Internet Engineering Task Force
Internet Draft

Wu/Schulzrinne/Lennox
Columbia University

draft-wu-cpl-schema-02.txt
March 3, 2003
Expires: September 2003

An Extensible Markup Language Schema for Call Processing Language (CPL)

STATUS OF THIS MEMO

Abstract

This document provides an Extensible Markup Language (XML) Schema for the Call Processing Language (CPL). The original CPL specification only provides a Document Type Declaration (DTD) to describe the structure of the language. Compared with XML DTDs, XML schemas have many advantages such as performing stricter type checking, providing pre-defined data types and being able to derive new data types from existing ones. We further split the CPL schema into two parts, one contains elements common to all the telecommunication entities, such as user agents or presence agents. The other contains elements, such as 'proxy', specifically for network servers.

1 Introduction

The Call Processing Language (CPL) [1] is a language that can be used to describe and control Internet telephony services. It is based on the Extensible Markup Language (XML) [2], a common hierarchical format for describing structured data.

There are two primary formats used to describe the structure of an XML document. The older one, the Document Type Declaration (DTD) is part of the original XML specification; the newer one, the XML schema [3] was defined later. At the time the CPL specification was written, the XML schema specification had not yet been finalized, so the CPL specification only provides a DTD for the CPL.

Compared with XML DTDs, XML schemas have many advantages. XML schemas perform stricter type checking and provide many pre-defined data types. Also, XML schemas allow the derivation of new data types from existing ones, making it easier to use XML schemas for CPL extensions.

This document therefore defines an XML schema for the CPL. And we recommend that all future extensions of CPL should use schema definitions only.

We notice that the original CPL standard contains elements specifically for network servers, such as the 'proxy' and 'redirect' actions. To make CPL applicable to the other entities, such as user agent or presence agent, we put the elements specifically for network servers in a separate XML schema, and keep the elements common to all the entities in a schema, named base CPL schema. With the separation, it is easier to define CPL extensions for user agents or presence agents, without dealing with the actions user agents or presence agents cannot perform. All the future CPL extensions MUST base on the base CPL schema.

2 Overview of the schema

Most of the data types defined in the CPL DTD can be easily translated to an XML schema. However, two limitations on XML schema make several data types of CPL difficult to define. The first limitation is that several CPL data types are defined as having case-insensitive values, whereas XML schemas cannot easily define case-insensitive strings, so in the CPL schema, we have to use string patterns to define some tokens. For example, we have to use <xs:pattern value="[y|Y][e|E][a|A][r|R][l|L][y|Y]"/> to define the token 'YEARLY' in a case-insensitive manner. The second limitation of XML schema is that it only allows white space as a list delimiter. However, the time-switch of CPL follows Internet Calendaring and

Scheduling Core Object Specification (iCalendar COS), RFC 2445 [4], which uses comma (,) as its list delimiter. So we cannot use <xs:list> to define some data types, such as the ByDayType, which contains a list of days of the week. We have to use string pattern to define the list. In addition, the time format in iCalendar is different from that provided in XML schema, so we have to use <xs:string> to define the time, instead of using <xs:datetime> type.

To make CPL more extensible, in the XML schema of CPL, we introduce three abstract elements, namely 'trigger', 'switch', and 'action', which accordingly have the abstract type 'TriggerType', 'SwitchType', and 'ActionType'. Trigger is mapped to the top-level call processing action, such as 'incoming', in the original CPL standard. Any CPL trigger MUST be defined as the substitutionGroup of the abstract 'trigger' element, and have the type extended from the 'TriggerType'. Switch and action are the same as those defined in the original CPL standard. Any CPL switch MUST be defined as the substitutionGroup of the abstract 'switch' element, and have the type extended from the 'SwitchType'. Any CPL action MUST be defiend as the substitutionGroup of the abstract 'action' element, and have the type extended from the 'ActionType'.

