هیچ‌یک از مجموعه‌های تودرتو، دارای بُعد نیستند. و ارزشش توسط یکی از عبارات XPath زیر می‌شود بازیابی کرد:

```
uis:value("/outerSet/innerSet/var")
uis:value("var")
```

مثال ۲: یک فرمان از نوع `uis:timedCommand` با شناسانه `"cmd"` را در مجموعه‌ای با شناسانه `"setId"` تودرتو قرار دارد. و حالتش توسط یکی از عبارات XPath زیر می‌شود بازیابی کرد:

```
uis:value("/setId/cmd")
uis:value("cmd")
```

و ارزش مولفه `"ttc"` توسط یکی از عبارات XPath زیر می‌شود بازیابی کرد:

```
uis:value("/setId/cmd[ttc]")
uis:value("cmd[ttc]")
```

مثال ۳: یک اعلان با شناسانه `"notifyId"` در مجموعه‌ای با شناسانه `"setId"` تودرتو قرار دارد. و حالتش توسط یکی از عبارات XPath زیر می‌شود بازیابی کرد:

```
uis:value("/setId/notifyId")
uis:value("notifyId")
```

یکی از عبارات XPath زیر است:

مثال ۴: یک متغیر از نوع `xsd:string` با شناسانه `"var"` دارای یک بعد با نوع نمایه `xsd:string` است. که داخل یک عنصر مجموعه‌ای بدون بُعد با شناسانه `"setId"` تودرتو قرار دارد.

ارزش مولفه‌اش با نمایه `"alpha"` توسط یکی از عبارات XPath زیر می‌شود بازیابی کرد:

```
uis:value("/setId/var[alpha]")
uis:value("var[alpha]")
```

و ارزش مولفه با نمایه `"3*[2^3]"` توسط یکی از عبارات XPath زیر می‌شود بازیابی کرد (توجه داشته باشید که `"^"`، `"["`، `"]"` نیاز است که نادیده گرفته شود):

```
uis:value("/setId/var[3*^[2^3]]")
uis:value("var[3*^[2^3]]")
```

مثال ۵: مانند مثال ۵، اما در حال حاضر بازیابی ارزش مولفه با یک ارزش نمایه گرفته‌شده از متغیر دیگر با شناسانه `"index"`:

```
uis:value(concat("/setId/var[", uis:value("index"), "]"))
uis:value(concat("var[", uis:value("index"), "]"))
```

مثال ۶: یک فرمان از نوع `uis:timedCommand` با شناسانه `"cmd"` یک بعد همراه با نوع نمایه `xsd:integer` دارد. که داخل یک عنصر مجموعه‌ای بدون بُعد با شناسانه `"setId"` تودرتو قرار دارد. فیلد `"ttc"` فرمان با نمایه `0` (عدد صفر) توسط یکی از عبارات XPath زیر می‌شود بازیابی کرد:

```
uis:value("/setId/cmd[0][ttc]")
uis:value("cmd[0][ttc]")
```

مثال ۷: یک متغیر از نوع xsd:string با شناسانه "var" دارای دوبعد با انواع نمایه xsd:string و xsd:integer است. که داخل یک عنصر مجموعه‌ای بدون بُعد با شناسانه "setId" تودرتو قرار دارد. ارزش مولفه‌اش با نمایه‌های "3" و "none" توسط یکی از عبارات XPath زیر می‌شود بازیابی کرد:

```
uis:value("/setId/var[3][none]")
```

```
uis:value("var[3][none]")
```

مثال ۸: یک متغیر از نوع xsd:string با شناسانه "var" دارای دوبعد با انواع نمایه xsd:string و xsd:integer است. که داخل یک عنصر مجموعه‌ای خارجی یک-بعدهی با شناسانه "outerSet" و نوع نمایه xsd:boolean، و یک عنصر مجموعه‌ای داخلی یک-بعدهی با شناسانه "innerSet" و نوع نمایه xsd:boolean، تودرتو قرار دارد. ارزش مولفه‌اش با نمایه‌های "true" (برای عنصر outerSet)، "3" و "none" (برای عنصر var) توسط عبارت XPath زیر می‌شود بازیابی کرد:

```
uis:value("/outerSet[true]/innerSet/var[3][none]")
```

مثال ۹: مانند مثال ۹، اما حالا ارزش‌های سه متغیر دیگر (با شناسانه "innerSet"، "innerSet" و "innerSet") به عنوان ارزش‌های نمایه:

```
uis:value(concat("/outerSet[", uis:value("index1"), "/innerSet/var[", uis:value("index2"), "[",
```

```
uis:value("index3"), "]"))
```

مثال ۱۰: این مثال استفاده از مسیرهای نسبی را نشان می‌دهد. سوکت کمینه زیر، مجموعه‌ای یک-بعدهی با یک نوع نمایه xsd:integer تعریف می‌کند. برای هر نمونه مجموعه، یک متغیر "power" و یک متغیر "dimmer" تعریف می‌شود. ارزش "dimmer" تنها اگر نور مرتبط روشن باشد، می‌تواند تغییر کند. (مسیر نسبی "./power" در وابستگی نوشتن متغیر dimmer، به متغیر "power" اشاره می‌کند، با نمایه مجموعه والد^۱ یکسان با متغیر "dimmer" که شامل وابستگی نوشتن می‌شود).

```
<uiSocket name="http://example.com/lights/socket" id="socket"
```

```
xmlns="http://openurc.org/ns/uiocketdesc-2" xmlns:xsd="http://www.w3-org/2001/XMLSchema">
```

```
<set id="lights" dim="xsd:integer">
```

```
<variable id="power" type="xsd:boolean" />
```

```
<variable id="dimmer" type="dimmerType">
```

```
<dependency>
```

```
<write> value("./power") </write>
```

```
</dependency>
```

```
</variable>
```

```
</set>
```

```
<xsd:schema>
```

```
<xsd:simpleType name="dimmerType" id="dimmerTypeId">
```

```
<xsd:restriction base="xsd:integer">
```



```

<xsd:minInclusive value="0" />
<xsd:maxInclusive value="10" />
</xsd:restriction>
</xsd:simpleType>
</xsd:schema>
</uiSocket>

```

مثال ۱۱: این مثال از یک مسیر نسبی داخل وابستگی <relevant> یک مجموعه استفاده می‌کند. برای هر نمونه مجموعه، یک نمونه متغیر (با id="isRelevant") داخل نمونه مجموعه در حال واپایش است که آیا نمونه مجموعه به کاربر ارائه شود یا خیر.

```

<uiSocket name="http://example.com/relevantset/socket" id="socket"
xmlns="http://openurc.org/ns/uissocetdesc-2" xmlns:xsd="http://www.w3-org/2001/XMLSchema">
<set id="set1" dim="xsd:integer">
<variable id="isRelevant" type="xsd:boolean" />
<variable id="someData" type="xsd:string" />
<dependency>
<relevant> value("./isRelevant")</relevant>
</dependency>
</set>
</uiSocket> "

```

boolean uis:hasDefinedValue(string path) ۳-۵-۲-۵

این کارکرد باید به عنوان آرگومانش، مسیر یک متغیر یا فرمان سوکت را اخذ کند و اگر ارزش فعلی آن متغیر یا فرمان تعریف شده باشد، ارزش "true" و در غیر این صورت "false" را بازمی‌گرداند. برای انواع فرمان که اطلاعات حالت ندارند (uis:voidCommand)، "false" باید بازگردانده شود. مسیر آرگومان به عبارت XPath نشان می‌دهد که باید به یک رشته برسد.

یادآوری- از آنجا که XPath، عملگرهای "or" و "and" را تعریف می‌کند به طوری که عملوند سمت راست نادیده گرفته می‌شود اگر عملوند سمت چپ نتیجه را از پیش تعریف کرده باشد، می‌توان توسط احضار uis:hasDefinedValue(id) قبل از احضار uis:value(id) ارزش/حالت‌های تعریف نشده را بررسی کرد.

مثال: عبارت زیر هرگز یک نتیجه تعریف نشده نخواهد داشت. در صورتی که فرمان با شناسانه 'reset'، حالت تعریف نشده یا حالت 'succeeded' داشته باشد، عبارت صحیح است.
not(uis:hasDefinedValue('reset')) or uis:value('reset') eq 'succeeded'

boolean uis:isAvailable(string path) ۴-۵-۲-۵

این کارکرد باید به عنوان آرگومان، مسیر یک عنصر سوکت را اخذ کند (متغیر، فرمان یا اعلان) و اگر عنصر سوکت در زمان اجرا در دسترس باشد "true" و در غیر این صورت "false" را باز می‌گرداند. مسیر آرگومان به عبارت XPath نشان می‌دهد که باید به یک رشته برسد.

یادآوری ۱- عناصر سوکت می‌توانند به عنوان اختیاری در توصیف سوکت علامت‌گذاری شوند. برای آن عناصر سوکت، در زمان اجرا با احضار `uis:isAvailable()` می‌توان تعیین کرد که آیا در دسترس هستند یا خیر.

یادآوری ۲- از آنجا که XPath عملگرهای "or" و "and" را تعریف می‌کند به طوری که عملوند سمت راست نادیده گرفته می‌شود اگر عملوند سمت چپ نتیجه را از پیش تعریف کرده باشد، می‌توان توسط احضار `uis:isAvailable(id)` قبل از احضار `uis:hasDefinedValue(id)` یا `uis:value(id)` دسترس‌پذیری یک عنصر سوکت را بررسی کرد.

مثال: عبارت زیر فقط نتیجه صحیح خواهد داشت، اگر متغیر 'powerstate' در دسترس باشد، تعریف‌نشده نباشد و ارزش 'standby' داشته باشد.

```
uis:isAvailable('powerstate') and uis:hasDefinedValue('powerstate') and uis:value('reset') eq 'standby'
```

boolean uis:isNotifyActive() ۵-۵-۲-۵

این کارکرد باید هیچ آرگومانی اخذ نکند، و باید یک عبارت بولی بازگرداند. اگر هر عنصر `<notify>` سوکت فعال باشد، این کارکرد باید "true" بازگرداند و اگر هیچ عنصر `<notify>` در سوکت فعال نباشد، "false" باز می‌گرداند.

یادآوری - عبارت "not(uis:isNotifyActive())" می‌تواند برای بررسی حالت عادی مفید باشد (یعنی هیچ اعلانی فعال نیست).

boolean uis:sessionForward(string type, string uri) ۶-۵-۲-۵

این کارکرد پیش‌راندن^۱ نشست را بیان می‌کند (به استاندارد ISO/IEC 24752-1 مراجعه کنید). ارزش آرگومان نوع باید یا "destructive" یا "spawn" باشد. ارزش آرگومان URI باید نام (URI) سوکتی باشد شود که URC به سمتش پیش می‌رود. ارزش بازگشتی باید "true" باشد اگر هدف، رویداد رو به جلو نشست مشخص را به URC بفرستد، در غیر این صورت "false" است. این کارکرد مجاز است در پس‌شرایط^۲ فرمان‌های سوکت (به بند ۹-۹-۵ مراجعه کنید) استفاده شود، به منظور ارائه یک اشاره^۳ به URC که فرمان بتواند یک پیش‌راندن نشست را تحریک^۴ کند.

۶ ساختار توصیف سوکت

۱-۶ کلیات

توصیف سوکت باید یک سند XML با عنصر `<uiSocket>` به عنوان عنصر ریشه آن از فضای نام <http://openurc.org/ns/uisocketdesc-2> باشد.

1-Forwarding
2-Postconditions
3-Hint
4-Trigger

یک سند توصیف سوکت باید در UCS بر اساس استاندارد ISO / IEC 10646 کد شده باشد. برای کدبندی نویسه، باید "utf-8" یا "utf-16" استفاده شود.

یک توصیف سوکت باید در صورت امکان یک نوع MIME از "application/urc-uisocketdesc+xml" داشته باشد. پارامتر 'charset' (به IETF RFC 3023 مراجعه کنید) باید به منظور مشخص کردن کدبندی نویسه توصیف سوکت استفاده شود. ارزش آن باید "utf-8" یا "utf-16" باشد. اگر پارامتر 'charset' نباشد، مرحله‌ای که باید طی شود در XML 1.0 (ویرایش پنجم) مشخص شده است، زیربند ۳-۳-۴ باید دنبال شود تا کدبندی نویسه مشخص شود.

بندهای زیر مشخصه‌ها و عناصر فرعی عنصر <uiSocket> را با جزئیات بیشتری توضیح می‌دهد.
مثال:

```
<uiSocket
about="http://example.com/thermometer/socket"
id="socket"
xmlns="http://openurc.org/ns/uisocketdesc-2"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:dcterms="http://purl.org/dc/terms/"
<dcterms:conformsTo>http://openurc.org/ns/
uisocketdesc-2/isoiec24752-2-2013/dcterms:conformsTo
<!-- ... variables, commands, and notify elements, optionally contained in set elements ... -->
<!-- ... internal xsd type definitions ... -->
</uiSocket>
```

۲-۶ مشخصه 'about'

عنصر <uiSocket> باید یک مشخصه 'about' داشته باشد. مشخصه 'about' اشاره به سوکتی دارد که توسط توصیف سوکت شرح داده شده است. ارزش مشخصه 'about' باید به عنوان محتوای عنصر مشخص شود و باید نام سوکتی باشد که به عنوان یک URI بیان شده است. URI ممکن است قابل نگاشت به منبع باشد یا نباشد.

یادآوری ۱- سازندگان هدف، تشویق می‌شوند توصیف سوکت محصولاتشان را با ارسال توصیف سوکت به URI نام سوکت، در دسترس عموم قرار دهند.

به طور معمول این URI از URI هدف توسط عمل الحاق مشتق شده است.

مثال: اگر URI هدف <http://example.com/target> باشد، نتیجه <http://example.com/target/socket> خواهد بود.

یادآوری ۲- همان URI ای که به عنوان یک مرجع سوکت در توصیف سوکت به کار برده می‌شود (مشخصه 'about' از عنصر <uiSocket>)، و نام سوکت در توصیف هدف (مشخصه 'name' از عنصر <uiSocket> متناظر). برای جزئیات بیشتر در مورد توصیف هدف به استاندارد ISIRI 11493-4 مراجعه کنید.

۳-۶ مشخصه 'id'

عنصر <uiSocket> باید یک مشخصه 'id' داشته باشد که شناسانه‌ای را مشخص می‌کند که در میان همه شناسانه‌ها در توصیف سوکت منحصر به فرد است.

یادآوری - شناسانه برای مشخص کردن منابع بیرونی برای عنصر ریشه توصیف سوکت به کار برده می‌شود، همانطور که در استاندارد ISO/IEC 24752 قسمت ۵ با مثال بیان شده است.

۴-۶ مشخصه 'sufficient'

عنصر `<uiSocket>` مجاز است یک مشخصه 'sufficient' از نوع بولی داشته باشد. ارزش پیش فرض آن باید "false" باشد.

ارزش "true" نشان می‌دهد که همه فرمان‌ها در توصیف سوکت که خودشان هیچ مشخصه 'sufficient' ندارند، به طور کامل مشخص شده‌اند همانطور که در بند ۹-۶ توضیح داده شده است.

یادآوری - مشخصه 'sufficient' برای `<uiSocket>`، برای راحتی نویسنده ارائه شده است که می‌تواند توسط مشخصه‌های 'name' جداگانه‌ای روی عناصر `<command>` منفردی بازنویسی شود (به بند ۹-۱ مراجعه کنید).

۵-۶ مشخصه 'complete'

عنصر `<uiSocket>` مجاز است یک مشخصه 'complete' از نوع بولی داشته باشد. ارزش پیش فرض آن باید "false" باشد.

ارزش "true" نشان می‌دهد که همه فرمان‌ها در توصیف سوکت که خودشان هیچ مشخصه 'complete' ندارند، به طور کامل مشخص شده‌اند همانطور که در بند ۹-۶ توضیح داده شده است.

یادآوری - مشخصه 'complete' برای `<uiSocket>`، برای راحتی نویسنده ارائه شده است که می‌تواند توسط مشخصه‌های 'name' جداگانه‌ای روی عناصر `<command>` منفردی بازنویسی شود (به بند ۹-۱ مراجعه کنید).

۶-۶ مشخصه 'extend'

یک سوکت مجاز است یک یا چند سوکت دیگر را توسعه دهد (یعنی از یک یا چند سوکت به ارث ببرند). سوکتی که یک توسعه از یک یا چند سوکت دیگر است زیر-سوکت نامیده می‌شود، و سوکت‌هایی که توسعه داده می‌شود ابر-سوکت نامیده می‌شوند.

یک زیر-سوکت باید موارد زیر را از ابر-سوکت‌ها به ارث ببرد:

-همه مجموعه‌ها

-همه متغیرها، فرمان‌ها، و اعلان‌ها (با انواع آن‌ها)

-همه تعاریف عنصر و نوع داخلی (از قسمت `<xsd:schema>`)

یادآوری ۱- مخزن^۱ سوکت (یعنی عنصر `<uiSocket>`) به ارث برده نمی‌شود.

