INTERNATIONAL STANDARD

ISO/IEC 24752-3

First edition
2008-02-15

Information technology — User interfaces — Universal remote console —

Part 3: Presentation template

Technologies de l'information — Interfaces utilisateur — Console à distance universelle —

Partie 3: Modèle de présentation

Reference number
ISO/IEC 24752-3:2008(E)

PDF disclaimer

This PDF file may contain embedded typefaces. In accordance with Adobe's licensing policy, this file may be printed or viewed but shall not be edited unless the typefaces which are embedded are licensed to and installed on the computer performing the editing. In downloading this file, parties accept therein the responsibility of not infringing Adobe's licensing policy. The ISO Central Secretariat accepts no liability in this area.

Adobe is a trademark of Adobe Systems Incorporated.

Details of the software products used to create this PDF file can be found in the General Info relative to the file; the PDF-creation parameters were optimized for printing. Every care has been taken to ensure that the file is suitable for use by ISO member bodies. In the unlikely event that a problem relating to it is found, please inform the Central Secretariat at the address given below.

COPYRIGHT PROTECTED DOCUMENT

ISO copyright office
Case postale 56 • CH-1211 Geneva 20
Tel. + 41 22 749 01 11
Fax + 41 22 749 09 47
E-mail copyright@iso.org
Web www.iso.org

Published in Switzerland

Contents

Page

Foreword

ISO (the International Organization for Standardization) and IEC (the International Electrotechnical Commission) form the specialized system for worldwide standardization. National bodies that are members of ISO or IEC participate in the development of International Standards through technical committees established by the respective organization to deal with particular fields of technical activity. ISO and IEC technical committees collaborate in fields of mutual interest. Other international organizations, governmental and non-governmental, in liaison with ISO and IEC, also take part in the work. In the field of information technology, ISO and IEC have established a joint technical committee, ISO/IEC JTC 1.

International Standards are drafted in accordance with the rules given in the ISO/IEC Directives, Part 2.

The main task of the joint technical committee is to prepare International Standards. Draft International Standards adopted by the joint technical committee are circulated to national bodies for voting. Publication as an International Standard requires approval by at least 75 % of the national bodies casting a vote.

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights. ISO and IEC shall not be held responsible for identifying any or all such patent rights.

ISO/IEC 24752-3 was prepared by Joint Technical Committee ISO/IEC JTC 1, *Information technology,* Subcommittee SC 35, *User interfaces*.

ISO/IEC 24752 consists of the following parts, under the general title *Information technology — User interfaces — Universal remote console*:

— *Part 1: Framework*

— *Part 2: User interface socket description*

— *Part 3: Presentation template*

— *Part 4: Target description*

— *Part 5: Resource description*

Introduction

A presentation template (PreT) in the presentation template markup language is a user-oriented representation of a target described in a user interface socket specification. It maps user interface socket elements to interaction (presentation and user input) mechanisms. It provides a structure into which elements of the presentation are embedded. The presentation elements themselves are modality-independent, i.e. capable of adaptation to any delivery context.

The main content of a presentation template is a structured set of abstract interactors. These are user interface elements that describe a choice for the user to make, some input to obtain from the user, or some output to convey to the user. Each interactor is bound to a single socket element.

NOTE The PreT markup language is similar to the XForms form controls. However, significant differences exist. Firstly, PreT markup language does not include the specification of natural language atomic resources such as labels and help texts. These atomic resources are defined externally. Secondly, PreT interactors bind to user interface socket elements, and XForms form controls bind to elements in an XForms data model.

As a brief introduction to the presentation template concept, a short example is presented, showing a complete presentation template for a digital thermometer whose user interface socket is included in Annex B.

The thermometer is always on, as it is connected to the mains. It displays the current temperature, maximum and minimum temperatures, and can do this in Fahrenheit or centigrade. It includes a command for resetting the maximum and minimum temperatures. It requires the user to provide confirmation of a reset operation.