3 The XML schema of the CPL

The following are two XML schemas. One is the base CPL schema, the other is the CPL extensions for network servers. We have checked the schema with the examples in the original CPL specification. The namespace URIs for elements defined by this specification are URNs [5], using the namespace identifier 'ietf' defined by [6] and extended by [7]. The URN for the base CPL schema is:

urn:ietf:params:xml:ns:cpl

The URN for the CPL extensions for network servers is:

urn:ietf:params:xml:ns:cpl:nserver

The base CPL schema is as below:

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema targetNamespace="urn:ietf:params:xml:ns:cpl"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  xmlns="urn:ietf:params:xml:ns:cpl">
  <xs:complexType name="TriggerType" abstract="true">
    <xs:group ref="Node"/>
  </xs:complexType>
```

```
    <xs:element name="trigger" type="TriggerType"/>
    <xs:complexType name="ActionType" abstract="true"/>
    <xs:element name="action" type="ActionType"/>
    <xs:complexType name="SwitchType" abstract="true"/>
    <xs:element name="switch" type="SwitchType"/>
    <xs:group name="Location">
      <xs:choice>
        <xs:element name="location" type="LocationType"/>
        <xs:element name="lookup" type="LookupType"/>
        <xs:element name="remove-location" type="RemoveLocationType"/>
      </xs:choice>
    </xs:group>
    <xs:group name="Sub">
      <xs:all>
        <xs:element name="sub" type="SubAction"/>
      </xs:all>
    </xs:group>
    <xs:group name="Node">
      <xs:choice>
        <xs:element ref="switch" minOccurs="0"/>
        <xs:group ref="Location" minOccurs="0"/>
        <xs:group ref="Sub" minOccurs="0"/>
        <xs:element ref="action" minOccurs="0" maxOccurs="unbounded"/>
      </xs:choice>
    </xs:group>
    <xs:complexType name="OtherwiseAction">
      <xs:group ref="Node"/>
    </xs:complexType>
    <xs:complexType name="NotPresentAction">
      <xs:group ref="Node"/>
    </xs:complexType>
    <xs:simpleType name="YesNoType">
      <xs:restriction base="xs:NMTOKEN">
        <xs:enumeration value="yes"/>
        <xs:enumeration value="no"/>
      </xs:restriction>
    </xs:simpleType>
    <xs:simpleType name="AddressFieldType">
      <xs:restriction base="xs:NMTOKEN">
        <xs:enumeration value="origin"/>
        <xs:enumeration value="destination"/>
        <xs:enumeration value="original-destination"/>
      </xs:restriction>
    </xs:simpleType>
    <xs:simpleType name="AddressSubfieldType">
      <xs:restriction base="xs:NMTOKEN">
        <xs:enumeration value="address-type"/>
        <xs:enumeration value="user"/>
```

```
        <xs:enumeration value="host"/>
        <xs:enumeration value="port"/>
        <xs:enumeration value="tel"/>
        <xs:enumeration value="display"/>
        <xs:enumeration value="password"/>
        <xs:enumeration value="alias-type"/>
      </xs:restriction>
    </xs:simpleType>
    <xs:complexType name="AddressType">
      <xs:group ref="Node"/>
      <xs:attribute name="is" type="xs:string"/>
      <xs:attribute name="contains" type="xs:string"/>
      <xs:attribute name="subdomain-of" type="xs:string"/>
    </xs:complexType>
    <xs:complexType name="AddressSwitchType">
      <xs:sequence>
        <xs:element name="address" type="AddressType"
                    minOccurs="0" maxOccurs="unbounded"/>
        <xs:sequence minOccurs="0">
          <xs:element name="not-present" type="NotPresentAction"/>
          <xs:element name="address" type="AddressType"
                      minOccurs="0" maxOccurs="unbounded"/>
        </xs:sequence>
        <xs:element name="otherwise" type="OtherwiseAction" minOccurs="0"/>
      </xs:sequence>
      <xs:attribute name="field" type="AddressFieldType" use="required"/>
      <xs:attribute name="subfield" type="AddressSubfieldType" use="optional"/
>
    </xs:complexType>
    <xs:element name="address-switch" type="AddressSwitchType"
                substitutionGroup="switch"/>
    <xs:simpleType name="StringFieldType">
      <xs:restriction base="xs:NMTOKEN">
        <xs:enumeration value="subject"/>
        <xs:enumeration value="organization"/>
        <xs:enumeration value="user-agent"/>
        <xs:enumeration value="display"/>
      </xs:restriction>
    </xs:simpleType>
    <xs:complexType name="StringType">
      <xs:group ref="Node"/>
      <xs:attribute name="is" type="xs:string" use="optional"/>
      <xs:attribute name="contains" type="xs:string" use="optional"/>
    </xs:complexType>
    <xs:complexType name="StringSwitchType">
      <xs:sequence>
        <xs:element name="string" type="StringType"
                    minOccurs="0" maxOccurs="unbounded"/>
```