یادآوری ۲- ارجاع‌های فضای نام به ارث برده نمی‌شوند. اگر یک زیر-سوکت یک متغیر با یک تعریف نوع بیرونی اضافه کند، باید فضای نام را برای نوع بیرونی اعلان کند حتی اگر آن فضای نام در حال حاضر در ابر-قاب اعلان شده است.

عنصر `<uiSocket>` مجاز است یک مشخصه 'extend' با یک فهرست با فضای خالی جدا شده^۲ از نام‌های سوکت (URIها) داشته باشد که یک فهرست مرتب از ابر-سوکت‌هایی را مشخص می‌کند که زیر-سوکت از آن‌ها به ارث برده می‌شود.

1-Container

2-Space-separated

مثال: سوکت با اسم <http://example.com/thermometerplus/socket> سوکت با اسم <http://example.com/thermometerplus/socket> را توسعه می‌دهد.

در موردی که شناسانه یکسانی (ارزش مشخصه 'id') در عناصر سوکت یک یا چند سوپر-سوکت یا زیر-سوکت حادث می‌شود، آخرین بار بر موارد قبلی بازنویسی می‌شود، یعنی موارد قبلی عناصر سوکت توسط زیر-سوکت به ارث برده نمی‌شوند، ابر-سوکت‌ها در ترتیبشان، اول آورده می‌شوند سپس زیر-سوکت‌ها می‌آیند. ترتیب بین ابر-سوکت‌ها به ترتیبی است که نام‌های سوکت‌ها (URIها) در ارزش مشخصه به وجود می‌آیند.

اگر عنصر زیر-سوکت یک عنصر ابر-سوکت را بازنویسی کند (توسط استفاده از شناسانه یکسان)، عنصر ابر-سوکت مورد نظر نباید داخل زیر-سوکت به ارث برده شود، یعنی زیر-سوکت حاصل فقط شامل عنصر موردنظر می‌شود همانطور که در توصیف زیر-سوکت مشخص شده است.

اگر یک عنصر ابر-سوکت یک یا چند عنصر ابر-سوکت دیگر را با استفاده از یک شناسانه بازنویسی کند، عناصر بازنویسی شده (همه به جز آخری) باید نادیده گرفته شوند و تنها عنصر مورد نظر آخرین ابر-سوکت باید داخل زیر-سوکت به ارث برده شود (مگر اینکه یک عنصر با همان شناسانه تعریف شده در زیر-سوکت وجود داشته باشد که در این مورد عنصر زیر-سوکت، همه عناصر ابر-سوکت را با یک شناسانه بازنویسی می‌کند).

۶-۷ عنصر <dcterms:conformsTo>

عنصر <uiSocket> باید یک زیرعنصر <dcterms:conformsTo> داشته باشد که یک ارجاع به یک استاندارد منتشر شده‌ای مشخص کند که سوکت منطبق بر آن است. ارزش، یک URI، به عنوان محتوای عنصر ارائه می‌شود. ارزش <http://openurc.org/ns/uisocketdesc2/isoiec24752-2-2013> مشخص می‌کند که سوکت توصیف شده با این استاندارد مطابقت دارد.

یادآوری ۱- ارزش عنصر <dcterms:conformsTouiSocket> می‌تواند موقع آزمودن انطباق توصیف سوکت استفاده شود.

۶-۸ عنصر <dcterms:modified>

عنصر <uiSocket> مجاز است یک عنصر <dcterms:modified> شامل شود که نشان دهنده این است که توصیف سوکت نسبت به نسخه اصلی تغییر کرده است، در حالیکه هنوز دز مشخصه 'about' آن، همان URI سوکت ارجاع را به کار می‌برد (به بند ۶-۲ مراجعه کنید). محتوای آن از نوع xsd:date یا xsd:dateTime است.

یادآوری ۱- <dcterms:modified> با عنصر فراداده Dublin Core منطبق است، <http://purl.org/dc/terms/modified>.

۶-۹ خصیصه‌های توصیف سوکت از DCMI

هر عنصر و اصلاح عنصر (به جز <dcterms:conformsTo> و <dcterms:modified> که در بالا اشاره شدند) از استاندارد ISO 15836، مجموعه عنصر DCM یا اصطلاحات فراداده DCMI مجازند برای شرح توصیف سوکت به کار برده شوند. هر کدام از آنها مجازند چندین بار به عنوان زیرعناصر عنصر <uiSocket> اتفاق بیفتند.

به طور خاص، عناصر DCM زیر مجازند به کار برده شوند:

- <dc:creator> - نویسنده توصیف سوکت را مشخص می‌کند.

- <dc:publisher> - ارائه کننده توصیف سوکت را مشخص می‌کند.

<dc:contributer> - نویسنده همکار توصیف سوکت را مشخص می کند.
<dcterms:hasVersion> - نام سوکت جایگزین برای دسترسی به همان قابلیت یا قابلیت های مشابه را مشخص می کند.

<dcterisVersionoF> - نام سوکت اصلی در میان مجموعه سوکت های جایگزین برای دسترسی به همان قابلیت یا قابلیت های مشابه را مشخص می کند.

۶-۱۰ عناصر <variable>، <command>، <notify> و <set>

عنصر <uiSocket> باید یک مجموعه کامل <variable> و <command> و <notify> را شامل شود که همه اطلاعات که توسط کاربر می تواند دریافت شود و دستکاری شود و همه فرمان های موجود برای یک کاربر هدف را ضبط می کنند.

۷ مجموعه ها

۷-۱ کلیات

مجموعه ساختار داخلی سلسله مراتبی از یک سوکت واسط کاربر را بیان می کند. مجموعه ممکن است تو در تو باشد (مثال: یک عنصر <set> ممکن است در یک عنصر <set> دیگر باشد).

یادآوری - عنصر <set> تعریف شده در استاندارد ISO / IEC 24752 برای ارائه یک ساختار سلسله مراتبی داخلی به یک سوکت در نظر گرفته شده است.

۷-۲ مشخصه "id"

یک عنصر <set> باید مشخصه "شناسانه" داشته باشد.
مشخصه "id" باید از نوع ID مثل طرحواره XML قسمت ۲ تعریف شده باشد. این یک شناسانه استفاده شده است که برای اشاره به عنصر داخل توصیف سوکت و ابزاری برای اتصال منابع مانند برچسب فراهم می کند.
ارزش های "id" باید در میان همه "شناسانه" و "name" ها منحصر به فرد باشد.
یادآوری - ارزش های "id" به طور معمول به کاربران ارائه شده است و نیازی نیست که برای انسان قابل فهم باشد.

۷-۳ مشخصه 'dim'

مشخصه 'dim' مجموعه ای بعددار (با یک یا چند بعد) مشخص می کند. در زمان اجرا، مجموعه یک بار برای هر ترکیب ممکن از ارزش نمایه آن معرفی می شود. این نتایج در ارزش های مختلف برای هر عنصر سوکت وجود دارد که (مستقیم یا غیر مستقیم) در مجموعه بعددار موجود می باشد.

۷-۴ وابستگی مجموعه ای

۷-۴-۱ کلیات

۷-۴-۲ وابستگی <relevent>

یک مجموعه باید وابستگی <relevent> خود را به اعضای خود به ارث بدهد (در مورد مجموعه، متغیرها، فرمان ها و اعلان ها)، مگر اینکه عنصر یک وابستگی <relevent> خودش مشخص کرده باشد.

یادآوری- به ارث بردن وابستگی یک راه مناسب برای نویسندگان برای مشخص کردن یک وابستگی (در سطح مجموعه) که وجه مشترک همه (یا بیشتر) از اعضای است.

۷-۴-۳ وابستگی <write>

یک وابستگی <write> ممکن است به عنوان یک زیرعنصر <dependency> در مجموعه باشد. و باید دقیقاً یک بار رخ می دهد.
محتوای عنصر <write> باید به عنوان وابستگی <write> برای متغیرها (به بند ۸-۹-۲ مراجعه کنید) مشخص شده باشد.

۸ متغیرها

۸-۱ کلیات

متغیرهای برای افشای حالت یک هدف به URC استفاده می شوند. ارزش یک متغیر را می توان به کاربر از URC نمایش داد، و کاربر می تواند حالت هدف را با تغییر ارزش متغیر تغییر دهد.

۸-۲ مشخصه "id"

یک عنصر <variable> (به بند ۸-۱ مراجعه کنید) باید "id" مشخصه داشته باشد.

۸-۳ مشخصه "secret"

۸-۳-۱ کلیات

یک عنصر <variable> (به بند ۸-۱ مراجعه کنید) باید یک مشخصه "type" داشته باشد.
به منظور ابراز وابستگی در نحو XPath 2.0 (به بند ۵-۲ مراجعه کنید)، ارزش یک متغیر فهرست باید به عنوان یک دنباله از نوع تفکیک ناپذیر که از آن مشتق شده است (به بند ۵-۲-۲ مراجعه کنید) بازگو شود.

۸-۴ مشخصه 'secret'

عناصر <variable> (به بند ۸-۱ مراجعه کنید) ممکن است یک مشخصه 'secret' داشته باشند. این متغیر بولی نشان می دهد آیا ارزش متغیر در مورد امنیت و حفظ حریم خصوصی حساس است، و نیاز به حفاظت از حریم خصوصی برای حمایت از کاربر است.

۸-۵ مشخصه «sensitive»

یک عنصر <متغیر> (به بند ۸-۱ مراجعه کنید) ممکن است یک مشخصه «sensitive» داشته باشد. این متغیر بولی نشان می دهد آیا متغیر باید به کاربران تحت هر شرایطی ارائه شود.

۸-۶ ویژگی 'optional'

یک متغیر میتواند دارای مشخصه 'optional' نوع بولی باشد. مقدار پیشفرض آن باید "false" باشد.
در صورتی که "optional="true" باشد، متغیر در زمان اجرا به دلیل برخی محدودیت ها در دسترس نمیباشد، مقدار آن باید تعریف نشده و نامشخص بوده که این مشخص نبودن در زمان اجرا اشاره به cru دارد.

مثال: در صورت به کار بردن در عبارات وابستگی، یک متغیر اختیاری میتواند وجود داشته باشد در صورتی که قبل از مقدار دهی حتما تعریف شده باشد.

```
<write> uis:hasDefinedValue('myvar') and uis:value('myvar') eq 4 </write>
```

یادآوری ۱- مثال درباره محدودیت هایی که بر در دسترس بودن عناصر سوکت تاثیر میگذراند: کنترل دسترسی، محصولات مختلف با توصیف سوکت عمومی ولی پیاده سازی متفاوت (مثلا در UPnP DCPs)

یادآوری ۲ - در دسترس بودن عنصر سوکت در زمان اجرا اشاره دارد به CRU از طریق توانایی مربوط به NUT (استاندارد ISO/IEC 24752-1 را ببینید)

در صورتی که متغیر دارای مشخصه 'dim' باشد (بند ۸-۸ را ببینید) مشخصه 'optional' به کل مشخصه اعمال می شود، به عنوان مثال چه ان به صورت یک متغیر بعدی باشد یا خیر.

۷-۸ مشخصه 'final'

بازی متغیر ها از ابتدا نشست توسط هدف فراهم میشوند، و هیچگاه در طول نشست تغییر نمیکنند، این متغیر ها باید به صورت "final=true" مشخص شوند.

یک عنصر <variable> (بند ۸-۱ را ببینید) میتواند شامل مشخصه بولی 'final' باشد، مقدار پیشفرض آن باید "false" باشد.

مثال: <variable id="serialNumber" type="xsd:integer" final="true" />

یادآوری - یک متغیر نهایی قابل نوشتن توسط کاربر نابیده و در نتیجه هیچ وابستگی از نوع <write> را ندارد (بند ۸-۹-۳)

اگر متغیر مشخصه 'dim' را دارا باشد (بند ۸-۸) مشخصه 'final' به کل مشخصه اعمال میشود به طور مثال، همه مقادیر توسط هدف در ابتدا نشست مشخص می شود و در طول نشست اصلا تغییر نمیکنند.

۸-۸ مشخصه 'dim'

مشخصه 'dim' متغیر را به صورت بعدی مشخص میکند (با یک یا چند بعد). در زمان اجرا متغیر های بعدی دارای چندین مقدار میباشند که هر کدام برای ترکیب مشخصی از شاخص ها به کار میرود.

یادآوری ۱- یک متغیر یک بعدی یک ارایه با یک شاخص میباشد، و یک متغیر ۲ بعدی یک جدول است (فکر کنید که جدولی به صورت پراکنده تعریف شده باشد).

یادآوری ۲- هیچ مرتب سازی صریحی برای مقادیر منتج شده از متغیر های بعدی وجود ندارد، با این حال مراتب سازی مقادیر بر اساس مراتب سازی مقدار شاخص توصیه برای هدف توصیه می شود- در زمان انتقال مقدار به URC و برای URC- در زمان معرفی کردن به کاربر (استاندارد ISO/IEC 24752-1)

یادآوری ۳- بیشترین مجموعه ممکن مقادیر برای متغیر سوکت مشخص از حاصلضرب همه انواع شاخص که در زمان پیمودن مسیر از عنصر <uiSocket> تا متغیر سوکت بخصوص به وقوع پیوسته باشد (تعریف مسیر را در زیربند ۲-۵-۲-۵ ببینید). در هر حال همه ترکیب های ممکن در زمان اجرا اتفاق نمیافتد.

مشخصه 'dim' ممکن است برای متغیر ها معرفی شود، در صورت معرفی باید شامل یک لیست نا خالی، مرتب شده و با فاصله جدا شده از ارجاع های نوع که متغیر های انواع شاخص میباشد. اولین ارجاع نوع شاخص برای اولین بعد را مشخص

میکنند ، دومین نوع برای دومین بعد و به همین ترتیب ارجاع های شاخص معتبر شامل (الف) نام نوعی که در <xsd:schema> تعریف شده به صورت بخشی از توصیف سوکت ، و (ب) نام صلاحیت دار (QName) یک نوع خارجی.

یادآوری ۴- نوع شاخص میتواند مقدار نامحدودی از مقادیر ممکن شاخص را داشته باشد (مثلا xsd:integer دارای تعداد نامحدودی مقدار است) ب این حال در زمان اجرا مقادیر واقعی شاخص زیر مجموعه ای شماره از تعداد نامحدود مقادیر مجاز توسط نوع شاخص است.

یادآوری ۵- متغیر های بعدی مانند "ارایه های شرکت پذیر" در بعضی زبان های برنامه نویسی است مانند جاوا اسکریپت.

یادآوری ۶- مشخصه 'type' از متغیر های بعدی نوع هر کدام از مقدر موجود در آن متغیر را مشخص میکند ، و نه نوع همه متغیر ها را (که در ارایه موجودند).

مثال ۱: درجه صوتی دستگاه پخش صوت ۵.۱ به صورت یک بردار از ۶ عدد صحیح میتواند تعریف شود که هر کدام بین ۰ تا ۱۰۰ قرار دارند . هر مقدار مشخص کننده درجه صوتی برای کانال مربوطه میباشد.

```
...
<xsd:schema>
<xsd:simpleType name="volumeType">
<xsd:restriction base="xsd:integer">
<xsd:minInclusive> 0 </xsd:minInclusive>
<xsd:maxInclusive> 100 </xsd:minInclusive>
</xsd:restriction>
</xsd:simpleType>
<xsd:simpleType name="channelType">
<xsd:restriction base="xsd:string">
<xsd:enumeration value="FrontLeft" />
<xsd:enumeration value="FrontCenter" />
<xsd:enumeration value="FrontRight" />
<xsd:enumeration value="RearLeft" />
<xsd:enumeration value="RearRight" />
<xsd:enumeration value="SubWoofers" />
</xsd:restriction>
</xsd:simpleType>
/>xsd:schema<
```

یادآوری ۷- ارجاعات (مانند از منابع تفکیک ناپذیر) به متغیر های بعدی اشاره به یک بخش منحصر به فرد از متغیر دارد . در مثال بالا آدرس <http://example.com/sound/socket#volume> میتواند اشاره به درجه صوتی هر کدام از بخش های منحصر به فرد (کانال) داشته باشد . با این حال آدرس [http://example.com/sound/socket#dims\(volume\)](http://example.com/sound/socket#dims(volume)) میتواند اشاره به عموم متغیر ها با همه بخش هایش داشته باشد مانند درجه صوتی کانال ها به صورت یک موجودیت مرکب. برای جزئیات استاندارد ISO/IEC 24752-5 را ببینید .

یادآوری ۸- ترتیب در هر یک از بخش های یک متغیر بعدی که کاربر معرفی می شود، بر پایه تعریف ترتیب مربوط به نوع (ها) شاخص است . برای جزئیات استاندارد ISO/IEC 24752-1 را ببینید.

۸-۹ وابستگی های متغیر

۸-۹-۱ کلیات

وابستگی های متغیر زمانی واقع می شود که متغیر توسط کاربر قابل نوشتن یا مربوط باشد و چه وابستگی هایی بین مقادیر متغیر های موجود برقرار باشد.

همه وابستگی‌ها باید به عبارات XPath کد گذاری شده و در زمان اجرا اعتبارسنجی شوند. بازی از نوع های وابستگی‌ها ممکن در زمان نشست تغییر کنند .
وابستگی‌های متغیر باید توسط در بر گرفته شدن آنها در نشانگذاری

```
>dependency<  
<relevant>expr1</relevant>  
<write>expr2</write>  
</dependency>
```

مشخص می‌شود .