EXAMPLE

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

<pret name="http://example.com/thermometer/pret.xml"
      id="pret"
      xmlns="http://myurc.org/ns/pret"
      xmlns:dcterms="http://purl.org/dc/terms/">
      <dcterms:conformsTo>http://myurc.org/iso24752-3/2007</dcterms:conformsTo>
      <group id="readings">
            <output id="temperature" ref="http://example.com/thermometer/socket#temperature"/>
            <output id="maximum" ref="http://example.com/thermometer/socket#maximum"/>
            <output id="minimum" ref="http://example.com/thermometer/socket#minimum"/>
      </group>
      <select1 id="scale" ref="http://example.com/thermometer/socket#scale"/>
      <trigger id="reset" ref="http://example.com/thermometer/socket#reset"/>
      <modalDialog id="checkReset" ref=" http://example.com/thermometer/socket#checkReset">
            <trigger id="confirmReset" ref=" http://example.com/thermometer/socket#confirmReset"/>
            <trigger id="cancelReset" ref=" http://example.com/thermometer/socket#cancelReset"/>
      </modalDialog>
</pret>
```

Each of the constants, variables, commands and notify elements within the socket is represented by an interactor within the presentation template, and a grouping structure is provided, grouping the temperature readings together. The structure also provides a default order for presentation of the elements (the order of appearance within the description) and indicates which elements are relevant when the user is being asked to confirm a ‘reset’ command (notify state “checkReset” is active).

NOTE Resources such as labels, help texts and keywords pertaining to elements of a presentation template are not included within the presentation template itself but provided separately. Resources reference presentation template elements using the presentation template’s name (URI) and the element ids. Refer to ISO/IEC 24752-5 for further details.

Information technology — User interfaces — Universal remote console —

Part 3:
Presentation template

1 Scope

ISO/IEC 24752 is a multi-part International Standard to facilitate operation of information and electronic products through remote and alternative interfaces and intelligent agents.

This part of ISO/IEC 24752 defines a language (presentation template markup language) for describing modality-independent user interface specifications, or *presentation templates*, associated with a user interface socket description as defined by ISO/IEC 24752-2.

The purpose of a presentation template is to provide the URC with hints as to how to build a usable and consistent user interface for a target device or service that is described in a user interface socket description as referenced above. The hints are of an abstract nature, and are intended to apply to any delivery context. These hints primarily provide information on structuring, grouping and linearization of the socket elements. Elements within a presentation template can be referenced by atomic resources whose format is given by ISO/IEC 24752-5. Taken together, a presentation template, socket description, and appropriate atomic resources can be used to construct a user interface in any modality (e.g. visual, auditory, tactile, multimodal), through which a user can access and control a target.

2 Conformance

An extensible markup language (XML) file is a presentation template in conformance with this part of ISO/IEC 24752 if

- it has the MIME type specified in 5.2, if applicable, and
- its root element is the <pret> element as defined in Clause 6.

An XML file does not conform to this part of ISO/IEC 24752 if it uses any element, attribute or value that is not part of this specification.

NOTE 1 Target manufacturers who want to add information to a presentation template beyond the elements, attributes and values specified in this part of ISO/IEC 24752 can do so by externally providing (proprietary) resource descriptions that point into the structure of a presentation template. Refer to ISO/IEC 24752-5 for details.

NOTE 2 Future versions of this part of ISO/IEC 24752 might add new elements, attributes and values. They might also drop the policy of strict language conformance in favor of allowing for language extensions. Therefore, URC manufacturers are encouraged to implement their URCs so that unrecognized markup is ignored without failing.

3 Normative references

The following referenced documents are indispensable for the application of this document. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments) applies.

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

ISO/IEC 24752-2, *Information technology — User interfaces — Universal remote console — Part 2: User interface socket description*

ISO/IEC 10646, *Information technology — Universal Multiple-Octet Coded Character Set (UCS)*

ISO 15836:2003, *Information and documentation — The Dublin Core metadata element set*

DCMI Metadata Terms, http://dublincore.org/documents/dcmi-terms/

IETF RFC 2046, Multipurpose Internet Mail Extensions (MIME) Part Two: Media Types, November 1996, http://www.ietf.org/rfc/rfc2046.txt