```
        <xs:sequence minOccurs="0">
```

```
          <xs:element name="not-present" type="NotPresentAction"/>
          <xs:element name="string" type="StringType"
                      minOccurs="0" maxOccurs="unbounded"/>
        </xs:sequence>
        <xs:element name="otherwise" type="OtherwiseAction" minOccurs="0"/>
      </xs:sequence>
      <xs:attribute name="field" type="StringFieldType" use="required"/>
    </xs:complexType>
    <xs:element name="string-switch" type="StringSwitchType"
                substitutionGroup="switch"/>
    <xs:simpleType name="LanguageTagType">
      <xs:restriction base="xs:string"/>
    </xs:simpleType>
    <xs:complexType name="LanguageType">
      <xs:group ref="Node"/>
      <xs:attribute name="matches" type="LanguageTagType" use="required"/>
    </xs:complexType>
    <xs:complexType name="LanguageSwitchType">
      <xs:sequence>
        <xs:element name="language" type="LanguageType"
                    minOccurs="0" maxOccurs="unbounded"/>
        <xs:sequence minOccurs="0">
          <xs:element name="not-present" type="NotPresentAction"/>
          <xs:element name="language" type="LanguageType"
                      minOccurs="0" maxOccurs="unbounded"/>
        </xs:sequence>
        <xs:element name="otherwise" type="OtherwiseAction" minOccurs="0"/>
      </xs:sequence>
    </xs:complexType>
    <xs:element name="language-switch" type="LanguageSwitchType"
                substitutionGroup="switch"/>
    <xs:simpleType name="FreqType">
      <xs:restriction base="xs:NMTOKEN">
        <xs:pattern value="[s|S][e|E][c|C][o|O][n|N][d|D][l|L][y|Y]"/>
        <xs:pattern value="[m|M][i|I][n|N][u|U][t|T][e|E][l|L][y|Y]"/>
        <xs:pattern value="[h|H][o|O][u|U][r|R][l|L][y|Y]"/>
        <xs:pattern value="[d|D][a|A][i|I][l|L][y|Y]"/>
        <xs:pattern value="[w|W][e|E][e|E][k|K][l|L][y|Y]"/>
        <xs:pattern value="[m|M][o|N][n|N][t|T][h|H][l|L][y|Y]"/>
        <xs:pattern value="[y|Y][e|E][a|A][r|R][l|L][y|Y]"/>
      </xs:restriction>
    </xs:simpleType>
    <xs:simpleType name="DayType">
      <xs:restriction base="xs:NMTOKEN">
        <xs:pattern value="[m|M][o|O]"/>
        <xs:pattern value="[t|T][u|U]"/>
        <xs:pattern value="[w|W][e|E]"/>
        <xs:pattern value="[t|T][h|H]"/>
```