عنصر <dependency> باید به صورت یک زیر عنصر از <variable> معرفی شود (بند ۸-۱). در صورت معرفی باید فقط یک بار این اتفاق بیفتد.

۲-۹-۸ وابستگی <relevant>

وابستگی <relevant> زمانی که متغیر باید به کاربر معرفی شود مشخص می‌شود. مثلاً چقدر مربوط است با توجه به وضعیت هدف در آن زمان. در صورت درست بودن وابستگی <relevant>، متغیر باید به کاربر معرفی شود. این حتماً به آن معنا نیست که به صورت مستقیم از طریق رابط کاربر مشتق شده از سوکت در دسترس است، بلکه توسط بعضی مسیر های ناوبری کاربر قادر به پیدا کردن آن است .

عنصر <relevant> میتواند به صورت زیر عنصری از <dependency> باشد(بند ۸-۹-۱). در این صورت باید فقط یک بار واقع شود ، در صورت عدم وجود متغیر وابستگی <relevant> را از نزدیکترین عنصر <set> اجداد خود که دارای مشخصه <relevant> است به ارث میبرد.(بند ۷-۴-۲)

مثال: متغیر معرف یک شماره ثبت میتواند فقط در صورت دارا بودن مقدار "student" برای متغیر 'userType' باشد .

```
<dependency>  
<relevant> uis:value('userType') eq 'student' </relevant>  
</dependency>
```

محتویات وابستگی <relevant> باید شامل یک عبارت معتبر XPath با(با ملحقات اجازه داده شده در ۵-۲) که با عدد یا مقدار بولی جاگذاری می‌شود باشد . عدد ها باید در رنج [۰/۰ تا ۱/۰] باشند (هر دو ۰/۰ و ۱/۰ داخل مجموعه اند). این کار اجازه جاگذاری با هر عدد پایین تر برای کم شدن وابستگی میدهد ، که به صورت اعداد بزرگتر برای وابستگی بالاتر و اعداد کوچکتر برای وابستگی کمتر میباشد ، ۰/۰ به صورت نه مربوط و ۱/۰ به صورت مربوط تعریف شده ، در صورتی که عبارت XPath ی مقدار بولی را معرفی کند ، "true" به صورت ۱/۰ و "false" به صورت ۰/۰ مشخص می‌شود .

یادآوری - نتیجه عبارت XPath ممکن است در طول نشست تغییر کند .

در صورت مشخص نشدن وابستگی <relevant>(نه در متغیر و نه در اجدادش) یا اینکه به صورت یک وابستگی <relevant> نشان داده شده باشد ، وابستگی <relevant> باید همیشه صحیح باشد زمانی که عنصر <notify> فعال است.

۳-۹-۸ وابستگی <write>

وابستگی <write> مشخص میکند که کاربر قادر به تغییر مقدار متغیر میباشد یا خیر. در صورت غلط بودن مقدار وابستگی <write> تغییر مقدار متغیر امکان پذیر نیست در غیر این صورت متن آن را با مقدار جدید جایگزین نمود، با این حال هیچ تضمینی برای موفقیت اعمال تغییرات وجود ندارد.

عنصر <write> ممکن است به صورت زیر عنصری از <dependency> (۸-۹-۱) واقع شود، در صورتی که مشخصه 'final' مقدار "false" داشته باشد و یا در متغیر مربوطه موجود نباشد. در صورت وجود فقط یک بر واقع می‌شود. اگر موجود نبود متغیر وابستگی <write> را از نزدیک ترین عنصر <set> جد خود که دارای وابستگی <relevant> است به ارث میبرد (۳-۹-۸)

مثال ۱: یک سیستم رزرواسیون هتل میتواند شامل یک متغیر با نوع پول پرداختی با کردیت باشد که میتواند فقط در صورتی که کاربر پول خود را توسط کارت اعتباری پرداخت کرده عمل کند، این وابستگی به صورت زیر میتواند نشان داده شود:

```
<variable id="creditCard" type="creditCardType">
<dependency>
<write> uis:value('creditPayment') </write>
</dependency>
</variable>
```

مثال ۲: وابستگی‌ها را میتوان برای مشخص کردن قیود ترتیب به کار برد. به عنوان مثال، یک متغیر سرویس میتواند نوشتنی شود بعد از قبول کردن موافقت نامه لایسنس توسط کاربر، موافقت کاربر میتواند به صورت یک متغیر بولی با "id="warrantySigned" مشخص شده و باقی متغیرها برای نوشتن مشخصه‌های فرم به کار میروند

```
<variable id="serviceVar" type="serviceVarType">
<dependency>
<write> uis:value('warrantySigned') </write>
</dependency>
</variable>
```

محتویات عنصر <write> باید خالی یا یک عبارت معتبر XPath (با ملحقات مشخص شده در بند ۵-۲) باشد که به یک مقدار بولی تبدیل می‌شود.

یادآوری ۱- نتیجه عبارت XPath ممکن است در طول نشست تغییر کند.

در صورتی که عنصر <write> متغیر خالی باشد یا یک وابستگی <write> از مجموعه جد خود به ارث برده باشد ممکن است نامعلوم باشد.

یادآوری ۲- مقادیر نامعلوم شبیه مقدار بولی ۳ مقداری که اجازه عدم قطعیت را میدهد است. در صورت نامعلوم بودن وابستگی <write>، هیچ راهی به جز تلاش برای تغییر متغیر ندارد.

در صورت عدم مشخص کردن وابستگی <write> (نه در متغیر و نه در اجدادش) وابستگی <write> متغیر همش صحیح است حتی وقتی که عنصر <notify> فعال باشد.

۴-۹-۸ وابستگی <insert>

وابستگی <insert> مشخص میکند که کاربر آیا میتواند مجموعه مربوط از شاخص های معتبر را در زمان اجرا تغییر دهد یا خیر (که مجموعه مربوط با مشخصه 'dim' بروی عنصر <variable> مربوطه مشخص می شود، ۸-۱)

محتویات وابستگی <insert> میتواند یک عبارت XPath قابل تبدیل به مقدار بولی که در زمان نشست قابل تغییر است باشد. در صورت مقدار دهی به غلط امکان اضافه یا تغییر ترکیب شاخص ها وجود ندارد، در قی این صورت میتوان ترکیبی از شاخص ها را اضافه یا حذف کرد والی هیچ تضمینی برای اعمال تغییرات وجود ندارد.

یادآوری ۱- " اضافه کردن ترکیبی از شاخص " به معنی اضافه کردن بخشی از متغیر برای یک شاخص مشخص جدید در فضای شاخص مشخص شده توسط مشخصه 'dim' است. توجه داشته باشید که مقدار اولیه جز متغیر جدید توسط هدف مشخص می شود. "حذف یک ترکیب شاخص" به معنی پاک کردن یک جز متغیر برای یک شاخص ترکیب شاخص مشخص که در حال حاضر واقع شده است میباشد.

عنصر <insert> میتواند به صورت زیر عنصری از <dependency> باشد (۸-۹-۱) فقط در صورتی که متغیر مربوطه دارای مشخصه 'dim' باشد (۸-۸) و به صورت نهایی نباشد (۸-۷)، در این صورت فقط یک بار میتواند واقع شود. مقدار پیشفرض آن باید "false" باشد.

یادآوری ۲- هیچ نمونه ارث باری از وابستگی <insert> مربوط به عنصر <set> به آنهایی که شامل عنصر <variable> میشوند نیست (۸-۱). هر وابستگی <insert> تنها به مجموعه ای از شاخص ها که با مشخصه 'dim' در همان رد مشخص شده اند وابسته است.

مثال: با توصیف سوکت مقابل، یک جدول را با ستر و ستون مشخص میکنیم، cru میتواند درخواست حذف یا اضافه کردن به سطر ها داشته باشد والی برای ستون ها خیر

```
<set id="row" dim="xsd:positiveInteger">
<dependency>
<insert> true() </insert>
</dependency>
<variable id="column" type="xsd:integer" dim="xsd:positiveInteger">
<dependency>
<insert> false() </insert>
</dependency>
</variable>
</set>
```

۸-۹-۵ وابستگی <length>

عنصر <length> یک عبارت XPath را مشخص میکند (با دستور مشخص شده در ۵-۲) که با یک عدد جایگزین می شود. این عدد مشخص کننده طول اجازه داده شده برای متغیر در زمان اجرا است که نباید در طول نشست تغییر کند

یادآوری - رفتار از شکل تغییر وابستگی <length> تعریف نشده است. برای مثال cru ممکن است هرگونه تغییری را نادیده بگیرد. کنترل همنوایی ممکن است وابستگی <length> که به عنصر سوکتی غیر از متغیر نهایی ارجاع داده باشد را رد کند.

عنصر `<length>` میتواند به عنوان زیر عنصری از `<dependency>` (۸-۹-۱) عمل کند والی فقط در صورتی که نوع متغیر مربوطه از `xsd:string` و یا مشتاق شده از نوع `xsd:string` باشد. در صورت وجود فقط یک بر واقع می‌شود.

مثال ۱: مثال روبرو قید طولی به اندازه ۵ میگذارد

```
<length uis:value("lengthVar") />
```

عنصر `<length>` فقط وقتی باید استفاده شود که مقدار مجاز طول در مقدار دهی اولیه نشست مشخص شده باشد، در ارتباط با باقی عناصر سوکت. اگر متغیر دارای قید طولی که بتواند قبل از زمان اجرا مشخص شود باشد باید آن را به صورت یک نوع که در بخش `<xsd:schema>` (شرایط ۱۱) تعریف شده مشخص کند. اگر متغیری دارای قید طول هم در نوع خودش و عنصر `<length>` باشد، هر دی آنها باید یک عدد را نشان دهند.

۸-۹-۶ وابستگی `<minLength>`

عنصر `<minLength>` یک عبارت XPath را مشخص میکند (با دستور مشخص شده در ۵-۲) که به عدد تغییر میکند. این عدد حداقل طول متغیر در زمان اجرا را مشخص میکند، که در طول نشست نباید تغییر کند.

یادآوری - رفتار در زمان تغییر `<minLength>` تعریف نشده است.

عنصر `<minLength>` میتواند به عنوان زیر عنصری از `<dependency>` (۸-۹-۱) باشد ولی اگر فقط نوع متغیر مربوطه `xsd:string` باشد یا از `xsd:string` مشتاق شده باشد، در صورت وقوع فقط یک بار این اتفاق میفتد.

عنصر `<minLength>` باید فقط وقتی استفاده شود که حداقل طول اجازه داده شده در مقدار دهی اولیه نشست مشخص شده باشد در رابطه با دیگر عناصر سوکت، در صورتی که متغیر دارای قید حداقل طول که قبل از زمان اجرا مشخص شده باشد باید آن را به عنوان نوعی که در بخش `<xsd:schema>` (شرایط ۱۱) توصیف سوکت مشخص کند. اگر متغیر دارای قید حداقل طول در هر دو نوع خود و با عنصر `<length>` باشد باید با هر دو آنها مطابقت کند (به طوری که عدد بزرگتر به عنوان حداقل طول در نظر گرفته می‌شود).

۸-۹-۷ وابستگی `<maxLength>`

عنصر `<maxLength>` یک عبارت XPath را مشخص میکند (با دستور مشخص شده در ۵-۲) که به عدد تغییر میکند. این عدد حد اکثر طول متغیر در زمان اجرا را مشخص میکند، که در طول نشست نباید تغییر کند.

یادآوری - رفتار در زمان تغییر `<maxLength>` تعریف نشده است.

عنصر `<maxLength>` می‌تواند به عنوان زیر عنصری از `<dependency>` (۸-۹-۱) باشد ولی اگر فقط نوع متغیر مربوطه `xsd:string` باشد یا از `xsd:string` مشتق شده باشد، در صورت وقوع فقط یک بار این اتفاق می‌افتد.

عنصر `<maxLength>` باید فقط وقتی استفاده شود که حد اکثر طول اجازه داده شده در مقدار دهی اولیه نشست مشخص شده باشد در رابطه با دیگر عناصر سوکت، در صورتی که متغیر دارای قید حد اکثر طول که قبل از زمان اجرا مشخص شده باشد باید آن را به عنوان نوعی که در بخش `<xsd:schema>` (شرایط ۱۱) توصیف سوکت مشخص کند. اگر متغیر دارای قید حد اکثر طول در هر دو نوع خود و با عنصر `<length>` باشد باید با هر دو آنها مطابقت کند (به طوری که عدد کوچکتر به عنوان حد اکثر طول در نظر گرفته می‌شود).

۸-۹-۸ وابستگی `<pattern>`

عنصر `<pattern>` یک عبارت XPath را مشخص میکند (با دستور مشخص شده در ۵-۲) که به رشته تغییر میکند. این رشته یک عبارت عمومی که یک الگو را برای مقادیر مجاز متغیر در زمان اجرا مشخص میکند را نشان می‌دهد. دستور الگو به صورت قید `'pattern'` در شما `lmx` بخش ۲ مشخص شده. الگو نباید در طول نشست تغییر کند.

یادآوری ۱ - رفتار در زمان تغییر `<pattern>` تعریف نشده است.

عنصر `<pattern>` می‌تواند به عنوان زیر عنصری از `<dependency>` (۸-۹-۱) باشد ولی اگر فقط نوع متغیر مربوطه `xsd:string` باشد یا از `xsd:string` مشتق شده باشد.

یادآوری ۲ - در صورتی که فضای مقدار متغیر نیاز به توصیف داشته باشد توسط یک اجتماع الگوها آنها می‌توانند با هم `rO` شوند توسط اپراتور `"|"` موجود در عبارت عمومی.

مثال ۱: عبارت روبرو یک الگو برای نشان دادن رشته ای هر عدد صحیح مثبت به کار می‌رود

```
<pattern> "[1-9][0-9]*" </pattern>
```

مثال ۲: عبارت روبرو یک مقدار نهایی `'patternVar'` که به عنوان الگو استفاده می‌شود را مشخص میکند:

```
<pattern> uis:value("patternVar") </pattern>
```

یادآوری ۳ - اگر هر کدام از کارکترهای `'&'`، `'>'` یا `'<'` واقع شوند در داخل `<pattern>` باید توسط گریز `lmx` مشخص شوند.

مثال ۳: عبارت روبرو الگویی برای رشته ای به صورت `"A&B"`، `"A&A"` و `""`، مشخص میکند

```
<pattern> "[A-Z]&[A-Z]" </pattern>
```

عنصر `<pattern>` باید فقط زمانی استفاده شود که در زمان مقدار دهی اولیه نشست الگو مورد نظر مشخص شده باشد در رابطه با دیگر عناصر سوکت. در صورتی که متغیر دارای قید الگو که قبل از زمان اجرا مشخص شده باشد باید آن را به عنوان نوعی که در بخش `<xsd:schema>` (شرایط ۱۱) توصیف سوکت

مشخص کند. اگر متغیر دارای قید الگو در هر دو نوع خود و با عنصر `<pattern>` باشد باید با هر دو آنها مطابقت کند (به طوری که آنها با هم dnA منطقی میشوند).

۸-۹-۹ وابستگی `<minInclusive>`

عنصر `<minInclusive>` یک عبارت XPath را (۵-۲) را مشخص میکند که به مقداری مشابه مقدار همان نوع که متغیر به آن مربوط است مقدار دهی می‌شود. این مقدار حداقل طول (جامع) را برای زمان اجرا متغیر نشان میدهد که در طول نشست تغییر نمیکند
یادآوری ۱- رفتار در زمان تغییر `<minInclusive>` تعریف نشده است.

عنصر `<minInclusive>` میتواند به عنوان زیر عنصری از `<dependency>` (۸-۹-۱) باشد ولی اگر فقط نوع متغیر مربوط دارای نوعی باشد که فضای مقدار آن کاملاً مرتب باشد، در صورت وجود فقط یک بر واقع می‌شود.

یادآوری ۲- در `lms`، نوع داده ای توکار نه مرتب میباشد (مانند `xsd:string`) یا فقط بخشی مرتب اند (مانند `xsd:time`)، برای این انواع داده ای، `<minInclusive>` و دیگر قیود زمان اجرا که احتیاج به رابطه کاملاً مرتب دارند نمیتوان استفاده کرد.

مثال ۱: عبارت ذیل ۱۰ را به عنوان یک مقدار پایین جامع برای متغیر از نوع `xsd:integer` تعریف میکند:

```
<minInclusive> 10 </minInclusive>
```

مثال ۲: عبارت ذیل دوبرابر مقدار یک متغیر نهایی با `id` 'lowerBound' را به صورت مقدار پایین جامع برای متغیر از نوع `xsd:integer` تعریف میکند:

```
<minInclusive> 2 * uis:value("lowerBound") </minInclusive>
```

عنصر `<minInclusive>` باید فقط زمانی استفاده شود که در زمان مقدار دهی اولیه نشست الگو مورد نظر مشخص شده باشد در رابطه با دیگر عناصر سوکت. در صورتی که متغیر دارای قید مقدار پایین که قبل از زمان اجرا مشخص شده باشد باید آن را به عنوان نوعی که در بخش `<xsd:schema>` (شرایط ۱۱) توصیف سوکت مشخص کند. اگر متغیر دارای قید مقدار پایین در هر دو نوع خود و با عنصر `<pattern>` باشد باید با هر دو آنها مطابقت کند (مقدار بزرگتر انتخاب می‌شود).