IETF RFC 3986, Uniform Resource Identifier (URI): Generic Syntax, January 2005, http://www.ietf.org/rfc/rfc3986.txt

W3C Recommendation: Extensible Markup Language (XML) 1.0 (Third Edition), 04 February 2004, http://www.w3.org/TR/2004/REC-xml-20040204/

W3C Recommendation: Namespaces in XML, World Wide Web Consortium 14 January 1999, http://www.w3.org/TR/1999/REC-xml-names-19990114/

W3C Recommendation: XML Schema Part 2: Datatypes, W3C Recommendation 02 May 2001, http://www.w3.org/TR/2001/REC-xmlschema-2-20010502/

4 Terms and definitions

For the purposes of this document, the terms and definitions given in ISO/IEC 24752-1 and the following apply.

4.1
interactor
abstract or concrete user interface element that describes a choice for the user to make, some input to obtain from the user, or some output to convey to the user

4.2
abstract interactor
interactor that describes the selection, input, or output for a user interaction, without constraining the concrete form of the interaction

4.3
concrete interactor
interactor that describes the selection, input, or output for a user interaction, and includes information on the visual or non-visual realization of that interaction

EXAMPLE a list box or a particular speech grammar

4.4
uniform resource identifier
URI
name or address that refers to a resource, as defined in IETF RFC 3986

5 Relation to other standards

5.1 Relation to XML

This specification defines an XML based language. Markup in XML is case sensitive.

Tags and attributes are not localizable, i.e. their names and values are identical for all international languages. As with all XML based languages, white space characters immediately surrounding tags are non-significant.

This specification makes use of the namespaces concept defined in http://www.w3.org/TR/REC-xml-names/ to enable the import of element and attribute names defined elsewhere. All element and attribute names used in this document with no namespace prefix are defined by this International Standard and are part of the namespace http://myurc.org/ns/pret. If not used as default namespace, the namespace identifier 'pret' should be used for it.

This specification makes use of the following namespaces and uses the naming convention 'namespace:prefix' (e.g. 'xsd:schema') when referring to constructs from foreign namespaces.

- xmlns:xsd="http://www.w3.org/2001/XMLSchema": XML schema definition is used to describe type information for resources. This specification requires version 1.0 of the W3C XML Schema definition as identified by the URI shown.

- dc: The Dublin Core Metadata Element Set V1.1 namespace (http://purl.org/dc/elements/1.1/), as specified in ISO 15836:2003

- dcterms: The DCMI Metadata Terms namespace (http://purl.org/dc/terms).

Other namespaces may be included as required.

For an XML Schema definition for the PreT markup language see Annex A.

5.2 MIME type

A presentation template shall have a MIME type of “application/urc-pret+xml”, if applicable (as specified in IETF RFC 2046).

6 Structure of a presentation template

6.1 General

A presentation template is modality-independent, and can be thought of as a set of hints for rendering a socket. Elements in the description are mapped to elements in a user interface socket. A presentation template is defined as an XML document with the following structure.

A presentation template document shall be coded in UCS according to ISO/IEC 10646.

EXAMPLE
```
<pret name="http://example.com/thermometer/pret.xml" id="pret">
    <!-- … grouped abstract interactor elements … -->
</pret>
```

NOTE When the PreT markup language is used in an XML document, its namespace identifier http://myurc.org/ns/pret shall be included. See the example in Annex B for how to specify the namespace identifier in an XML document.

6.2 The 'name' attribute

The <pret> element shall have a 'name' attribute. It specifies the name of the presentation template expressed as a globally unique URI.

6.3 The 'id' attribute

The <pret> element shall have an 'id' attribute providing an identifier that is unique among all ids in the presentation template.

NOTE The identifier is used to specify external resources to the <pret> element, as exemplified in ISO/IEC 24752-5.

6.4 The 'cohesion' attribute

The <pret> element may have a 'cohesion' attribute. It specifies the level of cohesion of its direct subelements. Valid values are "strong", "normal" and "weak".

The default value is "normal".

This attribute and its meaning is described in section 7.4 where it applies to the <group> element.

NOTE The 'cohesion' attribute for <pret> is provided for consistency with the <group> element, and for future versions of this International Standard which may include the ability to reference a presentation template as a subcomponent from a presentation template. However the 'cohesion' attribute for <pret> should not be used because it is not clear what it means in this narrow context.

6.5 The <dcterms:conformsTo> element