```
        <xs:pattern value="[f|F][r|R]"/>
        <xs:pattern value="[s|S][a|A]"/>
        <xs:pattern value="[s|S][u|U]"/>
      </xs:restriction>
    </xs:simpleType>
    <xs:simpleType name="PositiveYearDayType">
      <xs:restriction base="xs:integer">
        <xs:minInclusive value="1"/>
        <xs:maxInclusive value="366"/>
      </xs:restriction>
    </xs:simpleType>
    <xs:simpleType name="NegativeYearDayType">
      <xs:restriction base="xs:integer">
        <xs:minInclusive value="-366"/>
        <xs:maxInclusive value="-1"/>
      </xs:restriction>
    </xs:simpleType>
    <xs:simpleType name="YearDayType">
      <xs:union memberTypes="PositiveYearDayType NegativeYearDayType"/>
    </xs:simpleType>
    <xs:simpleType name="ByYearDayType">
      <xs:restriction base="xs:string">
        <xs:pattern value="([+|-]?([1-9]|[1-9][0-9]|[1-2][0-9][0-9]|[3][0-5]
[0-9]|[3][6][0-6]))(,([+|-]?([1-9]|[1-9][0-9]|[1-2][0-9][0-9]|[3][0-5][0-9]|[3]
[6][0-6])))*"/>
      </xs:restriction>
    </xs:simpleType>
    <xs:simpleType name="PositiveMonthDayType">
      <xs:restriction base="xs:integer">
        <xs:minInclusive value="1"/>
        <xs:maxInclusive value="31"/>
      </xs:restriction>
    </xs:simpleType>
    <xs:simpleType name="NegativeMonthDayType">
      <xs:restriction base="xs:integer">
        <xs:minInclusive value="-31"/>
        <xs:maxInclusive value="-1"/>
      </xs:restriction>
    </xs:simpleType>
    <xs:simpleType name="MonthDayType">
      <xs:union memberTypes="PositiveMonthDayType NegativeMonthDayType"/>
    </xs:simpleType>
    <xs:simpleType name="ByMonthDayType">
      <xs:restriction base="xs:string">
        <xs:pattern value="([+|-]?([1-9]|[1-2][0-9]|[3][0-1]))(,([+|-]?([1-9]|
[1-2][0-9]|[3][0-1])))*"/>
      </xs:restriction>
    </xs:simpleType>
```

```
     <xs:simpleType name="ExtendedDayType">
       <xs:restriction base="xs:string">
         <xs:pattern value="([+|-]?([1-9]|[1-9][0-9]|[1-2][0-9][0-9]|[3][0-5]
[0-9]|[3][6][0-6]))?([M|m][O|o]|[T|t][U|u]|[W|w][E|e]|[T|t][H|h]|[F|f][R|r]|[S|
s][A|a]|[S|s][U|u])"/>
```

```
      </xs:restriction>
    </xs:simpleType>
    <xs:simpleType name="ByDayType">
      <xs:restriction base="xs:string">
        <xs:pattern value="([+|-]?([1-9]|[1-9][0-9]|[1-2][0-9][0-9]|[3][0-5]
[0-9]|[3][6][0-6]))?([M|m][O|o]|[T|t][U|u]|[W|w][E|e]|[T|t][H|h]|[F|f][R|r]|[S|
s][A|a]|[S|s][U|u])(,([+|-]?([1-9]|[1-9][0-9]|[1-2][0-9][0-9]|[3][0-5][0-9]|[3]
[6][0-6]))?([M|m][O|o]|[T|t][U|u]|[W|w][E|e]|[T|t][H|h]|[F|f][R|r]|[S|s][A|a]|
[S|s][U|u]))*"/>
      </xs:restriction>
    </xs:simpleType>
    <xs:simpleType name="SecondMinuteType">
      <xs:restriction base="xs:integer">
        <xs:minInclusive value="0"/>
        <xs:maxInclusive value="59"/>
      </xs:restriction>
    </xs:simpleType>
    <xs:simpleType name="BySecondMinuteType">
      <xs:restriction base="xs:string">
        <xs:pattern value="([0-9]|[1-5][0-9])(,([0-9]|[1-5][0-9]))*"/>
      </xs:restriction>
    </xs:simpleType>
    <xs:simpleType name="HourType">
      <xs:restriction base="xs:integer">
        <xs:minInclusive value="0"/>
        <xs:maxInclusive value="23"/>
      </xs:restriction>
    </xs:simpleType>
    <xs:simpleType name="ByHourType">
      <xs:restriction base="xs:string">
        <xs:pattern value="([0-9]|[1][0-9]|[2][0-3])(,([0-9]|[1][0-9]|[2]
[0-3]))*"/>
      </xs:restriction>
    </xs:simpleType>
    <xs:simpleType name="PositiveWeekType">
      <xs:restriction base="xs:integer">
        <xs:minInclusive value="1"/>
        <xs:maxInclusive value="53"/>
      </xs:restriction>
    </xs:simpleType>
    <xs:simpleType name="NegativeWeekType">
      <xs:restriction base="xs:integer">
        <xs:minInclusive value="-53"/>
        <xs:maxInclusive value="-1"/>
      </xs:restriction>
    </xs:simpleType>
    <xs:simpleType name="WeekType">
      <xs:union memberTypes="PositiveWeekType NegativeWeekType"/>
```