۸-۹-۱۰ وابستگی `<maxInclusive>`

عنصر `<maxInclusive>` یک عبارت XPath را (۵-۲) را مشخص میکند که به مقداری مشابه مقدار همان نوع که متغیر به آن مربوط است مقدار دهی می‌شود. این مقدار حد اکثر طول (جامع) را برای زمان اجرا متغیر نشان میدهد که در طول نشست تغییر نمیکند
یادآوری ۱- رفتار در زمان تغییر `<maxInclusive>` تعریف نشده است.

عنصر `<maxInclusive>` میتواند به عنوان زیر عنصری از `<dependency>` (۸-۹-۱) باشد ولی اگر فقط نوع متغیر مربوط دارای نوعی باشد که فضای مقدار آن کاملاً مرتب باشد، در صورت وجود فقط یک بار واقع می‌شود.

عنصر `<maxInclusive>` باید فقط زمانی استفاده شود که در زمان مقدار دهی اولیه نشست الگو مورد نظر مشخص شده باشد در رابطه با دیگر عناصر سوکت . در صورتی که متغیر دارای قید مقدار بالا که قبل از زمان اجرا مشخص شده باشد باید ان را به عنوان نوعی که در بخش `<xsd:schema>` (شرایط ۱۱) توصیف سوکت مشخص کند . اگر متغیر دارای قید مقدار بالا در هر دو نوع خود و با عنصر `<pattern>` باشد باید با هر دو آنها مطابقت کند (مقدار کوچکتر انتخاب می‌شود).

۱۱-۹-۸ وابستگی `<<minExclusive>`

عنصر `<<minExclusive>` یک عبارت XPath را (۵-۲) را مشخص میکند که به مقداری مشابه مقدار همان نوع که متغیر به ان مربوط است مقدار دهی می‌شود . این مقدار حداقل طول (صریح) را برای زمان اجرا متغیر نشان میدهد که در طول نشست تغییر نمیکند
یادآوری ۱- رفتار در زمان تغییر `<<<minExclusive>` تعریف نشده است .

عنصر `<<minExclusive>` میتواند به عنوان زیر عنصری از `<dependency>` (۸-۹-۱) باشد ولی اگر فقط نوع متغیر مربوط دارای نوعی باشد که فضای مقدار ان کاملا مرتب باشد، در صورت وجود فقط یک بار واقع می‌شود .

عنصر `<<minExclusive>` باید فقط زمانی استفاده شود که در زمان مقدار دهی اولیه نشست الگو مورد نظر مشخص شده باشد در رابطه با دیگر عناصر سوکت . در صورتی که متغیر دارای قید مقدار پایین که قبل از زمان اجرا مشخص شده باشد باید ان را به عنوان نوعی که در بخش `<xsd:schema>` (شرایط ۱۱) توصیف سوکت مشخص کند . اگر متغیر دارای قید مقدار پایین در هر دو نوع خود و با عنصر `<pattern>` باشد باید با هر دو آنها مطابقت کند (مقدار بزرگتر انتخاب می‌شود).

۱۰-۹-۸ وابستگی `<<maxExclusive>`

عنصر `<<maxExclusive>` یک عبارت XPath را (۵-۲) را مشخص میکند که به مقداری مشابه مقدار همان نوع که متغیر به ان مربوط است مقدار دهی می‌شود . این مقدار حد اکثر طول (صریح) را برای زمان اجرا متغیر نشان میدهد که در طول نشست تغییر نمیکند
یادآوری ۱- رفتار در زمان تغییر `<<<maxExclusive>` تعریف نشده است .

عنصر `<<maxExclusive>` میتواند به عنوان زیر عنصری از `<dependency>` (۸-۹-۱) باشد ولی اگر فقط نوع متغیر مربوط دارای نوعی باشد که فضای مقدار ان کاملا مرتب باشد، در صورت وجود فقط یک بار واقع می‌شود .

عنصر `<<maxExclusive>` باید فقط زمانی استفاده شود که در زمان مقدار دهی اولیه نشست الگو مورد نظر مشخص شده باشد در رابطه با دیگر عناصر سوکت . در صورتی که متغیر دارای قید مقدار بالا که قبل از زمان اجرا مشخص شده باشد باید ان را به عنوان نوعی که در بخش `<xsd:schema>` (شرایط ۱۱) توصیف سوکت مشخص کند . اگر متغیر دارای قید مقدار بالا در هر دو نوع خود و با عنصر `<pattern>` باشد باید با هر دو آنها مطابقت کند (مقدار کوچکتر انتخاب می‌شود).

۸-۱۰ انتخاب

۸-۱۰-۱ کلیات

متغیر لیست (متغیری که نوع آن مشتق از لیست است) ممکن است مجموعه ای از مقادیر را در فضای مقدار خاصی محدود باشد (انتخاب بسته) و یا امکان مقادیر پیشنهادی برای کاربر را فراهم کند (انتخاب باز). عناصر لیست باید از همان نوع عناصر محدود شده برخوردار باشند یا زیر نوع مربوطه. مجموعه مقادیر با عنصر `<selection>` برای متغیر محدود شده مشخص می‌شود، هر دو مقادیر پویا و ایستا میتوانند تعریف و ترکیب شوند توسط یک عنصر `<selection>`.

عناصر `<selection>` میتواند در عنصر `<variable>` واقع باشد، در این صورت فقط یک بار واقع می‌شود.

مثال `</selection> ... <selection>`

۸-۱۰-۲ مشخصه 'closed'

مشخصه 'closed' با نوع بولی میتواند در همکاری با عنصر `<selection>` برای مشخص کردن اینکه مقادیر بسته یا باز هستند استفاده شود، مقدار پیشفرض "true" به مانی انتخاب بسته است.

مثال `</selection> ... <selection closed="false">`

مقدار "false" به معنی انتخاب باز نشانگر انتخاب با مقادیر انتخابی پیشنهادی است که کاربر ممکن است مقداری متفاوت با پیشنهادات استفاده کند.

۸-۱۰-۳ مجموعه انتخاب پویا یا ایستا

۸-۱۰-۳-۱ کلیات

انتخاب میتواند مرکب از یک یا چند مجموعه انتخاب که پویا یا ایستا هستند باشد. هر مجموعه انتخاب باید توسط عناصر جدا `<selectionSetStatic>` یا `<selectionSetDynamic>` مشخص شوند و دارای مشخصه 'id' از نوع `xsd:ID` باشند که در مجموعه بتوان به آنها ارجاع نمود.

مثال: یک انتخاب بسته مرکب از دو مجموعه انتخاب یکی حاوی مجموعه ای ایستا از شهرها (`myStaticCities`) که بر اساس تعریف نوع `staticCityType` و دیگری با مجموعه ای پویا از شهرها (`myDynamicCities`) که در زمان اجرا توسط متغیر `dynCitiesVariable` مشخص شده. توجه کنید که `staticCityType` از نوع `uis:stringListItem` مشتق شده (مثال ۱۱-۳) که خود مشتق شده از `xsd:string` است (۱۱-۳). و `dynCitiesVariable` از نوع `uis:stringList` است.

```
<selection>
<selectionSetStatic id="myStaticCities" typeRef="staticCityType"/>
<selectionSetDynamic id="myDynamicCities" varRef="dynCitiesVariable"/>
</selection>
```

۸-۱۰-۳-۲ عنصر `<selectionSetStatic>`

یک مجموعه انتخاب ایستا (در زمان اجرا تغییر نمیکند) توسط عنصر `<selectionSetStatic>` مشخص می‌شود که باید شامل مشخصه 'typeRef' که شامل نام نوع است باشد.

نوع مرجع باید توسط محدودیت از انواع ساده مشتق شده باشد، اجتماع شمارشی نیز مورد قبول است و مکانیزمی برای دریافت مجموعه انتخاب را ارائه میکند. برای دیدن مثال برای این انواع ۱۱-۴ را ببینید.

انتخاب معتبر زمانی است که تمام عناصر با نوع داده شده مطابق باشند .

یادآوری - عامل کاربر ممکن است مجموعه ای از مقادیر را با رندر کردن آنها تاثیر به روی نوع مرجع آنها داشته باشد و مجموعه انتخاب در زمانی که همچنین مجموعه ای موجود باشد . مثلا تقسیم کننده ها میتوانند قسمت های مختلف یک جعبه انتخاب را در محیط IUG انجام دهند.

۸-۱۰-۳ عنصر <selectionSetDynamic>

مجموعه انتخاب پویا باید با <selectionSetDynamic> به همراه مشخصه 'varRef' که باید شامل di متغیر نوع uis:csvlist یا di لیست با فاصله جدا شده باشد (۸-۳-۳) که نوع عضو هماهنگ با نوع متغیر میزبان میباشد. متغیر لیست ارجاعی نباید بعدی باشند .

انتخاب معتبر همه عناصری که در لیست متغیر قرار دارند را شامل است .

در صورتی که متغیر میزبان بعدی نباشد و در مجموعه ای بعدی نیز قرار نداشته باشد لیست متغیر های ارجاعی نباید شامل یک یا چند مجموعه بعدی باشند و در صورتی که متغیر میزبان بعدی باشد مسیر مرتبط برای لیست متغیر ها باید از طریق به اشتراک گذاری شاخص های مجموعه هایی که دارای مشخصه 'varRef' میباشد باشد .

مثال: در مجموعه تلویزیون متغیر "channel" فقط برای مقادیر موجود در مجموعه "channelList" اجازه داده می شود ، در طول زمان اجرا مقدار متغیر "channelList" میتواند در هر زمان تغییر کند توسط تغییر مقادیر مجموعه "channel".

```
<variable id="channel" type="xsd:string">
<selection closed="true">
<selectionSetDynamic id="currentChannels" varRef="channelList" />
</selection>
</variable>
<variable id="channelList" type="uis:stringList">
```

۸-۱۱ اطلاعات نگاشت متغیر مربوط به سکو

عنصر <mapping> میتواند هر اندازه استفاده شود به صورت زیر عنصری از <variable> (۸-۱) برای قرار دادن اطلاعات نگاشت متغیر مربوط به سکو.

عنصر و باید مشخصه 'platform' که توسط ISO/IEC 24752 محدود نشده را دارا باشد .

عنصر <mapping> ممکن است محتویات عناصر گوناگون و زیر عناصر باشد با این حال زیر عناصر باید از فضای نامی متفاوت با فضای نام siu باشند .

مثال: اطلاعات نگاشت مختص به UPnP برای یک متغیر ممکن است به صورت ذیل باشد :

```
<mapping platform="UPnP">
...
/>mapping<
```

یادآوری ۱- توصیف سوکت که شامل اطلاعات مختص سکو باشد بیطرفی سکو خود را از دست میدهد ، اگر چه چندین نگاشت ممکن است در توصیف سوکت اتفاق بیفتد (هر کدام برای یک سکو). توصیه می شود مکانیسم های دیگر برای اتصال

اطلاعات مختص سکو را در نظر بگیرید . به عنوان مثال نگاشت اطلاعات میتواند در یک فایل خارجی با ارجاع به عناصر توصیف سوکت اعمال شود .

یادآوری ۲- تولید کنندگان و سکو ها به شدت از به کار گیری عنصر <mapping> در همراه کردن محتویات فعال یا اجرایی در توصیف سوکت ناامید میباشند . این کار میتواند مقدمه یک ریسک امنیتی برای اجزا مثل توصیف سوکت و اجرا چنین محتویاتی نیز به همین صورت .

۸-۱۲ خواص متغیر از IMCD

هر عنصر و عنصر پالوده از ISO 15836, مجموعه فرا داده هسته دوبلین یا مجموعه هسته فرا داده دوبلین (imcd) میتواند برای توصیف متغیر های سوکت به کار رود . هر کدام از آنها میتوانند چندین بر به صورت زیر عنصری از <variable>(۸-۱) به کار روند .

۹ فرمانها

۹-۱ کلیات

عناصر فرمان برای ضبط فرمانهای استفاده می شوند که کاربر ممکن است برای یک هدف صادر کند. فرمان یک کارکرد اصلی است که کاربر می تواند برای اجرای یک هدف درخواست کرده و نمی توان تنها توسط یک متغیر ارائه شود.

فرمان باید برخی از کارکردهایی فراتر از دستکاری ارزش یک متغیر واحد را پشتیبانی کند.

مثال: دکمه جستجو بر روی دستگاه پخش سی دی و یک دکمه ارسال را در فرم برخط.

عنصر <command> ممکن است به عنوان زیرعنصر <set> (به بند ۷-۱ مراجعه کنید) و یا به عنوان زیرمجموعه از <uiSocket> عمل کند (به بند ۶-۱ مراجعه کنید).

۹-۲ مشخصه های "id"

عنصر <command> (به بند ۹-۱ مراجعه کنید) باید خاصیت "id" را دارا باشد.

مشخصه های "id" باید از نوع ID باشد که توسط طرحواره XML قسمت ۲: نوع داده ها، تعریف شده اند. که شناسانه ای را فراهم می کند که برای اشاره به عنصر در داخل توصیفات سوکت و به عنوان ابزاری برای انقیاد منابع بیرونی تعریف شده مانند برچسبها، استفاده می شود. ارزش "id" باید در میان همه مشخصه های "id" و "name" در توصیفات سوکت منحصر به فرد باشند.

مثال: <command id="seek">

یادآوری- ارزش های "id" به طور معمول برای کاربران ارائه شده و نیازی نیست که برای انسان قابل فهم باشد.

۹-۳ مشخصه 'type'

۹-۳-۱ کلیات

عنصر <command> ممکن است مشخصه 'type' را دارا باشد، نوع فرمان، حالتی که فرمان می تواند داشته باشد را مشخص می کند. حالت، اطلاعاتی درباره آخرین اجرا (درخواست) فرمان را شامل می شود.

مشخصه 'type' ارزش QName را دریافت می‌کند. (به طرح‌واره XML مراجعه کنید) اگر پیشوند فضای نام مشخص نشده باشد، فضای نام <http://openurc.org/ns/uiSocketdesc-2> در نظر گرفته می‌شود.

مثال ۱: ارزش‌های نوع زیر از لحاظ معنایی یکسان می‌باشند: "type="uis:voidCommand" و "type="voidCommand". با فرض اینکه شناسانه فضای نام uiSocket به <http://openurc.org/ns/uiSocketdesc-2> محدود شود.

زیرقسمت‌های زیر سه نوع فرمان را تعریف می‌کند. نوع فرمان به طور پیش فرض باید uis:voidCommand باشد.

uis:voidCommand ۲-۳-۹

type="uis:voidCommand": فرمان باید بدون حالت باشد.

فرمان‌های از نوع uis:voidCommand نباید خروجی و یا ورودی خروجی داشته باشند. **یادآوری-** این محدودیت لازم است از آنجا که یک فرمان با پارامترهای خروجی و یا ورودی / خروجی قادر به اجرا چند نمونه از همان فرمان در یک زمان نیست، در این صورت تنها یک مجموعه‌ای از پارامترهای خروجی است که می‌تواند نتایج را دریافت کند.

uis:basicCommand ۳-۳-۹

type="uis:basicCommand": فرمان مجموعه‌ای از حالت‌های مقابل را دارا می‌باشد: "initial",

"rejected", "inProgress", "done", "succeeded", "failed".

معانی این حالت‌ها به صورت زیر می‌باشد:

- "initial": فرمان برای اجرا درخواست نشده است.

- "rejected": اجرای فرمان درخواست شده ولی هدف درخواست را رد کرده است.

- "inProgress": فرمان در حال حاضر طبق درخواست در حال اجرا می‌باشد.

- "done": فرمان اجرا شده است.

- "succeeded": فرمان با موفقیت اجرا شده است (این مورد از حالت "done" مشخص تر بوده و در صورت فهم توسط هدف ترجیح دارد)

- "failed": فرمان اجرا شده ولی موفقیت آمیز نبوده است (این مورد از حالت "done" مشخص تر بوده و در صورت فهم توسط هدف ترجیح دارد)

یادآوری ۱- در صورت عدم موفقیت فرمان هدف باید به صورت مشخص پیغام خطا را برای کاربر از راه‌های دیگر فراهم کند به عنوان مثال با نشان دادن اعلان.

یادآوری ۲- حالت فرمان اطلاعاتی مبنی بر اجرای دوباره فرمان در اختیار نمی‌گذارد. این کار بر عهده وابستگی <write> می‌باشد (به بند ۳-۹-۹ مراجعه کنید).

یادآوری ۳- فرمان از نوع "uis:basicCommand" نمی‌تواند چندین بار برای هدف فراخوانی شود مگر اجرای قبلی آن به پایان برسد.

مثال : دکمه طبقه همکف که درون آسانسور فشار داده شده مثالی از حالت "inProgress" می‌باشد.

uis:timedCommand ۴-۳-۹

”uis:timedCommand” type= : مانند uis:basicCommand است به جز اینکه حالت ”inProgress” یک

فیلد اضافه به نام ”ttc” دارا می باشد.