The <pret> element shall have a subelement <dcterms:conformsTo> that specifies a reference to an established standard to which the presentation template conforms. The value, a URI, is provided as element content. The value http://myurc.org/iso24752-3/2007 indicates that the described presentation template conforms to this International Standard.

EXAMPLE <dcterms:conformsTo>http://myurc.org/iso24752-3/2007</dcterms:conformsTo>

NOTE 1 The value of the <dcterms:conformsTo> element can be used when testing for conformance of a presentation template.

NOTE 2 The <dcterms:conformsTo> element is taken from the set of Dublin Core Metadata Terms.

6.6 The <dcterms:modified> element

The <pret> element may include a <dcterms:modified> element. This indicates that the presentation template has been modified from its original version, but still uses the same name as identifier (see section 7.1). Its content is of type xsd:date or xsd:dateTime.

EXAMPLE <dcterms:modified>2003-12-30</dcterms:modified>

NOTE The <dcterms:modified> element is taken from the set of Dublin Core Metadata Terms.

A presentation template should remain stable wherever possible. A presentation template that is changed shall be assigned a new name (see section 6.2) or <dcterms:modified> element. If a presentation template

has multiple versions with the same name (but different modification dates), elements that have changed should be assigned a new identifier.

6.7 Presentation template properties from DCMI

Any element and element refinement (except <dcterms:conformsTo> and <dcterms:modified> which are referenced above) from the set of Dublin Core Metadata Initiative (DCMI) Metadata Terms may be used to describe a presentation template, if appropriate. Each of them may occur multiple times within a <pret> element. In particular, the following DCMI terms may be applied (as specified in ISO 15836:2003):

- <dc:creator> specifying the author of the presentation template
- <dc:contributor> specifying co-authors of the presentation template
- <dc:publisher> specifying the provider of the presentation template

6.8 Grouped abstract interactors

A *presentation template* specification describes a user interface as a set of abstract interactors. Interactors are modality-independent constructs that describe the selection, input, or output for a user interaction, without constraining the concrete form of the interaction. They are described in Section 8. More than one interactor may be bound to the same socket element.

The set of abstract interactors is hierarchically grouped, providing a basic structure. Each interactor is bound to a <constant>, <variable>, <command> or <notify> element in the user interface socket, and inherits properties of that socket element, including relevant and write dependencies, type information, and type restrictions. Where an interactor is bound to a constant or variable, the value of the interactor is the value of the constant/variable. If a user changes a variable's value through use of the interactor, the socket variable's value changes accordingly. Where an interactor is bound to a command, activation of the interactor causes activation of the socket command, and for commands with associated state, the interactor reflects the state of the command. Where an interactor is bound to a notify element, it is presented to the user only when the notify element is active.

7 Groups

7.1 General

Semantics about the relationships among interactors are useful when creating concrete user interfaces. One form of semantics is provided by structuring the interactors into hierarchical groups of related elements. In a presentation template, this is achieved by using the <group> element, which acts as a container for defining a hierarchy of interactors (except <modalDialog>, see 8.11). Groups may be nested.

A group is defined by the following markup:

EXAMPLE <group id="vcrControls"> … </group>

The grouping provided shall be as generic as possible, so that it can be applied in multiple delivery contexts (e.g. one, few or many options presented at a time).

7.2 The ‘id’ attribute

Every group shall have an ‘id’ attribute providing a string identifier that is unique among all ids within the presentation template.

NOTE The identifier is used to specify external resources to the group as exemplified in ISO/IEC 24752-5.

7.3 The ‘navindex’ attribute

A group may be assigned a specific position in a navigation sequence by using the attribute ‘navindex’. navindex is a non-negative integer in the range 0-32767 of type xsd:nonNegativeInteger. The default navigation order is the order in which interactors appear within the presentation template document.

EXAMPLE <group id="group1" navindex="10"/>

The ordering provided shall be as generic as possible, so that it can be applied in multiple delivery contexts (e.g. one, few or many options presented at a time).

The ‘navindex’ attribute is nested, meaning that the defined ordering applies within the immediate context of the enclosing group, rather than the entire presentation template. Groups or interactors with no ‘navindex’ attribute should follow interactors with an explicit ‘navindex’ attribute, regardless of its value.

NOTE The ‘navindex’ attribute applies to both groups and interactors. See section 8.2.3 for a code example.