```
    </xs:simpleType>
    <xs:simpleType name="ByWeekType">
      <xs:restriction base="xs:string">
        <xs:pattern value="([+|-]?([1-9]|[1-4][0-9]|[5][0-3]))(,([+|-]?([1-9]|
[1-4][0-9]|[5][0-3])))*"/>
      </xs:restriction>
```

```
    </xs:simpleType>
    <xs:simpleType name="MonthType">
      <xs:restriction base="xs:integer">
        <xs:maxInclusive value="12"/>
        <xs:minInclusive value="1"/>
      </xs:restriction>
    </xs:simpleType>
    <xs:simpleType name="ByMonthType">
      <xs:restriction base="xs:string">
        <xs:pattern value="([1-9]|[1][1-2])(,([1-9]|[1][1-2]))*"/>
      </xs:restriction>
    </xs:simpleType>
    <xs:complexType name="TimeType">
      <xs:group ref="Node"/>
      <xs:attribute name="dtstart" type="xs:string" use="required"/>
      <xs:attribute name="dtend" type="xs:string"/>
      <xs:attribute name="duration" type="xs:string"/>
      <xs:attribute name="freq" type="FreqType"/>
      <xs:attribute name="until" type="xs:string"/>
      <xs:attribute name="count" type="xs:integer"/>
      <xs:attribute name="interval" type="xs:integer" default="1"/>
      <xs:attribute name="bysecond" type="BySecondMinuteType"/>
      <xs:attribute name="byminute" type="BySecondMinuteType"/>
      <xs:attribute name="byhour" type="ByHourType"/>
      <xs:attribute name="byday" type="ByDayType"/>
      <xs:attribute name="bymonthday" type="ByMonthDayType"/>
      <xs:attribute name="byyearday" type="ByYearDayType"/>
      <xs:attribute name="byweekno" type="ByWeekType"/>
      <xs:attribute name="bymonth" type="ByMonthType"/>
      <xs:attribute name="wkst" type="DayType" default="MO"/>
      <xs:attribute name="bysetpos" type="YearDayType"/>
    </xs:complexType>
    <xs:simpleType name="TZIDType">
      <xs:restriction base="xs:string"/>
    </xs:simpleType>
    <xs:simpleType name="TZURLType">
      <xs:restriction base="xs:anyURI"/>
    </xs:simpleType>
    <xs:complexType name="TimeSwitchType">
      <xs:sequence>
        <xs:element name="time" type="TimeType"/>
        <xs:sequence minOccurs="0">
          <xs:element name="not-present" type="NotPresentAction"/>
          <xs:element name="time" type="TimeType"
                      minOccurs="0" maxOccurs="unbounded"/>
        </xs:sequence>
        <xs:element name="otherwise" type="OtherwiseAction" minOccurs="0"/>
      </xs:sequence>
```