فیلد ”ttc” از نوع xsd:duration می باشد.

ارزش ”ttc” ممکن است مشخص نشده باشد.

اگر فرمان در حالت ”inProgress” باشد ارزش ”ttc” به صورت دوره ای کاهش می یابد تا زمانی که فرمان به

حالت دیگری تغییر حالت بدهد. ارزش ”ttc” در صورتی که مشخص شده باشد یک جز-اطلاعات از طرف

هدف بوده که مدت تخمینی اجرای فرمان را مشخص می کند که البته بدون ضمانت از طرف هدف

می باشد. ارزش ”ttc” برای urc فقط خواندنی بوده و در هر حالتی به جز ”inProgress” بی معنی می باشد.

یادآوری ۱- فرمان از نوع ”uis: timedCommand” نمی تواند چندین بار برای هدف فراخوانی شود مگر اجرای قبلی آن به

پایان برسد.

یادآوری ۲- ارزش فیلد ”ttc” را می توان با الحاق ”[ttc]” به xpath در مسیر کارکرد uis:value(path) به دست آورد به بند

۵-۲-۲-۵ مراجعه کنید.

مثال : یک فرمان ”book” در یک سیستم بلیط برخط از نوع uis:timedCommand است و زمان تخمین زده شده برای

فرآیند رزرو در زمان واقعی را نشان می دهد

۴-۹ مشخصه ’sensitive’

عنصر <command> (به بند ۹-۱ مراجعه کنید) می تواند خاصیت ’sensitive’ را دارا باشد که خاصیتی

بولی بوده که به حساس بودن عمل اشاره دارد. ارزش پیش فرض باید ”false” باشد.

مثال : یک فرمان دکمه تماس اضطراری فرم که دارای پیامدهای قانونی خواهد بود با نشانه گذاری پایین توصیف شده است:

```
<command id=’emergency’ sensitive=’true’ />
```

۵-۹ مشخصه ’sufficient’

عنصر <command> (به بند ۹-۱ مراجعه کنید) می تواند شامل خاصیت ’sufficient’ از نوع بولی باشد.

ارزش ”true” نشان می دهد که فرمان به اندازه کافی مشخص شده است. یک فرمان به اندازه کافی مشخص

است اگروتنهاگر مجموعه وابستگی ادعایش، در صورت وجود، مشخص باشد. مجموعه ای از وابستگی ادعا

کافی است اگروتنهاگر تمام عبارات ارزیابی به درست شده باشند و فرمان با موفقیت بر روی هدف به پایان

برسد. وابستگی، ادعا شده است در بند ۹-۹-۵ مشخص شده است.

یادآوری ۱- اگروتنهاگر در جمله بالا باید بصورت اگر و تنها اگر نوشته شود و جملات قبل و بعد از آن به صورت معادل گرفته

می شوند پس اگر یک کدام صحیح باشد ان یکی نیز صحیح است و بالعکس اگر یکی غلط باشد دیگری نیز غلط است.

مثال : برای دکمه فرمان ”طبقه ۴” آسانسور، شرایط بعدی زیر کافی نیست، چون فرمان می تواند ناموفق و ارزش عبارت

هنوز هم می تواند درست باشد:

```
<dependency>
```

```
<assert> uis:value(’currentFloor’) > 0</assert>
```

```
</dependency>
```

با این حال شرایط بعدی زیر می‌تواند مشخص باشد:

```
<dependency>  
<assert> uis:value('currentFloor') eq 4 </assert>  
</dependency>
```

یادآوری ۲- در مورد توصیف کافی از فرمان‌ها، یک URC هوشمند می‌تواند فنون پیشرفته استنتاجی اعمال کند و در نتیجه یک واسط کاربر بهتر ارائه کند. برای مثال، می‌توان نتیجه گرفت که یک فرمان نیازی نیست فراخوانی شود اگر وابستگی ادعا آن در حال حاضر صحیح باشد.

در صورت وجود، مشخصه 'sufficient' از <command> (به بند ۹-۱ مراجعه کنید) مشخصه 'sufficient' از <uiSocket> را تحت الشعاع قرار می‌دهد (به بند ۶-۴ مراجعه کنید)، اما تنها برای آن فرمان خاص است. اگر ارزشی وجود نداشت، ارزش مشخصه 'sufficient' از <uiSocket> به فرمان اعمال می‌شود.

۹-۶ مشخصه 'complete'

عنصر <command> (به بند ۹-۱ مراجعه کنید) می‌تواند شامل مشخصه 'complete' از نوع بولی باشد. ارزش "true" نشان می‌دهد که همه فرمان‌ها در توصیف سوکت به طور کامل با توجه به پارامترهای خود مشخص شده‌اند. فرمان به طور کامل مشخص است اگر تنها اگر ورودی و خروجی مجموعه آن پارامتر کامل باشد. پارامترهای فرمان در بند ۹-۱۲ مشخص شده است.

مجموعه‌ای از پارامترهای ورودی فرمان کامل است اگر تنها اگر نتیجه اجرای موفقیت آمیز این فرمان تضمین شده باشد که با همان مجموعه از ارزش‌های پارامتر ورودی هر زمان اجرا شود، نتیجه یکسان باشد. **یادآوری ۱-** اگر تنها اگر در جمله بالا باید بصورت اگر و تنها اگر نوشته شود و جملات قبل و بعد از آن به صورت معادل گرفته می‌شوند پس اگر یک کدام صحیح باشد ان یکی نیز صحیح است و بالعکس اگر یکی غلط باشد دیگری نیز غلط است.

یادآوری ۲- در مورد توصیف کافی از فرمان‌ها، یک URC هوشمند می‌تواند فنون پیشرفته استنتاجی اعمال کند و در نتیجه یک واسط کاربر بهتر ارائه کند.

در صورت وجود، مشخصه 'complete' از <command> (به بند ۹-۱ مراجعه کنید) مشخصه 'complete' از <uiSocket> را تحت الشعاع قرار می‌دهد (به بند ۶-۵ مراجعه کنید)، اما تنها برای آن فرمان خاص است. اگر ارزشی وجود نداشت، ارزش مشخصه 'complete' از <uiSocket> به فرمان اعمال می‌شود.

۹-۷ مشخصه 'optional'

عنصر <command> (به بند ۹-۱ مراجعه کنید) می‌تواند شامل خاصیت "optional" از نوع بولی باشد. ارزش پیش فرض این متغیر "false" می‌باشد.

ارزش "true" به این معناست که فرمان در حال حاضر به دلیل مختلف در دسترس نمی‌باشد.

یادآوری ۱- نمونه‌های محدودیت که بر دسترس‌پذیری یک عنصر سوکت تاثیر دارند عبارتند از: واپایش دسترسی، محصولات مختلف با استفاده از یک سوکت مشترک اما با کمی تغییر در اجرا (از جمله در DCPS از UPnP).

یادآوری ۲- دسترس‌پذیری یک عنصر سوکت در زمان اجرا، به URC از طریق وسیله مخصوص TUN نشان داده می‌شود.

۸-۹ مشخصه 'dim'

عنصر `<command>` (به بند ۹-۱ مراجعه کنید) دارای مشخصه 'dim' که یک فرمان را به صورت بعددار (با یک یا چند بعد) معرفی می‌کند. در زمان اجرا یک فرمان بعددار باید حالت‌های متعدد و برای هر یک، ترکیب خاصی از نمایه را استفاده کند.

در صورت وجود، مشخصه 'dim' باید شامل فهرستی از ارجاع‌های نوع مرتب شده و جدا شده توسط فاصله باشد که به عنوان نوع نمایه فرمان‌ها عمل می‌کنند. اولین مرجع نوع ارجاع بعد اول را مشخص می‌کند، دومین نوع برای دومین بعد و همین‌طور سومین و غیره. ارجاعات نوع نمایه معتبر شامل: (الف) نام نوع تعریف شده (به بند ۱۱-۵ مراجعه کنید) یا وارد شده (به بند ۱۱-۶ مراجعه کنید) در `<xsd:schema>` به عنوان بخشی از توصیفات سوکت و (ب) نام کامل تایید شده (QName) از یک نوع بیرونی (به بند ۱۱-۷ مراجعه کنید)

نمایه فرمان نباید حاوی ارزش `ttc` باشد که برای نشان دادن فیلد `ttc` اش محفوظ شده است.

یادآوری ۱- مشخصه 'type' مربوط به فرمان‌ها چند بعدی نوع هر ارزش دارای آن فرمان را مشخص می‌کند و نه نوع کلی فرمان را.

یادآوری ۲- هیچ مرتب‌سازی صریحی برای نتایج حالت در فرمان‌ها چند بعدی وجود ندارد با این حال مرتب‌سازی بر اساس ارزش نمایه در هدف توصیه می‌شود برای انتقال حالت به یا از URC، برای مشخص کردن حالت برای کاربر (به استاندارد ISO/IEC 24752-1 مراجعه کنید).

یادآوری ۳- بزرگترین مجموعه ممکن از ارزش‌های برای نتایج فرمان سوکت که شامل تمام نمایه‌های انواع است در زمان رفتن در مسیر از عنصر `<سوکت>` به فرمان سوکت خاص رخ می‌دهد (به تعریف مسیر در بند ۵-۲-۵-۲ مراجعه کنید). با این حال، تمام ترکیبات نمایه در واقع ممکن است در زمان اجرا رخ دهند.

یادآوری ۴- نوع نمایه ممکن است یک تعداد نامحدود از ارزش نمایه را مشخص کند (به عنوان مثال `xsd:integer` بی نهایت ارزش‌های متفاوت می‌گیرد). با این حال، ارزش نمایه واقعی زیر مجموعه متناهی از ارزش‌های مجاز توسط نوع نمایه در زمان اجرا هستند.

یادآوری ۵- استفاده از یک فرمان بعددار تنها زمانی معقول و منطقی است که مجموعه‌ای از فرمان‌ها واقعا همگن را دارا باشیم، یعنی هر فرمان دارای مشخصه‌های یکسان از جمله برجسب (پیام) و متن کمک باشد.

یادآوری ۶- از فرمان به عنوان مثال، `http://example.com/thermometer/socket#reset` URI به هر مراجعه

تک جزئی فرمان در فرمان بعدی با `'id'=reset` انجام می‌شود. با این حال، HTTP URI:

`//example.com/thermometer/socket#dims(reset)` برای دیدن جزئیات برای فرمان کلی با تمام قطعات مراجعه کنید

ISO / IEC 24752-5

یادآوری ۷- ترتیبی، که در آن مولفه‌های یک فرمان بعددار به کاربر ارائه شده است، بر اساس ترتیب تعریف‌شده از انواع نمایه مربوط است. برای جزئیات بیشتر به استاندارد ISO / IEC 24752-1 مراجعه کنید.

۹-۹ وابستگی‌های فرمان

۹-۹-۱ کلیات

وابستگی‌های فرمان اشاره دارد به زمانی که یک فرمان توسط کاربر اجرا می‌شود و اینکه تاثیرات آن فرمان چه می‌باشد (شرایط بعدی)

وابستگی فرمان توسط نشانگر فرمان به صورت زیر نشان داده می‌شود:

<dependency>

<relevant>expr1</relevant>

<write>expr2</write>

</dependency>

عنصر <dependency> یک بار و به عنوان زیرعنصری از <command> ظاهر می‌شود.

۹-۹-۲ وابستگی <relevant>

وابستگی <relevant> هنگامی که یک فرمان به کاربر معرفی می‌شود کاربرد دارد، به عنوان مثال چقدر بر اساس حالت خاصی از هدف مربوط است، اگر این وابستگی صحیح باشد فرمان به کاربر معرفی می‌شود البته این به این معنی نیست که مستقیماً از طریق واسط کاربر مشتق شده از سوکت نشان داده شود بلکه به صورت مسیر راهنما که کاربر قادر به یافتنش است انجام می‌شود.

عنصر <relevant> می‌تواند به عنوان زیرعنصری از <dependency> عمل کند (به بند ۹-۹-۱ مراجعه

کنید) اگر موجود باشد فقط یک بار واقع می‌شود و در صورت عدم وجود آن را از نزدیک‌ترین عنصر <set> واقع در والد خود که دارای این عنصر است به ارث می‌برد.

مثال: فرمانی که برای رزرو کردن به کار می‌رود فقط زمانی مرتبط است که کاربر رزرو کردن را تعریف کرده باشد این

وابستگی توسط عنصر <relevant> به صورت زیر تعریف می‌شود:

```
<command id="makeReservation" type="uis:basicCommand">
```

```
<dependency>
```

```
<relevant> uis:value('activity') eq 'makingReservation' </relevant>
```

```
</dependency>
```

```
</command>
```

محتوای عنصر <relevant> باید برای وابستگی متغیر <relevant> مشخص شود (به بند ۸-۹-۲ مراجعه

کنید)

اگر هیچ وابستگی <relevant> تعریف نشده باشد (نه در فرمان و یا والد هایش) و یا زمانی که خالی باشد،

وابستگی <relevant> در فرمان باید صحیح باشد حتی اگر عنصر <notify> فعال باشد.

۹-۹-۳ وابستگی <write>

این وابستگی مشخص می‌کند که آیا کاربر می‌تواند فرمان را فعال سازد یا خیر. که به ان وابستگی <write>

یا پیش شرط گویند، اگر پیش شرط غلط باشد امکان اجرا عمل فراهم نیست در غیر این صورت عمل

می‌تواند اجرا شود. ولی با این حال هیچ تضمینی برای موفقیت آن وجود ندارد.

عنصر <write> می‌تواند به عنوان یک زیرعنصر از <dependency> عمل کند (به بند ۹-۹-۱ مراجعه کنید) اگر موجود باشد فقط یک بار واقع می‌شود و اگر موجود نباشد فرمان وابستگی <write> را از نزدیک‌ترین والد <set> خود که شامل وابستگی <write> می‌باشد به ارث می‌برد (به بند ۸-۹-۳ مراجعه کنید).
مثال: دکمه آسانسور برای رفتن به طبقه ۴ فقط در صورتی کار می‌کند که آسانسور در همان طبقه نباشد. این وابستگی به صورت زیر تعریف می‌شود:

```
<dependency>  
<write> uis:value('currentFloor') ne 4 </write>  
</dependency>
```

محتوای عنصر <write> باید برای وابستگی متغیر <write> مشخص شود (به بند ۹-۹-۱ مراجعه کنید). اگر وابستگی <write> فرمان، خالی بود یا یک <write> خالی را از والد <write> به ارث برده باشد، وابستگی فرمان نامعلوم می‌باشد.

یادآوری - مقدمه ارزش‌های نامشخص مشابه است با منطق بولی ۳ ارزشی که به ما اجازه داشتن عدم اطمینان را می‌دهد. اگر وابستگی <write> مشخص نباشد URC هیچ راهی به جز امتحان کردن اینکه فرمان اجرا می‌شود یا نه را ندارد.

اگر هیچ وابستگی <write> مشخص نشده باشد (نه در فرمان و نه در والد) وابستگی <write> فرمان همیشه صحیح می‌باشد حتی زمانی که عنصر <notify> فعال است.

۹-۹-۴ وابستگی <insert>

وابستگی <insert> مشخص می‌کند که آیا کاربر می‌تواند ترکیبی از نمایه‌های مرتبط را در زمان اجرا تغییر دهد (که مجموعه مرتبط به صورت نمایه‌هایی توسط مشخصه 'dim' بر روی فرمان مرتبط عمل می‌کند، به بند ۹-۱ مراجعه کنید).

محتویات وابستگی <insert> می‌توانند به عنوان یک عبارت XPATH برای ارزیابی یک ارزش بولی که می‌تواند در طول نشست تغییر کند، عمل کنند.

اگر ارزش آن غلط باشد نمی‌توان ترکیب نمایه را اضافه و یا کم نمود با این حال می‌توان امتحان کرد که آیا اضافه و یا کم می‌شود ولی هیچ تضمینی نیست که این عمل موفقیت‌آمیز باشد.

یادآوری ۱- اضافه کردن ترکیب نمایه به معنی اضافه کردن مولفه فرمان برای ترکیب نمایه مشخص در فضای نمایه است که توسط مشخصه 'dim' مربوطه مشخص می‌شود. توجه داشته باشید که وضعیت مولفه فرمان اضافه شده توسط هدف اتخاذ می‌شود، حذف کردن ترکیب نمایه به معنی حذف مولفه فرمان برای ترکیب نمایه مشخص در حال حاضر است.

عنصر <insert> می‌تواند به عنوان زیرعنصر از <dependency> عمل کند (به بند ۹-۹-۱ مراجعه کنید) اگر فرمان مربوطه مشخصه 'dim' را دارا باشد (به بند ۹-۸ مراجعه کنید). اگر موجود بود فقط یک بار می‌تواند اتفاق بیفتد.

ارزش پیش فرض آن باید "false()" باشد.

یادآوری ۲- هیچ ارث بری توسط وابستگی <insert> از عنصر 'set' وجود ندارد هر عنصر <insert> تنها به دسته‌ای از نمایه‌ها که توسط عنصر 'dim' در همان سطح عمل می‌کند، مربوط می‌شود.