7.4 The ‘cohesion’ attribute

A group may use a ‘cohesion’ attribute to specify the level of cohesion of the group. It describes how strongly the group's elements cohere to each other relative to their coherence to the siblings of the group. Valid cohesion levels are "strong", "normal" or "weak".

The default value is “normal”.

- "strong": Very strong cohesion, to the exclusion of all other groups. In other words, when dealing with this group, the user probably does not need to access other groups at all. It is recommended to use the group label when presenting this group. It is also recommended that only this group be included in a presentation, though a navigation route to reach the other available parts of the presentation shall be available.

- "normal" (default value): This is a group of items that fit logically together. It is recommended to use the group label when presenting this group.

- "weak": These items can be grouped together but need not be explicitly identified as a group.

8 Interactors

8.1 General

A presentation template uses interactors to describe user interface elements with which the user interacts. These include constant values and dynamic text displayed to the user, controls the user can operate, and values the user can fill in. Every interactor shall have an ‘id’ attribute providing a name for the interactor that is unique within the presentation template.

Note: The set of interactors is a subset of the form controls defined in the XForms 1.0 specification, plus one novel interactor for modal dialogs. All of the interactors in XForms are supported here, except for <submit> and <upload>.

The following interactors are valid:

- <input>: Enables free-form data entry

- <secret>: For entering information that is considered sensitive and not echoed to the user as it is being entered e.g. password entry

- <textarea>: Enables free-form data entry in a multi-line format

- <select1>: Allows the user to make a single selection from a set of choices
- <select>: Allows the user to make multiple selections from a set of choices
- <range>: Allows selection from a sequential range of values
- <trigger>: Triggers an action (e.g. a command)
- <output>: Displays a value that cannot be edited by the user
- <modalDialog>: Describes a dialog prompted by a target-triggered event

Interactors bind to socket elements with certain restrictions on the legal bindings, as specified in the description of each interactor. Annex D provides a summary of the legal bindings in matrix form.

NOTE 1 The XForms <submit> form control is not used in this International Standard. In a presentation template, the <trigger> form control should be used as an alternative, and bound to a command in a UI socket.

NOTE 2 The XForms <upload> form control is not used in this International Standard. In a presentation template, if a variable value is to be a local file, this would be achieved using the input interactor, bound to a socket variable of the appropriate type.

8.2 Interactor attributes

8.2.1 The 'id' attribute

All interactors shall have an 'id' attribute. id is a unique identifier among all 'id' attributes in a presentation template.

EXAMPLE <input id="temperature" ref=" http://example.com/thermometer/socket#temperature"/>

NOTE: The identifier is used to specify external resources for an interactor as exemplified in ISO/IEC 24752-5".

8.2.2 The 'ref' attribute

All interactors shall have a 'ref' attribute. It specifies a single socket element that the interactor is bound to. The value is the concatenation of the URI for the socket description (given as the value of the about attribute of the <uiSocket>element in the socket description document), the pound sign ('#') and the value of the 'id' attribute of the socket element (as fragment identifier).

EXAMPLE <input id="temperature" ref=" http://example.com/thermometer/socket#temperature"/>

There are restrictions on the legal bindings. Please see the pertinent interactor description for what socket elements a particular interactor may bind to. Additionally, Annex D provides an overview matrix of possible bindings.

NOTE It is possible that interactors of the same presentation template refer to elements in different sockets.

8.2.3 The 'navindex' attribute

Any interactor may be assigned a specific position in a navigation sequence by using the attribute navindex. navindex is a non-negative integer in the range 0-32767 of type xsd:nonNegativeInteger. The default navigation order is the order in which interactors appear within the presentation template document.

EXAMPLE 1 <input navindex="10"/>

The ordering provided shall be as generic as possible, so that it can be applied in multiple delivery contexts (e.g. one, few or many options presented at a time).

The ‘navindex’ attribute is nested, meaning that the defined ordering applies within the immediate context of the enclosing group, rather than the entire presentation template. Groups or interactors with no ‘navindex’ attribute should follow interactors with an explicit ‘navindex’ attribute, regardless of its value.

EXAMPLE 2:

```
<input id="gender" ref="http://example.com/registrationForm/socket#gender" />
<input id="name" ref="http://example.com/registrationForm/socket#name" navindex="1" />
<group id="address" navindex="1000">
      <input id="state" ref="http://example.com/registrationForm/socket#state" />
      <input id="street" ref="http://example.com/registrationForm/socket#street" navindex="1" />
      <input id="city" ref="http://example.com/registrationForm/socket#city" navindex="4" />
      <input id="country" ref="http://example.com/registrationForm/socket#country" />
</group>
<input id="email" ref="http://example.com/registrationForm/socket#email" />
```

This would order the items as: name, street, city, state, country, gender, email.

8.2.4 The ‘incremental’ attribute

The <input>, <secret>, <textarea>, <select1>, <select> and <range> interactors may have an ‘incremental’ attribute. It specifies whether any user-triggered change in the value of the interactor shall be propagated immediately to the socket element the interactor is bound to.

The value of this attribute is a Boolean. For the <input>, <secret> and <textarea> interactors, the default value is "false". For the <select1>, <select> and <range> interactors, the default value is "true".

EXAMPLE 1

```
<input id="search"
    ref=" http://example.com/thermometer/socket#search"
    incremental="true" />
```

In this example, a search input field propagates every keystroke of the user. This might be used to offer a dynamic list of topics that are relevant for all words that begin with the value of the search input field.

EXAMPLE 2