۵-۹-۹ وابستگی <assert>

وابستگی <assert> یک عبارت بولی را که پس از انجام موفقیت‌آمیز فرمان به صحیح تغییر می‌کند را معرفی می‌کند که به آن تایید یا پس‌اشراط می‌گویند.

یادآوری ۱- اگر وابستگی <assert> برای تضمین کافی باشد می‌توان از پس‌اشراط صحیح اینطور برداشت نمود که فرمان موفق بوده است، برای دیدن جزئیات به بند ۵-۹ مراجعه کنید.

برای فرمان‌ها از نوع `uis:voidCommand` (به بند ۲-۳-۹ مراجعه کنید) وضعیت نهایی برای هر درخواست فرمان حتما صحیح می‌باشد برای فرمان‌ها از نوع `uis:basicCommand` یا `uis:timedCommand` وضعیت نهایی برای تمام درخواست‌های فرمان که جواب آن‌ها در وضعیت موفق می‌باشد حتما صحیح است. عنصر <assert> می‌تواند به عنوان زیرعنصری از <dependency> عمل کند (به بند ۱-۹-۹ مراجعه کنید) در این صورت فقط یک بار اتفاق می‌افتد.

مثال ۱: بعد از انجام موفقیت‌آمیز رفتن به طبقه ۴ در آسانسور ارزش متغیر طبقه فعلی به ۴ تغییر می‌کند:

```
<dependency>
<assert uis:value('currentFloor') eq 4 </assert>
</dependency>
```

مثال ۲: این فرمان شروع پخش باعث درخواست URC برای باز کردن یک زیرنشست توسط هدف می‌شود:

```
<command id="startPlay" type="uis:voidCommand">
<dependency>
<assert uis:sessionForward("spawn", "http://example.com/target/play") </assert>
</dependency>
</command>
```

یادآوری ۲- URC‌های پیشرفته می‌توانند از دانش موجود در وضعیت نهایی برای تولید واسط کاربر کارتر استفاده کنند. وضعیت نهایی مخصوصا زمانی که ارزش آن به اندازه تنظیم شده باشد کارا است.

محتوی <assert> می‌تواند خالی یا یک عبارت صحیح XPath برای ارزیابی یک ارزش بولی باشد. اگر وابستگی <assert> فرمان خالی باشد وابستگی فرمان نا مشخص است. اگر وابستگی <assert> مشخص نشده باشد، وابستگی <assert> فرمان همیشه صحیح می‌باشد.

۱۰-۹ اطلاعات نقشه مخصوص بستر برای فرمان‌ها

برای داشتن اطلاعات نقشه مخصوص بستر فرمان‌ها عنصر <mapping> می‌تواند به هر ارزشی به عنوان زیرعنصری از <command> به کار رود (به بند ۱-۹ مراجعه کنید).

عنصر <mapping> باید دارای مشخصه 'platform' باشد که ارزش آن توسط استاندارد ISO/IEC 24752 محدود نشده باشد.

عنصر <mapping> می‌تواند مقدار متفاوتی از عناصر و یا زیرعنصرها را شامل باشد که زیرعنصرها می‌توانند شامل فضاهای نام متفاوت از uis باشند.

یادآوری ۱- توصیفات سوکتی که دارای اطلاعات مخصوص به مخصوص بستر باشد خاصیت بیطرفی خود را از دست می‌دهد. در هر حال چندین نگاشت می‌تواند در توصیفات سوکت موجود باشد (هر کدام برای یک بستر) توصیه می‌شود دیگر فنون

مربوط به اتصال به مخصوص بستر مطالعه شود. به عنوان مثال نگاشت می‌تواند در یک پوشه بیرونی با ارجاع به آن در توصیفات سوکت باشد.

یاداوری ۲- صاحبان بستر و فروشندگان از استفاده عنصر <mapping> برای بارگذاری محتویات فعال یا اجرایی در توصیفات سوکت کاملاً دلسرد شده‌اند، این کار باعث افزایش مخاطره امنیتی برای اجزا مشتق شده مثل توصیف سوکت و یا اجرا شدنی مانند محتویات می‌شود.

۹-۱۱ خصیصه‌های فرمان از DCMI

هر عنصر یا عنصر بهبود از مجموعه عنصر DCMI یا مجموعه‌ای از اصطلاحات فراداده DCMI، در صورت امکان برای توضیح فرمان سوکت به کار برده می‌شود.

۹-۱۲ پارامترهای فرمان

۹-۱-۱۲ کلیات

عنصر <command> می‌تواند هر تعداد از زیرعنصر <param> را دارا باشد (به بند ۹-۱ مراجعه کنید) که هر کدام پارامتر مشخص مربوط به فرمان را توصیف می‌کنند.
یاداوری ۱- محدوده پارامتر فرمان می‌تواند هم محدود به فرمان و هم عمومی باشد (در سوکت).

مثال: فرمان برای آسانسور شامل ۳ پارامتر است؛ طبقه مبدا، طبقه مقصد، و زمان باز ماندن در هنگامی که کسی وارد نمی‌شود. دو پارامتر اولی محلی می‌باشند چرا که در هر طرف تغییر می‌کنند ولی زمان انتظار، یک پارامتر عمومی است که می‌تواند برای هر درخواست عمل کند.

```
<command id="callElevator" type="uis:basicCommand">  
<param id="originFloor" dir="in" type="xsd:integer" />  
<param id="destinationFloor" dir="in" type="xsd:integer" />  
<param idref="waitTime" dir="in" />  
</command>
```

یاداوری ۲- URC‌های پیشرفته می‌توانند از اطلاعات پارامترهای فرمان برای به وجود آوردن واسط کاربر کارتر استفاده کنند، این اطلاعات مخصوصاً اگر تضمین شود که کاملاً مشخص شده باشد بدرد بخور می‌باشند.

۹-۱۲-۲ مشخصه 'id' (پارامتر محلی)

عنصر <param> می‌تواند یا یک مشخصه 'id' و یا یک 'idref' داشته باشد.
عنصر <param> به همراه مشخصه 'id' به نام پارامتر محلی شناخته می‌شود.
ارزش پارامتر محلی برای کاربر در دسترس است فقط اگر وابستگی فرمان <relevant> به صحیح ارزیابی شود.

ارزش مشخصه 'id' باید در میان تمام مشخصه‌های 'id' موجود در توصیف سوکت یکتا باشد.
مشخصه 'id' باید از نوع ID که توسط طرح‌واره XML فصل یازدهم قسمت ۲ تعریف شده است، یک شناسانه را برای ارجاع به عنصر در توصیف سوکت و به عنوان وسیله انقیاد بیرونی منبع تعریف شده مانند برچسب را ارایه می‌دهد. مشخصه 'id' برای کاربر قابل ارائه نیست و نیازی نیست برای انسان قابل درک باشد.

یادآوری - در احضار فرمان، URC ارزش‌های تمام پارامترهای ورودی (شامل پارامترهایی با مشخصه‌های 'in' یا 'inout') را به هدف ارسال می‌کند. بعد از اجرا هدف همه ارزش‌های پارامترهای خروجی محلی را به URC بازمی‌گرداند (پارامترهای محلی با ارزش‌های "out" و "inout").

۹-۱۲-۳ مشخصه 'idref' (پارامتر عمومی)

عنصر <param> می‌تواند با مشخصه 'id' یا 'idref' مشخص شود (به بند ۹-۱۲-۱ مراجعه کنید) عنصر <param> و مشخصه 'idref' را پارامتر عمومی می‌نامند.

مشخصه 'idref' باید ارجاع به متغیر یا اعلان در همان سوکت داشته باشد. ارزش مشخصه 'idref' باید ارزش مشخصه 'id' عنصر <variable> یا <notify> در توصیف سوکت مربوط باشد، به عنوان یک استثنا مشخصه 'idref' می‌تواند به مجموعه‌ای از سوکت با 'id' یکسان متصل شود به این معنی که تمام متغیرها موجود در مجموعه به صورت پارامتر عمومی فرمان، استفاده می‌شوند.

یادآوری ۱- ارجاع به پارامتری به صورت عمومی "out" به این معنی است که اجرای فرمان بر حالت اعلان تاثیر می‌گذارد.

پارامترهای محلی دیگر فرمان‌ها توسط مشخصه 'idref' ارجاع نمی‌شوند.

مشخصه 'idref' باید از نوع IDREF که توسط طرح‌واره XML فصل یازدهم قسمت ۲ تعریف شده است، که به وسیله آن قیدی میان فرمان ارسال شده و متغیرهای موجود در آن برقرار می‌شود، اگر ارزش مشخصه 'dir' (به بند ۹-۱۲-۴ مراجعه کنید) شامل "in" بود هدف از ارزش کنونی متغیر مرجع به عنوان ارزش ورودی اجرای فرمان استفاده می‌کند. اگر ارزش id برابر با "out" باشد هدف بعد از اجرای فرمان ارزش‌های را بروز کرده تا تاثیرات جواب در آن دیده شود. اگر ارزش فرمان 'dir' برابر "inout" باشد هدف قبل از اجرا ارزش‌ها را از مرجع خوانده و بعد از اجرا آن‌ها را به روز می‌کند.

مشخصه 'idref' برای کاربر قابل ارائه نیست و نیازی نیست برای انسان قابل درک باشد.

پارامترهای عمومی نباید دارای مشخصه 'type' باشند (به بند ۹-۱۲-۴-۴ مراجعه کنید) بجای آن نوع آن‌ها از طریق ارجاع به تعریف متغیر سوکت مشخص می‌شود (به بند ۸-۳ مراجعه کنید).

یادآوری ۲- در احضار فرمان، URC ارزش‌های پارامترهای عمومی را دریافت یا ارسال نمی‌کند. در عوض مقادیرشان بین URC و

یادآوری ۳- ارجاع به پارامتری به صورت عمومی "out" به این معنی است که اجرای فرمان بر حالت اعلان تاثیر می‌گذارد.

۹-۱۲-۴ مشخصه 'dir'

۹-۱۲-۴-۱ کلیات

عنصر <param> باید مشخصه 'dir' را دارا باشد (به بند ۹-۱۲-۱ مراجعه کنید)

ارزش‌های آن باید یکی از "in"، "out" و یا "inout" باشد. ارزش "in" نشان می‌دهد پارامتر از نوع ورودی است، ارزش "out" نشانگر پارامتر خروجی و ارزش "inout" نشانگر این است که پارامتر هم از نوع ورودی و هم خروجی می‌باشد، به این معنی که پارامتر قبل از اجرا خوانده شده و بعد از آن نیز به روز رسانی می‌شود.

یادآوری - فرمان‌ها از نوع uis:voidCommand نمی‌توانند پارامتری از نوع خروجی و یا ورودی خروجی داشته باشند.

۹-۱۲-۴-۲ پارامترهای ورودی

در مشخصه 'dir' (به بند ۹-۱۲-۴-۱ مراجعه کنید) ارزش "in" نشانگر پارامتر ورودی می‌باشد که پارامتری است که ارزش آن قبل از اجرا توسط هدف برای اثر گذاری به روی اجرا و نتایج خوانده می‌شود. فرمان می‌تواند هر اندازه‌ای از پارامترهای عمومی یا محلی ورودی را داشته باشد. ارزش پارامتر ورودی توسط URC در صورت وجود وابستگی <write> با ارزش درست یا نامشخص می‌تواند تغییر کند.

برای پارامترهای ورودی عمومی وابستگی‌های <relevant> و <write> مشخص می‌کنند که آیا ارزش‌های پارامترها می‌تواند توسط URC و یا کاربران آن تغییر کند یا خیر (به بند ۸-۹-۲ مراجعه کنید). **یادآوری** - پارامترهای ورودی محلی مشابه عنصر سوکت می‌باشند ولی ارزش آنها با هدف، فقط هنگام فراخوانی فرمان و فقط از طرف URC به هدف همگام‌سازی می‌شود.

۹-۱۲-۴-۳ پارامترهای خروجی

در مشخصه 'dir' (به بند ۹-۱۲-۴-۱ مراجعه کنید) ارزش "out" اشاره به ارزش‌های خروجی دارد که ارزش آنها توسط هدف بعد از اجرا به روز رسانی می‌شود تا بر روی نتایج تاثیر داشته باشد. فرمان می‌تواند هر اندازه‌ای از پارامترهای عمومی یا محلی خروجی را داشته باشد. ارزش پارامترهای محلی خروجی نباید توسط URC یا کاربران آن تغییر کند. برای پارامترهای خروجی عمومی وابستگی‌های <relevant> و <write> (به بند ۸-۹-۲ مراجعه کنید) نشانگر اجازه دسترسی یا تغییر توسط URC و کاربرانش می‌باشد. **یادآوری** - پارامترهای خروجی محلی مانند عناصر سوکت می‌باشند، ولی ارزش‌های آنها با هدف تنها زمانی همگام سازی می‌شود که اجرای فرمان پایان یافته و این کار فقط از هدف به سمت URC امکان پذیر می‌باشد.

۹-۱۲-۴-۴ پارامترهای ورودی-خروجی

در مشخصه 'dir' ارزش "inout" نشانگر پارامتر ورودی-خروجی می‌باشد که به عنوان پارامترهای ورودی و خروجی از فرمان یکسانی به کار برده می‌شوند. فرمان می‌تواند هر اندازه‌ای از پارامترهای ورودی-خروجی محلی یا عمومی داشته باشد. ارزش پارامتر محلی ورودی-خروجی می‌تواند در صورت وجود و داشتن ارزش صحیح وابستگی <write> توسط URC و یا کاربرش تغییر کند.

برای پارامترهای عمومی ورودی-خروجی وابستگی‌های <relevant> و <write> مشخص می‌کنند که ارزش این پارامتر قابل دستیابی و تغییر توسط URC و کاربرانش می‌باشد یا خیر (به بند ۸-۹-۳ مراجعه کنید). **یادآوری ۱**- پارامترهای محلی ورودی خروجی مانند عناصر سوکت می‌باشند، ولی ارزش آنها با هدف در این دو حالت همگام‌سازی می‌شود: (۱) زمانی که فرمان از طرف URC به سمت هدف درخواست می‌شود، و (۲) بعد از کامل شدن اجرای فرمان از طرف هدف به URC.

۹-۱۲-۵ مشخصه 'type'

یک پارامتر محلی باید دارای ویژگی 'type' باشد (به بند ۹-۱۲-۲ مراجعه کنید). پارامترهای عمومی نباید دارای مشخصه 'type' باشند.

ارزش مشخصه 'type' نشانگر نوع پارامتر می‌باشد. انواع معتبر مانند انواع عنصر <variable> می‌باشند (به بند ۸-۳ مراجعه کنید).

۹-۱۲-۶ مشخصه 'secret'

پارامترهای محلی (به بند ۹-۱۲-۲ مراجعه کنید) می‌توانند دارای مشخصه 'secret' باشند، ارزش پیش فرض آن برابر غلط می‌باشد.

پارامترهای عمومی (به بند ۹-۱۲-۳ مراجعه کنید) نمی‌توانند دارای ویژگی 'secret' باشند. مشخصه 'secret' باید همان معنی را برای عنصر <variable> داشته باشد (به بند ۹-۱۲-۱ مراجعه کنید).

۹-۱۲-۷ مشخصه 'sensitive'

پارامترهای محلی (به بند ۹-۱۲-۲ مراجعه کنید) می‌توانند دارای مشخصه 'sensitive' باشند و ارزش پیش فرض آن باید غلط باشد.

پارامترهای عمومی (به بند ۹-۱۲-۳ مراجعه کنید) نباید دارای مشخصه 'sensitive' باشند. مشخصه 'sensitive' باید همان معنی عنصر <variable> را داشته باشد.

۹-۱۲-۸ زیرعنصر <selection>

عنصر <selection> می‌تواند به عنوان زیرعنصر از <param> عمل کند (۹-۱۲-۱) ولی فقط برای ورودی‌های محلی (به بند ۹-۱۲-۴-۲ مراجعه کنید) و پارامترهای محلی ورودی-خروجی (به بند ۹-۱۲-۴-۴ مراجعه کنید) اگر موجود باشد فقط یک‌بار واقع می‌شود. عنصر <selection> مجموعه‌ای از ارزش‌های برای محدود کردن فضای ارزش‌های پارامتر را فراهم می‌کند. (انتخاب بسته) و یا ارزش‌های پیشنهادی برای ورود کاربر فراهم می‌کند (انتخاب باز)، و از همان گرامر مورد استفاده برای تعریف متغیرها بهره می‌گیرد (به بند ۸-۱۰ مراجعه کنید).