```
<range id="volume"
    ref="http://example.com/television/socket#volume" />
```

In this example, the volume would be changing as the user is moving a slider used to render the <range> interactor on a graphical URC platform.

8.2.5 Labels, help texts, keywords, and access keys provided as resources

No attributes are provided for specifying an interactor's labels, help texts, keywords, or access keys. They may be specified as atomic resources. Refer to ISO/IEC 24752-5 for further details.

8.3 The <input> interactor

The <input> interactor enables free-form data entry.

EXAMPLE

```
<input id="message" ref="http://example.com/emailer/socket#message"/>
```

The <input> interactor may bind to any of the following socket elements via the ‘ref’ attribute, if the socket element has secret="false".

- Variable of type xsd:boolean (or derived from xsd:boolean)
- Variable of type xsd:string (or derived from xsd:string)
- Variable of either type xsd:decimal or xsd:double (or derived from these types)

- Variable of type xsd:duration (or derived from xsd:duration)

- Variable of types xsd:dateTime, xsd:time, xsd:date, xsd:gYearMonth, xsd:gYear, xsd:gMonthDay, xsd:gDay or xsd:gMonth (or derived from any of these types)

If a socket element has secret="true" it shall be bound to the <secret> interactor (see section 8.4).

8.4 The <secret> interactor

The <secret> interactor enables data entry in a manner that makes it difficult for any third party present to discern the data that is being entered. A common use is for password entry.

EXAMPLE <secret id="password" ref=" http://example.com/homesecurity/socket#password"/>

The <secret> interactor may bind to the same socket elements as the <input> interactor, via the 'ref' attribute, provided the socket element has secret="true".

8.5 The <textarea> interactor

The <textarea> interactor enables free-form data entry and is intended for use in entering multi-line content, e.g., the body of an email message.

EXAMPLE <textarea id="emailbody" ref="http://example.com/emailclient/socket#body" />

The <textarea> interactor may bind to any of the following socket elements:

- Variable of type xsd:string (or derived from xsd:string). Typically the type definition will indicate that its value may be long by using appropriate constraining facets, e.g. minLength="100" or maxLength="240"

- Variable of a type that is derived from xsd:string by list. This is to allow for multiple selection done by entering the selection values rather than by picking from a list of values

8.6 The <select1> interactor

The <select1> interactor allows the user to make a single selection from multiple choices.

EXAMPLE <select1 id="scale" ref="http://example.com/thermometer/socket#scale"/>

The <select1> interactor shall have a set of options from which the user can select. These options shall be specified within the socket. In the socket, options are either specified implicitly by deriving a type by restriction, or are specified by attaching a <selection> element to the variable to which the interactor is bound. The <selection> element may specify a dynamic or static option set.

The <select1> interactor may be open or closed. A closed selection allows the user to pick from a given set of values only. An open selection allows for free text input in addition to a given set of values (suggestions).

Whether the <select1> interactor is closed or open depends on the specification of options in the pertaining socket element and its type definition. Open selections are based on socket elements whose type is derived by union from a string and a derivation by enumeration; or are based on socket elements who have a <selection> subelement with attribute closed="false".

The <select1> interactor may bind to any of the following socket elements, provided options are specified for them, and they have secret="false":

- Variable has <selection> subelement; or its type is derived by enumeration of string values

- Variable of type xsd:decimal or xsd:double (or type derived from any of these)

- Variable of type xsd:duration (or type derived from xsd:duration)
- Variable of type xsd:dateTime, xsd:time, xsd:date, xsd:gYearMonth, xsd:gYear, xsd:gMonthDay, xsd:gDay or xsd:gMonth (or any type derived from one of these types)

In the PreT markup language, lists of numbers shall be static (not dynamic).

8.7 The <select> interactor

The <select> interactor allows the user to make multiple selections from multiple choices.

EXAMPLE <select id="alarmtypes" ref="http://example.com/alarmclock/socket#alarmtypes" />

The <select> interactor shall have a set of options from which the user can select. The set of options may be closed or open. See section 8.6 for details how to specify closed and open option sets in the socket.

The <select> interactor may bind to any of the following socket elements, provided options are specified for them, and they have secret="false":

- Variable is of a type that is derived by list from one of the following types or any derivation from them: xsd:string, xsd:duration, xsd:dateTime, xsd:time, xsd:date, xsd:gYearMonth, xsd:gYear, xsd:gMonthDay, xsd:gDay or xsd:gMonth

8.8 The <range> interactor

The <range> interactor allows selection from a sequential range of values.

EXAMPLE <range id="alarmtime" ref="http://example.com/alarmclock/socket#alarmtime" />

The <range> interactor shall bind to a socket variable that has a range of values defined in the socket description. Ranges are defined in the socket description by deriving types by restriction. The <range> interactor shall bind to a variable whose type has a totally ordered value space, i.e. its fundamental facet ordered="total". See ISO/IEC 24752-2 for details.

The <range> interactor may bind to any of the following socket elements, provided a range is specified for them, and they have secret="false":

- Variable whose type is derived from xsd:decimal or xsd:double
- Variable whose type is derived from xsd:duration
- Variable whose type is derived from xsd:dateTime, xsd:time, xsd:date, xsd:gYearMonth, xsd:gYear, xsd:gMonthDay, xsd:gDay or xsd:gMonth

8.9 The <trigger> interactor

The <trigger> interactor allows for user-triggered actions.

EXAMPLE <trigger id="reset" ref="http://example.com/thermometer/socket#reset"/>

The <trigger> interactor shall bind to a socket command.

8.10 The <output> interactor

The <output> interactor displays a value that cannot be edited by the user.

EXAMPLE <output id="temperature" ref="http://example.com/thermometer/socket#temperature"/>

The <output> interactor shall be bound to one of the following socket elements:

- Any constant – the value of the constant is displayed
- Any variable – the value of the variable is displayed
- Any command – the status of the command is displayed

8.11 The <modalDialog> interactor

A modal dialog is an interaction that suspends normal operation of the user interface until the dialog has been handled. It is used to present information, warnings and errors generated by a target.

A <modalDialog> interactor shall occur as subelement of <pret> only (not as subelement of <group>). <pret> may have any number of <modalDialog> subelements.

The <modalDialog> interactor shall bind to a socket <notify> element.

A <modalDialog> interactor is represented by the following markup:

EXAMPLE 1 <modalDialog id="verifyReset" ref=http://example.com/thermometer/socket#checkReset />

A <modalDialog> may contain other interactors. These are interactors that are necessary or helpful to the user in handling the notification. By default, all interactors are readable by a user when a notify state is active. However, only a subset of the interactors may be relevant during a notify state. Inclusion of an interactor within a <modalDialog> provides a hint that this interactor is important in handling this notification. For example, a television may notify a user that their chosen channel is currently unavailable and offer the option of changing to the nearest available channel or staying on the current channel. Information on the current channel would be helpful in making this decision. The presentation template for the television would indicate this by specifying the <modalDialog> as shown:

EXAMPLE 2

```
<modalDialog id=channelNotAvailable ref="http://example.com/tv#noChannel>
    <trigger id="chooseNearest" ref="http://example.com/tv#chooseNearestChannel"/>
    <trigger id="chooseCurrent" ref="http://example.com/tv#keepCurrentChannel"/>
    <output id="currentChannelReminder" ref="http://example.com/tv#channel"/>
</modalDialog>
```

Note that the same variable "channel" in the socket will be bound to an <input> or <select1> interactor elsewhere in the presentation template, allowing the user to directly control the current channel.

The interactors within a <modalDialog> may be grouped using the <group> element.

Annex A
(informative)

XML schema definition for presentation template

An XML Schema Definition for presentation templates is available at http://myurc.org/ns/pret.

Annex B
(informative)

Example user interface socket for a digital thermometer

NOTE This example is drawn from ISO/IEC 24752-2.

The thermometer is connected to the mains, and so is always available. It displays the current temperature and recent maximum and minimum temperatures, and can do this in Fahrenheit or centigrade. It includes a command for resetting the maximum and minimum temperatures back to the current temperature.

The socket specification is shown below. It contains a constant of the numerical value 570, the read-only variables "temperature", "maximum" and "minimum" of type double representing the current temperature and maximum and minimum recorded temperatures, and a writeable enumerated type variable "scale" representing the temperature scale used in the display (centigrade or Fahrenheit). A "reset" command to reset the maximum and minimum temperatures to the current temperature is also available. When a "reset" command is given, the thermometer enters a confirmation state in which the user confirms or cancels the command. This is represented by the "checkReset" notify state, and the "confirmReset" and "cancelReset" commands, which are only executable when the "checkReset" notify state is active. When the "checkReset" notify state is active, all other variables and commands can be accessed but not written (i.e. the only thing a user can do is to confirm or cancel the reset operation).

The final element of the socket description is an XSD type schema specifying the enumerated type indicating the temperature scale to be used.

Note that this specification is a static document and does not include information on the current state of the socket – values for the variables and other dynamically changing information are provided separately. Also note that the example does not contain commands for accessing the values of the variables – it is assumed that the value of a variable will be made accessible to a user in an appropriate way. Commands are limited to those functions that cannot be represented by data values.


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

<!-- Note: This document should be served with a MIME type of “application/urc-uisocketdesc+xml”, if applicable -->



<uiSocket
      about="http://example.com/thermometer/socket"
      id="socket"
      xmlns="http://myurc.org/ns/uisocketdesc"
      xmlns:uis="http://myurc.org/ns/uisocketdesc"
      xmlns:dc="http://purl.org/dc/elements/1.1/"
      xmlns:dcterms="http://purl.org/dc/terms/"
      xmlns:xsd="http://www.w3.org/2001/XMLSchema"
      xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
      xsi:schemaLocation="http://myurc.org/ns/uisocketdesc http://myurc.org/ns/uisocketdesc
            http://purl.org/dc/elements/1.1/ http://dublincore.org/schemas/xmls/qdc/2006/01/06/dc.xsd
            http://purl.org/dc/terms/ http://dublincore.org/schemas/xmls/qdc/2006/01/06/dcterms.xsd">

      <dcterms:conformsTo>http://myurc.org/iso24752-2/2007</dcterms:conformsTo>
      <dc:creator>Trace R&amp;D Center</dc:creator>
      <dc:title xml:lang="en">User interface socket description for a digital thermometer</dc:title>
      <dcterms:modified>2007-08-20</dcterms:modified>

      <constant id="modelNumber" type="xsd:double">
            570
            <dependency>
                  <relevant>false()</relevant>
            </dependency>
```

```
    </constant>
    <variable id="temperature" type="xsd:double">
        <dependency>
            <write>false()</write>
        </dependency>
    </variable>
    <variable id="maximum" type="xsd:double">
        <dependency>
            <write>false()</write>
        </dependency>
    </variable>
    <variable id="minimum" type="xsd:double">
        <dependency>
            <write>false()</write>
        </dependency>
    </variable>
    <variable id="scale" type="scaleType"/>
    <command id="reset"/>
    <notify id="checkReset" category="alert">
        <dependency>
            <explicitAck> false() </explicitAck>
            <acknowledge>
              (uis:hasDefinedValue('confirmReset') and uis:value('confirmReset') eq 'done')
              or (uis:hasDefinedValue('cancelReset') and uis:value('cancelReset') eq 'done')
            </acknowledge>
        </dependency>
    </notify>
    <command id="confirmReset" type="uis:basicCommand">
        <dependency>
            <relevant>uis:hasDefinedValue('checkReset') and uis:value('checkReset') eq 'active'</relevant>
            <write>uis:hasDefinedValue('checkReset') and uis:value('checkReset') eq 'active'</write>
        </dependency>
    </command>
    <command id="cancelReset" type="uis:basicCommand">
        <dependency>
            <relevant>uis:hasDefinedValue('checkReset') and uis:value('checkReset') eq 'active'</relevant>
            <write>uis:hasDefinedValue('checkReset') and uis:value('checkReset') eq 'active'</write>
        </dependency>
    </command>
    <xsd:schema>
        <xsd:simpleType name="scaleType" id="idScaleType">
            <xsd:restriction base="xsd:string">
              <xsd:enumeration value="F"/>
              <xsd:enumeration value="C"/>
            </xsd:restriction>
        </xsd:simpleType>
    </xsd:schema>
</uiSocket>
```