مثال: یک دستگاه پردازش پوشه‌های چند رسانه‌ای به کاربر اجازه انتخاب تنظیمات از چند تنظیم پیش فرض را می‌دهد. زمانی که فرمان "selectPreset" فراخوانی می‌شود کاربر می‌تواند از یکی از متغیرهای موجود در فهرست "presetNameList" دریافت کند.

```
<variable id="presetNameList" use="required" type="uis:csvlist">
<dc:description>This variable contains a comma-separated list of valid preset names
currently supported by this device. Its value changes if/when the device changes the set
of presets that it supports. This may occur in conjunction with a vendor-defined action or
some other non-UPnP event. This state variable will include any of the predefined presets
that are supported by the device.</dc:description>
</variable>
<command id="selectPreset" use="required" type="uis:basicCommand">
<dc:description>Set a preset configuration.</dc:description>
<param id="presetName" type="xsd:string" dir="in">
<selection closed="true">
<selectionSetDynamic id="availablePresets" varRef="presetNameList"/>
</selection>
</param>
</command>
```

۹-۱۲-۹ نگاهت اطلاعات مخصوص بستر برای پارامترهای فرمان

برای داشتن نگاهت اطلاعات مخصوص بستر عنصر <mapping> می‌تواند هر چند بار به عنوان زیرعنصر از <param> (به بند ۹-۱۲-۱ مراجعه کنید) برای پارامترهای محلی (به بند ۹-۱۲-۲ مراجعه کنید) به کار رود. برای پارامترهای عمومی نمی‌توان استفاده نمود (۹-۱۲-۳)

عنصر <mapping> باید مشخصه 'platform' را دارا باشد که ارزش آن توسط استاندارد ISO/IEC 24752 محدود نشده باشد.

عنصر <mapping> می‌تواند زیرعنصر و عنصرهای متفاوتی داشته باشد، با این حال فضای نام زیرعنصرها باید با فضای نام uis متفاوت باشد.

یادآوری ۱- توصیفات سوکتی که دارای اطلاعات نگاهت مخصوص بستر باشد بی طرفی خود را از دست می‌دهد. با این حال متن نگاهت‌های متعددی را در توصیفات سوکت قرار داد (هر کدام برای یک بستر) توصیه می‌شود دیگر تکنیک‌های اتصال به فناوری‌های مخصوص بستر در نظر گرفته شود. به عنوان مثال اطلاعات نگاهت می‌تواند در یک پوشه بیرونی با ارجاع به توصیفات سوکت مزبور استفاده شود.

یادآوری ۲- فروشندگان و بسترها از استفاده کردن از عنصر <mapping> برای قرار دادن محتویات فعال یا اجرایی در توصیفات سوکت دلسرد شده‌اند. این می‌تواند باعث به وجود آمدن مخاطره امنیتی برای اجزا مانند توصیفات سوکت و اجرا همچنین محتوایی شود.

۹-۱۲-۱۰ مشخصه‌های پارامتر فرمان از DCMI

همه عنصرها و عناصر مشتق شده از استاندارد ISO 15836، فرا داده DC، مجموعه عناصر و یا مجموعه فراداده DCMI شرایط فرا داده ممکن است برای توصیف پارامترهای فرمان به کار روند، در این صورت هر کدام از آن‌ها می‌توانند چندین بار داخل عنصر <param> واقع شوند (به بند ۹-۱۲-۱ مراجعه کنید)

۱۰ اعلان‌ها

۱-۱۰ کلیات

اعلان‌ها حالت‌های خاصی هستند که در آن‌ها، به مانند حالت‌های استثنا، عملیات معمول معلق می‌شود. هدف ممکن است در هر زمان اعلان را فراخوانی (فعال شدن) یا اخراج (غیر فعال) کند. **مثال:** مثال‌هایی از اعلان‌ها شامل یک اعلام توسط سامانه آدرس عمومی در فرودگاه، یک زنگ ساعت، یا پاسخ به ورودی نامعتبر برای یک فیلد از یک فرم می‌شوند.

عنصر <notify> ممکن است یک یا چند بار به عنوان زیرعنصر <>set (۷.۱ را ببینید) یا زیرعنصر <uiSocket> (۶.۱ را ببینید) رخ دهد.

یک عنصر <notify> باید یکی از این سه حالت را داشته باشد: "فعال"، "غیر فعال"، یا "در پشته قرار گرفته". حالت پیش فرض "غیر فعال" است. هدف ممکن است عنصر <notify> را در هر زمان فعال یا غیر فعال سازد. برخی از متغیرها و دستورات ممکن است همانگونه که در وابستگی‌های <write> و <relevant> شان بیان شده، تنها زمانی که اعلان فعال است قابل خواندن و قابل نوشتن باشند (۸.۹.۲).

۸.۹.۳، ۹.۹.۲ و ۹.۹.۳ را ببینید).

یک هدف ممکن است یک اعلان را در حالی که اعلان دیگری در حال حاضر فعال است، فعال سازد. در این مورد، اعلان دوم نسبت به اولی اولویت می یابد. عامل کاربر باید یک پشته از اعلان ها به صورت جدید ترین در بالا را نگه دارد. تمام اعلان های فعال که در راس نمی باشند در حالت "در پشته قرار گرفته" تلقی می شوند (این امر نسبت به عامل کاربر داخلی می باشد و با هدف به اشتراک گذاشته نمی شود).

واسط های کاربر باید بر اساس یک سوکت واسط کاربر به ترتیبی که اعلان ها در آن فعال می شوند احترام بگذارند، به شکلی که تازه ترین اعلان فعال شده همیشه همانی است که به کاربر ارائه می شود. زمانی که بالاترین اعلان غیر فعال می شود، اعلان بعدی در پشته دوباره فعال می شود و به کاربر ارائه می گردد. بیشتر اعلان ها نیاز به تایید از جانب کاربر دارند (بسته به مشخصه "type"، ۱۰.۳ را ببینید).

یادآوری - عنصر `<notify>` همراه با یک پیام متنی برای نمایش به کاربر نمی آید. در عوض، پیامی که باید نمایش داده شود، به گونه ای که در اوراق منابع تعریف شده است (ISO/IEC 24752-5 را ببینید)، توسط منابع تفکیک ناپذیر ارائه می شود (به شکل متن، تصویر یا ویدئو).

۱۰-۲ مشخصه "id"

یک عنصر `<notify>` (به بند ۱۰-۱ مراجعه کنید) باید یک مشخصه "id" داشته باشد.

مشخصه "id" باید همانگونه که در "طرحواره XML قسمت ۲: انواع داده" تعریف شده است از نوع ID باشد. این امر یک شناسانه فراهم می کند که در ارجاع به عنصر از درون توضیحات سوکت و به عنوان ابزاری برای مقید ساختن منابعی که به صورت خارجی تعریف شده اند مانند برچسب ها، به کار می رود. مقدار "id" باید در بین همه مشخصه های "id" و "name" درون توضیحات سوکت، منحصر به فرد باشد. **یادآوری**: مقادیر مشخصه "id" به صورت معمول به کاربران ارائه نمی شوند و نیاز ندارند که قابل فهم برای انسان باشند.

مثال: `< notify id="fireAlarmWarning">`

۱۰-۳ مشخصه "نوع"

یک عنصر `<notify>` (به بند ۱۰-۱ مراجعه کنید) ممکن است یک مشخصه "type" داشته باشد، که مشخص کننده یک نوع گفت و گو از پیش تعریف شده می باشد.

یادآوری ۱ - مشخصه "type" سازوکاری مناسب برای یک توسعه دهنده سوکت است تا در جایی که مناسب است، اشکال درونساز گفتگوی استاندارد را به کار برد. انواع از پیش تعریف شده باید برای بسیاری از موارد کاربری بسنده باشند.

یادآوری ۲ - به همانگونه که برای همه اعلان ها، برچسب ها، و دیگر مولفه های واسط کاربری وابسته به زبان، اعلان هایی که از نوع گفت و گوی از پیش تعریف شده استفاده می کنند می توانند به عنوان منابع تفکیک ناپذیر در اوراق منابع مشخص نموده شوند (ISO / IEC 24752-5 را ببینید).

در صورت وجود، مقدار مشخصه "type"، باید یکی از مقادیر زیر را داشته باشد:

- "show": یک اعلان باید به کاربر ارائه شود. کاربر نیازی ندارد که دریافت اعلان را تایید کند (برای مثال هیچ گفت و گوی معینی مورد نیاز نیست).

- "confirm": اعلان باید به کاربر ارائه شود، و کاربر نیاز است که دریافت اعلان را تایید کند.
یادآوری ۳- در یک واسط کاربر بصری، این اعلان می‌تواند به عنوان یک گفت و گو معین نمایش داده شود، با نشان دادن یک پیام و دکمه "OK".

- "confirmCancel": همان "confirm"، اما کاربر می‌تواند بین تایید و لغو را انتخاب کنید؛
یادآوری ۴- در یک واسط کاربر بصری، این اعلان می‌تواند به عنوان یک گفت و گو معین نمایش داده شود، با نشان دادن یک پیام و دو دکمه برای "OK" و "Cancel".

- "yesNo": اعلان باید به کاربر ارائه شود و کاربر نیاز است در با "yes" یا "no" به اعلان پاسخ دهد؛
یادآوری ۵- در یک واسط کاربر بصری، این اعلان می‌تواند به عنوان یک گفت و گو معین نمایش داده شود، با نشان دادن یک پیام و دو دکمه برای "yes" و "no".

مثال ۱: کد زیر اعلانی را نشان می‌دهد که کاربر باید با "yes" یا "no" به آن پاسخ دهد.
`<notify id="giftwrap" type="yesNo" />`

- "yesNoCancel": همان "yesNo"، اما کاربر همچنین می‌تواند لغو را انتخاب کند؛
- "option": یک اعلان باید به کاربر ارائه شود و کاربر نیاز است یکی از گزینه های متعدد را انتخاب کند.
مجموعه ای از گزینه باید در یک عنصر `<selection>` تو در تو مشخص شود (به بند ۸-۱۰ مراجعه کنید).
`<selection>` ممکن است شامل یک یا بیشتر مجموعه انتخاب باز و بسته باشد (به بند ۸-۱۰-۳ مراجعه کنید).

یادآوری ۶- در یک واسط کاربر بصری، این اعلان می‌تواند به عنوان یک گفت و گو معین نمایش داده شود، با نشان دادن یک پیام و دکمه های مختلف، یکی برای هر گزینه در دسترس.

یادآوری ۷- برچسب برای گزینه ها می‌تواند با اختصاص منابع تفکیک ناپذیر برای مقادیر پیوسته انواع سوکت و/ یا متغیرهای سوکت مشخص شود، همانطور که در عنصر `<selection>` تو در تو اشاره شده است.

مثال ۲: کد زیر نشان دهنده اعلانی است که اجازه می‌دهد کاربر بین مقادیر شمردنی از نوع داخلی سوکت "problemType"، انتخاب کند (برای مثال: می‌تواند مقادیر شمارش زیر را داشته باشد: "proceed"، "cancel"، "call help").

```
<notify id="problem" type="option">
<selection>
<selectionSetStatic id="problemSelectionSet" typeRef="problemType" />
</selection>
</notify>
```

- "optionCancel": همان "option" است اما کاربر همچنین می‌تواند لغو را انتخاب کند؛
- "custom" (مقدار پیش فرض): نوع سفارشی. در این مورد، عنصر `<notify>` (به بند ۱۰-۱ مراجعه کنید) باید زیرعنصرهایی داشته باشد تا مشخص سازد چه ورودی از کاربر مورد نیاز است (به بند ۱۰-۱ مراجعه کنید). پس از اینکه کاربر ورودی را فراهم کرد، باید اعلان را تایید کند؛
- "customCancel": همان "custom" است اما کاربر همچنین می‌تواند لغو را انتخاب کند.

یادآوری ۸- در یک واسط کاربر گرافیکی، کاربر، یک یا چندین مولفه ورودی (برای مثال: دکمه های رادیویی، چک باکس، فیلد های ورودی)، و دکمه "OK" را می بیند. آن ها دکمه "OK" را پس از آنکه بررسی کنند که همه مقادیر به شکل دلخواه مقداردهی شده اند فشار می دهند.

۱۰-۴ مشخصه "category"

یک عنصر `<notify>` (به بند ۱۰-۱ مراجعه کنید) مجاز است که یک مشخصه "category" داشته باشد. رده یک عنصر `<notify>` ماهیت اعلان را توضیح می دهد.

یادآوری ۱- یک URC می تواند از اطلاعات رده بپای ا ارائه یک اعلان در روش های مختلف استفاده کند. برای مثال می توان آیکون های مختلف را برای نشان دادن سطح اضطرار کاربر به کار برد.

یادآوری ۲- رده متفاوت از نوع یک اعلان است. در واقع، هر ترکیبی از نوع ها و رده ها امکان پذیر است.

مشخصه "category" باید یکی از مقادیر زیر را داشته باشد:

- "info": هدف در حال تامین اطلاعات برای کاربر است؛

- "alert": هدف در حال هشدار یک وضعیت به کاربر است.

- "error": هدف یک خطا را تشخیص داده است.

مثال ۱: برای اطلاع کاربر از تاخیر پرواز در فرودگاه، رده اطلاعات مناسب است.

مثال ۲: برای اطلاع کاربر که آسانسور به طبقه مورد نظر نزدیک است، رده هشدار مناسب است.

مثال ۳: برای اطلاع کاربر که تنها ۲ تا از ۱۰ مورد درخواست موجود هست، رده خطا مناسب است.

مقدار پیش فرض باید "info" باشد.

۱۰-۵ مشخصه "sensitive"

عنصر `<notify>` (به بند ۱۰-۱ مراجعه کنید) مجاز است یک مشخصه "sensitive" داشته باشد. مشخصه "sensitive" باید یک مقدار بولی داشته باشد که نشان دهد آیا اعلان اطلاعات از نظر قانونی حساس را ارائه می کند.

مثال: اعلان ممکن است یک هشدار امنیتی فراهم کند که شکل ارائه آن پیامدهای قانونی داشته باشد.

این مثال با نشانه گذاری توصیف شده است:

```
<notify id="fireAlarmWarning" sensitive="true" />
```

مقدار پیش فرض باید "false" باشد.

۱۰-۶ مشخصه "optional"

عنصر `<notify>` (به بند ۱۰-۱ مراجعه کنید) مجاز است یک مشخصه "optional" از نوع بولی داشته باشد. مقدار پیش فرض آن باید "false" باشد.

اگر `optional="true"` باشد، اعلان ممکن است در زمان اجرا به خاطر محدودیت های گوناگون موجود نباشد.

یادآوری ۱- مثالهایی از محدودیت هایی که بر روی دسترسی عنصر سوکت تاثیر می گذارند: واپایش دسترسی، محصولات گوناگون که توضیحات سوکت مشترک به کار می برند اما با کمی تغییر در پیاده سازی (همانگونه که در UPnP DCPS)

یادآوری ۲- در دسترس بودن یک عنصر سوکت در زمان اجرا به URC از طریق ویژه TUN نشان داده می شود.

۷-۱۰ مشخصه "dim"

مشخصه "dim" یک اعلان را به صورت ابعادی (با یک یا چندین بعد) مشخص می سازد. در زمان اجرا یک اعلان ابعادی چندین حالت دارد، هر کدام برای ترکیب مشخصی از نمایه ها. عنصر <notify> (به بند ۱۰-۱ مراجعه کنید) مجاز است یک مشخصه "dim" داشته باشد. در صورت وجود باید شامل فهرست غیر خالی مرتب شده با فاصله از هم جدا شده از ارجاعات نوع باشد که نوع های نمایه اعلان هستند. اولین ارجاع نوع نمایه اولین بعد را، دومین نوع، دومین بعد را، و به همین ترتیب مشخص می سازد. ارجاعات نوع نمایه معتبر عبارتند از: (الف) نام یک نوع است که در قسمت <xsd:schema> از توضیحات سوکت تعریف یا وارد شده اند و (ب) نام کاملاً احراز شده (QName) از یک نوع بیرونی می باشد.

نمایه اعلان نباید مقدار "TTC" که برای نشان دادن فیلد "TTC" اش محفوظ داشته شده (۵.۲.۵.۲ را ببینید) را داشته باشد.

یادآوری ۱- بزرگترین مجموعه ممکن از مقادیر برای یک اعلان سوکت خاص از حاصلضرب همه انواع نمایه که از رفتن بر روی مسیر از عنصر <uiSocket> تا پایین به سمت اعلان سوکت خاص رخ می دهد (تعریف مسیر در ۵.۲.۵.۲ را ببینید) بدست می آید. با این حال، تمام ترکیبات نمایه در واقع در زمان اجرا رخ می دهد.

یادآوری ۲- هیچ ترتیب مشخص تعریف شده ای برای حالت های ناشی از اعلان ابعادی وجود ندارد. با این حال، مرتب کردن حابت ها بر اساس ترتیب مقادیر نمایه برای هدف - در انتقال حالت ها به URC - و برای URC - در ارائه حالت ها به کاربر توصیه می شود (ISO / IEC 24752-1 را ببینید).

یادآوری ۳- نوع نمایه ممکن است یک تعداد نامحدود از مقادیر نمایه ممکن را مشخص کند (برای مثال xsd: عدد صحیح بینهایت مقدار دارد). با این حال، مقادیر نمایه واقعی در زمان اجرا زیر مجموعه متناهی از مقادیر مجاز توسط نوع نمایه هستند.

یادآوری ۴- استفاده از یک اعلان ابعادی تنها اگر مجموعه اعلان ها واقعا همگن باشند معقول است، یعنی هر یک دارای مشخصه های یکسان از جمله برچسب (پیام) و متن کمک باشد.

یادآوری ۵- ارجاعات (برای مثال از منابع تفکیک ناپذیر) به اعلان های ابعادی به مولفه های منحصر به فرد از اعلان اشاره دارد. برای مثال آدرس <http://example.com/thermometer/socket#alert> به هر مولفه اعلان مجزایی در اعلان ابعادی با 'alert' id= اشاره دارد. با این حال، آدرس <http://example.com/thermometer/socket#alert> به اعلان به طور کلی با تمام مولفه ها اشاره دارد. ISO/IEC 24752-5 را برای جزئیات بیشتر ببینید.

یادآوری ۶- ترتیبی که در آن مولفه های یک اعلان ابعادی به کاربر ارائه می شود براساس ترتیب تعریف شده از نوع (های) نمایه مربوطه می باشد. ISO / IEC 24752-1 را برای جزئیات بیشتر ببینید.

۸-۱۰ مشخصه 'timeout'

عنصر `<notify>` (به بند ۱-۱۰ مراجعه کنید) ممکن است یک خصیصه `timeout`^۲ داشته باشند برای نشان دادن اینکه اعلان در زمانی که فعال است ممکن است مهلتش تمام شود.

مثال: کد زیر نشان دهنده یک اعلان با یک فاصله طور پیش فرض از ۱ دقیقه و ۱۵ ثانیه.

```
<notify id="timeoutNotification" type="yesNo" timeout="PT1M15S" />
```

همه عناصر `<notify>` (به بند ۱-۱۰ مراجعه کنید) که یک مهلت اتمام پاسخ کاربر دارند باید فاصله پیش فرض با یک مشخصه را نمایندگی کند. این مشخصه نشان دهنده زمان مجاز برای کاربران برای پاسخ به اعلان است قبل از اینکه هدف آن را رد کند. ارزش مشخصه باید حداقل ۱۰ ثانیه، و باید برای تبادل اعلان پیچیده طولانی تر باشد.

یادآوری ۱- توصیه می‌شود که اهداف یک گزینه برای کاربر به منظور گسترش مهلت اتمام یک اعلان (به بند ۱-۱۰ مراجعه کنید) فراهم کنند. این می‌تواند با آگاه ساختن کاربر قبل از مهلت اتمام رخ می‌دهد، و اجازه دادن به کاربر که `timeout` را گسترش دهد از طریق یک گفت و گو باشد. روش دیگر، هدف می‌تواند مهلت اتمام را به طور خودکار بر اساس تنظیمات کاربر تنظیم کند.

۱۰-۹ وابستگی اعلان

۱۰-۹-۱ کلیات

عنصر `<dependency>` می‌تواند به عنوان یک زیر عنصر از `<notify>` وجود داشته باشد (به بند ۱-۱۰ مراجعه کنید) در صورت وجود فقط یک بار ظاهر می‌شود.

۱۰-۹-۲ وابستگی `<insert>`

وابستگی `<insert>` مشخص می‌کند که آیا کاربر می‌تواند در زمان اجرا مجموعه‌ای از نمایه‌های مربوط را تغییر دهد یا خیر (به طوری که مجموعه نمایه‌ها توسط مشخصه `'dim'` روی عنصر `<notify>` مشخص می‌شود، به زیربند ۱-۱۰ را مراجعه کنید).

محتوای وابستگی `<insert>` باید یک عبارت معتبر XPath که به متغیر بولی مقدار گذاری شده است و در حین نشست تغییر می‌کند باشد. اگر غلط مقدار گذاری شود در نتیجه برای اضافه یا حذف کردن نمایه‌ها مناسب نیست در غیر این صورت برای تلاش به اضافه یا حذف کردن ترکیب نمایه مناسب است اگر چه هیچ تضمینی برای موفقیت ندارد.

یادآوری ۱- «اضافه کردن ترکیبی از نمایه» به معنی اضافه کردن مولفه اعلان توسط مشخصه `'dim'` مربوطه است. توجه داشته باشید که وضعیت اعلان اضافه شده توسط هدف تعیین می‌شود. «حذف یک ترکیب نمایه» بمعنی حذف مولفه اعلان برای ترکیب نمایه‌ای است که در حال وقوع است.

عنصر `<insert>` ممکن است به صورت زیرعنصری از `<dependency>` موجود باشد (به بند ۱-۹ مراجعه کنید) فقط در صورتی که عنصر `<notify>` موجود دارای مشخصه `'dim'` باشد (به بند ۱-۷ مراجعه کنید) در صورت وجود فقط یک بار می‌تواند ظاهر شود و مقادیر پیش فرض آن `“false()”` است.

یادآوری ۲- هیچ ارث برای از وابستگی `<insert>` مربوط به عنصر `<set>` برای آنها که عنصر `<notify>` را دارند وجود ندارد. هر وابستگی `<insert>` تنها به نمایه‌هایی وابسته است که در مشخصه `'dim'` آنها ذکر شده باشد و در همان سطح باشد.

۱۰-۱۰ متغیرها و فرمان‌های اعلان

برای اعلان‌های سنتی که به طور مثال مشخصه 'type' نداشته یا به صورت "type='custom'" باشد (به زیربند ۱۰-۳ مراجعه کنید) عنصر <notify> (به زیربند ۱۰-۱ مراجعه کنید) باید حاوی یک یا چند عنصر <variable> یا <command> باشد به همراه همه مشخصه‌ها و زیرعنصرهای مرتبط که در بند ۸ و بند ۹ مشخص شده است. برای اعلان‌های با نوع به غیر از "custom" عنصر <notify> نباید حاوی هیچ عنصر <variable> یا <command> باشد.

فرمان‌ها و متغیرهای اعلان باید مقدار تعریف نشده‌ای داشته باشند و آن را نباید به کاربر تا زمان فعال شدن اعلان نشان دهند.

زمانی که اعلان فعال شد یک محاوره کمکی باید برای کاربر نمایش داده شود که حاوی تمامی متغیرها و فرمان‌های مربوط است و توسط وابستگی‌های مجزا مشخص شده است. کاربر در این زمان می‌تواند با عناصر موجود در آن کار کند: مانند تغییر مقدار متغیر و/یا فعال سازی اجرای فرمان‌ها.

کاربر لازم است نهایتاً محاوره کمکی را تایید یا لغو کند (در صورت هماهنگی با نوع اعلان)

مثال: کد مقابل یک اعلان برای اطلاعات تعیین هویت کاربر را درخواست می‌کند (نام کاربری، رمز). این اعلان در صورتی نمایش داده می‌شود که کاربر عملیات برتر را اجرا کرده باشد.

یادآوری - در رابط کاربر گرافیکی، کاربر می‌تواند برای تأیید دکمه تأیید را فشار دهد و ارتباط با محاوره کمکی و عناصر آن را پایان دهد، همین طور دکمه لغو می‌تواند برای اعلان‌هایی با نوع "optionCancel"، "yesNoCancel" و "customCancel" به کار رود.

اگر به هر دلیلی در زمان تبادل کاربر با محاوره کمکی هدف وضعیت اعلان را به "inactive" تغییر دهد، کاربر در این مورد با خبر خواهد شد و محاوره کمکی بسته خواهد شد.

۱۰-۱۱ نگاشت اطلاعات مختص به بستر برای اعلان‌ها

عنصر <mapping> می‌تواند هر تعداد بار به صورت زیر عنصری از (به بند ۱۰-۱ مراجعه کنید) <notify> وجود داشته باشد برای قرار دادن نگاشت اطلاعات مختص به بستر برای اعلان‌ها.

عنصر <mapping> باید مشخصه 'platform' را که مقدار آن توسط استاندارد ISO/IEC 24752 ممنوع نشده باشد را دارا باشد

عنصر <mapping> ممکن است دارای عناصر یا زیرعناصر متفاوتی باشد با این حال زیر عناصر باید از فضای نامی به غیر از فضای نام uis باشند.

یادآوری ۱ - توصیف سوکتی که شامل اطلاعات نگاشت بستر باشد بیطرفی بستر خود را از دست می‌دهد، به همین علت چندین نگاشت برای توصیف سوکت در این شرایط لازم است (هر کدام برای یک بستر) توصیه می‌شود فناوری‌های دیگر برای مشخص ساختن رابط بستر مشخص را در نظر داشته باشید به عنوان مثال اطلاعات نگاشت می‌تواند توسط یک فایل بیرونی با ارجاع به عناصر موجود در توصیف سوکت باشد.

یادآوری ۲- کارخانه‌ها و بسترها برای اضافه کردن محتوا فعال یا اجرایی توسط عنصر <mapping> در توصیف سوکت دلسرد شده‌اند چرا که این کار باعث به وجود آمدن تهدید امنیتی برای مولفه درخواست شده در توصیف سوکت یا اجرای آن می‌شود.

۱۰-۱۲ خصیصه‌های اعلان در DCMI

هر عنصر یا عنصر پالایش شده در استاندارد ISO 15836، مجموعه عناصر هسته فراداده دوبلین یا مجموعه هسته فراداده دوبلین ابتکاری DCMI، موضوعات فرا داده ممکن است برای توصیف اعلان سوکت به کار رود، در این صورت هر کدام از آنها می‌تواند چندین بار به صورت زیر عنصری از <notify> اتفاق بیفتد (به زیر بند ۱۰-۱ مراجعه کنید)

۱۱ تعریف نوع

۱-۱۱ کلیات

توصیف سوکت می‌تواند شامل تعریف نوع متغیرهای موجود در همان سند باشد، این انواع باید از محدودیت‌ها و یا فهرست انواع توکار و یا دیگر نوع‌های مشتق شده گرفته شده باشد. به علاوه مشتق شدن در نتیجه اجتماع اجازه داده شده است در صورت مشتق شدن نوع از همان نوع داده توکار.

مثال: یک نوع مشتق شده که از اجتماع دو نوع مشتق شده از xsd:string تشکیل شده است می‌تواند در توصیف سوکت به کار روند.

برای دیدن جزئیات چگونگی مشتق کردن نوع در XSD " طرح‌واره XML قسمت ۲ را ببینید.

عنصر ریشه <uiSocket> (به زیر بند ۶-۱ مراجعه کنید) می‌تواند شامل یک عنصر <xsd:schema> محتوی تعریف نوع باشد.

عنصر <xsd:schema> نباید شامل مشخصه 'targetNamespace' باشد. انواع تعریف شده در توصیف سوکت فقط انواع محلی هستند (بدون معرف فضای نام) و از خارج از توصیف سوکت نمی‌توانند ارجاع داده شوند.

ویژگی 'name' در <xsd:simpleType> یا <xsd:complexType> باید یکتا در همه واقع شدن ویژگی‌های 'name' و 'id' روی هر عنصر XML در توصیف سوکت باشد.

یادآوری - مقدار ویژگی 'name' برای مشخص کردن برچسب یا هر منبع تفکیک ناپذیر دیگر برای نوع socket-internal در برگه منبع به کار می‌رود، مقادیر همه ویژگی‌های 'name' و 'id' باید یکتا باشد.

۱۱-۲ وجه‌ها^۱

نوع متغیرها باید با دقت کافی در تعریف سوکت مشخص شهر، وجه مشخص کننده فضای استفاده شده توسط مقدار نوع می‌باشد. انواع اساسی و محدودیتی وجه وجود دارد. برای جزئیات تعریف طرح‌واره XML بخش ۲ را ببینید.

مثال ۱: یک وجه اصلی 'ordered' می‌تواند هر کدام از مقادیر "partial"، "false" یا "total" را داشته باشد، این مقادیر اشاره‌ای است به چگونگی تغییر متغیر توسط کاربر در لایه معرفی. برای متغیرهایی که کاملاً از نوع سفارش داده شده می‌باشند یک روش ورودی مرتبط مثل یک منو کشویی با فرمان بالا و پایین وجود داشته باشد.

اگر محدودیتی در مجموعه قانونی مقادیر وجود داشته باشد این کار با مشتق کردن یک نوع ساده از نوع طرح‌واره XML با محدودیت کردن تعریف وجه در طرح‌واره XML قسمت ۲ انجام می‌شود.

محدودیت‌های موجود در طرح‌واره XML قسمت ۲ شامل length, minLength, maxLength, pattern, minExclusive, enumeration, whitespace, maxInclusive, maxExclusive, minInclusive است.

مثال ۲: یک متغیر که نشانگر یک شماره سریال کارمند است می‌تواند توسط یک رشته با طول ۹ کارکتر توسط این مشتق شدن از نوع داده رشته تعریف شده.

یک انتخاب محدود شده مجموعه مقادیر نشان دهنده استفاده از وجوه شمارشی است.

مثال ۳: یک طرح‌واره برای تعریف کردن روز های تعطیل در <http://www.w3.org/TR/2001/REC-xmlschema-2-20010502/#rf-fund-facets> به صورت مشتق شده از نوع xsd:gMonthDay نشان داده شده است.

۱۱-۲ فهرست مقادیر رشته ای

نوع uis:stringListItem برای ساختن یک فهرست از رشته ها از پیش تعریف شده است. uis:stringListItem از رشته توسط محدودیت مشتق شده و حاوی هیچ فاصله خالی نمی‌باشد. یک مجموعه ایستا از مقادیر (برای ارجاع در <selectionSetStatic> زیربند ۸-۹ را ببینید) می‌تواند توسط ساخت شمارشی از uis:stringListItem مشتق شود.

نوع uis:stringListItem باید به صورت پیش تعریف زیر باشد:

مثال: نوع جدید staticCityType برای دارا بودن فهرست ایستایی از شهر ها ساخته شده که در مثال ۳ از بند ۸-۹-۵ به کار رفته است.

در وجه length, minLength و maxLength می‌تواند برای اشاره اندازه مقادیری که فهرست می‌تواند دریافت کند به کار رود. بقیه وجه ها می‌توانند شمارشی یا الگویی باشند.

نوع پیش تعریف uis:stringList می‌تواند برای اشاره به فهرست رشته ای که هیچ مجموعه مقدار آیتم معتبر تا زمان اجرا ندارند به کار رود.

یادآوری - نوع های پیش تعریف uis:stringList و uis:stringListItem برای راحتی کار نویسنده توصیف سوکت فراهم شده. اگر نویسنده بخواهد از فهرست به غیر از نوع رشته استفاده کند باید از گرامر فراهم شده توسط XSD برای گرفتن از فهرست استفاده کند و مطمئن شود که مقدار آیتم از فاصله خالی استفاده نمی‌کند.

۱۱-۴ مشخص کردن ساختار در داخل فضای مقدار نوع

فضای مقدار نوع می‌تواند تعداد زیادی از مقدار های ممکن را داشته باشد و این فضای مقدار می‌تواند ساختار طبیعی داشته باشد. این ساختار می‌تواند با تعریف متغیرهای نوع به صورت مجموع زیر-نوع ها نشان داده شود.

یادآوری - ساختار نشان داده شده در اینجا فقط برای ترجمه ماشینی معنی دار است و منابع مورد نیاز برای تفسیر کاربر این ساختار مثل برچسب ها برای نوع ها و مقادیر جداگانه باید فراهم شود.

۱۱-۵ نوع Socket-internal

یک سوکت می‌تواند انواع ساده یا پیچیده را برای متغیر ها و ابعاد تعریف کند که این کار را با استفاده از عناصر `<xsd:simpleType>` و `<xsd:complexType>` (و زیر عناصر مربوطه) همان طور که در طرح‌واره XML قسمت ۱ آورده انجام می‌شود. به آنها "انواع درون سوکتی" گفته می‌شود. ویژگی 'name' در انواع درون سوکتی باید در بین تمام 'id' و 'name' داخل توصیف سوکت یکتا باشد.

۱۱-۶ وارد کردن طرح‌واره

یک سوکت ممکن است نوع و تعریف عنصر را از فضای نام خارجی وارد کند. که این کار را توسط عنصر `<xsd:import>` شرح داده شده در طرح‌واره XML بخش ۱ آورده انجام می‌شود. عنصر `<import>` می‌تواند شامل ویژگی های 'namespace' و 'schemaLocation' باشد به معنی تعریف شده در طرح‌واره XML قسمت ۱ باشد. مقدار ویژگی 'namespace' باید برابر با فضای نام هدف شما خارجی ارجاع داده شده توسط ویژگی 'schemaLocation' باشد.

۱۲ ملاحظات امنیتی

برای کاربردها و محیط هایی که به حفظ حریم خصوصی و یکپارچگی حساس هستند، توصیف سوکت باید توسط یک سطح مناسب امنیتی محافظت شود. فروشندگان تشویق می‌شوند تا به کار گیری خدمات‌های حفظ حریم شخصی و یکپارچگی را در نظر داشته باشند. هر چند سنجه‌های امنیتی قوی خارج از دامنه کاربرد این استاندارد است.

پیوست الف

(اطلاعاتی)

اسنادی برای توصیف سوکت واسط کاربر

این پیوست اطلاعات شامل مراجع برخطی می‌شوند که مربوط به این استاندارد است. منبع این اسناد به خدمت وب openURC به منظور خلاصه شدن این استاندارد واگذار شده‌اند.

الف-۱ یک توصیف سوکت برای ترموستات دیجیتالی که در اسناد WSDL بالا وجود داشت:

<http://openurc.org/TPL/basiv-thermostat-1/basic-thermostat-uis>

الف-۲ یک تعریف طرح واره برای توصیف سوکت واسط کاربر مطابق این استاندارد:

<http://openurc.org/ns/uiocketdesc-2>