```
      <xs:attribute name="tzid" type="TZIDType"/>
      <xs:attribute name="tzurl" type="TZURLType"/>
    </xs:complexType>
    <xs:element name="time-switch" type="TimeSwitchType"
                substitutionGroup="switch"/>
    <xs:simpleType name="PriorityValues">
      <xs:restriction base="xs:NMTOKEN">
        <xs:pattern value="[e|E][m|M][e|E][r|R][g|G][e|E][n|N][c|C][y|Y]"/>
        <xs:pattern value="[u|U][r|R][g|G][e|E][n|N][t|T]"/>
        <xs:pattern value="[n|N][o|O][r|R][m|M][a|A][l|L]"/>
        <xs:pattern value="[n|N][o|O][n|N]-[u|U][r|R][g|G][e|E][n|N][t|T]"/>
      </xs:restriction>
    </xs:simpleType>
    <xs:complexType name="PriorityType">
      <xs:group ref="Node"/>
      <xs:attribute name="less" type="PriorityValues"/>
      <xs:attribute name="greater" type="PriorityValues"/>
      <xs:attribute name="equal" type="PriorityValues"/>
    </xs:complexType>
    <xs:complexType name="PrioritySwitchType">
      <xs:sequence>
        <xs:element name="priority" type="PriorityType"/>
        <xs:sequence minOccurs="0">
          <xs:element name="not-present" type="NotPresentAction"/>
          <xs:element name="priority" type="PriorityType"
                      minOccurs="0" maxOccurs="unbounded"/>
        </xs:sequence>
        <xs:element name="otherwise" type="OtherwiseAction" minOccurs="0"/>
      </xs:sequence>
    </xs:complexType>
    <xs:element name="priority-switch" type="PrioritySwitchType"
                substitutionGroup="switch"/>
    <xs:simpleType name="LocationPriorityType">
      <xs:restriction base="xs:float">
        <xs:minInclusive value="0.0"/>
        <xs:maxInclusive value="1.0"/>
      </xs:restriction>
    </xs:simpleType>
    <xs:complexType name="LocationType">
      <xs:group ref="Node"/>
      <xs:attribute name="url" type="xs:anyURI"/>
      <xs:attribute name="priority" type="LocationPriorityType"/>
      <xs:attribute name="clear" type="YesNoType" default="no"/>
    </xs:complexType>
    <xs:complexType name="LookupType">
      <xs:sequence>
        <xs:element name="success" minOccurs="0">
          <xs:complexType>
```

```
            <xs:group ref="Node"/>
          </xs:complexType>
        </xs:element>
        <xs:element name="notfound" minOccurs="0">
          <xs:complexType>
            <xs:group ref="Node"/>
          </xs:complexType>
        </xs:element>
        <xs:element name="failure" minOccurs="0">
          <xs:complexType>
            <xs:group ref="Node"/>
          </xs:complexType>
        </xs:element>
      </xs:sequence>
      <xs:attribute name="source" type="xs:string" use="required"/>
      <xs:attribute name="timeout" type="xs:integer" default="30"/>
      <xs:attribute name="use" type="xs:string"/>
      <xs:attribute name="ignore" type="xs:string"/>
      <xs:attribute name="clear" type="YesNoType" default="no"/>
    </xs:complexType>
    <xs:complexType name="RemoveLocationType">
      <xs:group ref="Node"/>
      <xs:attribute name="location" type="xs:string"/>
      <xs:attribute name="param" type="xs:string"/>
      <xs:attribute name="value" type="xs:string"/>
    </xs:complexType>
    <xs:complexType name="SubAction">
      <xs:attribute name="ref" type="xs:string"/>
    </xs:complexType>
    <xs:complexType name="AncillaryType"/>
    <xs:complexType name="SubactionType">
      <xs:group ref="Node"/>
      <xs:attribute name="id" use="required"/>
    </xs:complexType>
    <xs:complexType name="LogAction">
      <xs:complexContent>
        <xs:extension base="ActionType">
          <xs:group ref="Node"/>
          <xs:attribute name="name" type="xs:string"/>
          <xs:attribute name="comment" type="xs:string"/>
        </xs:extension>
      </xs:complexContent>
    </xs:complexType>
    <xs:element name="log" type="LogAction" substitutionGroup="action"/>
    <xs:element name="cpl">
      <xs:complexType>
        <xs:sequence>
          <xs:element name="ancillary" type="AncillaryType" minOccurs="0"/>
```

```
          <xs:element name="subaction" type="SubactionType" minOccurs="0"/>
          <xs:element ref="trigger" minOccurs="0" maxOccurs="unbounded"/>
        </xs:sequence>
      </xs:complexType>
    </xs:element>
  </xs:schema>
```

The CPL schema for network servers is as below:

```
  <?xml version="1.0" encoding="UTF-8"?>
  <xs:schema targetNamespace="urn:ietf:params:xml:ns:cpl:nserver"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xmlns:xs="http://www.w3.org/2001/XMLSchema"
    xmlns:CPL="urn:ietf:params:xml:ns:cpl"
    xmlns="urn:ietf:params:xml:ns:cpl:nserver"
    elementFormDefault="qualified" attributeFormDefault="unqualified">
    <xs:import namespace="urn:ietf:params:xml:ns:cpl"
      schemaLocation="http://www.ietf.org/internet-drafts/draft-wu-cpl-
schema-02.txt"/>
    <xs:complexType name="IncomingType">
      <xs:complexContent>
        <xs:extension base="CPL:TriggerType"/>
      </xs:complexContent>
    </xs:complexType>
    <xs:element name="incoming" type="IncomingType"
                substitutionGroup="CPL:trigger"/>
    <xs:complexType name="OutgoingType">
      <xs:complexContent>
        <xs:extension base="CPL:TriggerType"/>
      </xs:complexContent>
    </xs:complexType>
    <xs:element name="outgoing" type="OutgoingType"
                substitutionGroup="CPL:trigger"/>
    <xs:simpleType name="OrderingType">
      <xs:restriction base="xs:NMTOKEN">
        <xs:enumeration value="parallel"/>
        <xs:enumeration value="sequential"/>
        <xs:enumeration value="first-only"/>
      </xs:restriction>
    </xs:simpleType>
    <xs:complexType name="ProxyAction">
      <xs:complexContent>
        <xs:extension base="CPL:ActionType">
```

```
            <xs:sequence>
```

```
            <xs:element name="busy" minOccurs="0">
              <xs:complexType>
                <xs:group ref="CPL:Node"/>
              </xs:complexType>
            </xs:element>
            <xs:element name="noanswer" minOccurs="0">
              <xs:complexType>
                <xs:group ref="CPL:Node"/>
              </xs:complexType>
            </xs:element>
            <xs:element name="failure" minOccurs="0">
              <xs:complexType>
                <xs:group ref="CPL:Node"/>
              </xs:complexType>
            </xs:element>
            <xs:element name="redirection" minOccurs="0">
              <xs:complexType>
                <xs:group ref="CPL:Node"/>
              </xs:complexType>
            </xs:element>
            <xs:element name="default" minOccurs="0">
              <xs:complexType>
                <xs:group ref="CPL:Node"/>
              </xs:complexType>
            </xs:element>
          </xs:sequence>
          <xs:attribute name="timeout" type="xs:integer" use="optional"/>
          <xs:attribute name="recursive" type="CPL:YesNoType"
                        use="optional" default="yes"/>
          <xs:attribute name="ordering" type="OrderingType"
                        use="optional" default="parallel"/>
        </xs:extension>
      </xs:complexContent>
    </xs:complexType>
    <xs:element name="proxy" type="ProxyAction"
                substitutionGroup="CPL:action"/>
    <xs:complexType name="RedirectAction">
      <xs:complexContent>
        <xs:extension base="CPL:ActionType">
          <xs:attribute name="permanent" type="CPL:YesNoType" default="no"/>
        </xs:extension>
      </xs:complexContent>
    </xs:complexType>
    <xs:element name="redirect" type="RedirectAction"
                substitutionGroup="CPL:action"/>
    <xs:simpleType name="StatusType">
      <xs:restriction base="xs:NMTOKEN">
        <xs:enumeration value="busy"/>
```

```
        <xs:enumeration value="notfound"/>
        <xs:enumeration value="reject"/>
        <xs:enumeration value="error"/>
      </xs:restriction>
    </xs:simpleType>
    <xs:complexType name="RejectAction">
      <xs:complexContent>
        <xs:extension base="CPL:ActionType">
          <xs:attribute name="status" type="StatusType"/>
          <xs:attribute name="reason" type="xs:string"/>
        </xs:extension>
      </xs:complexContent>
    </xs:complexType>
    <xs:element name="reject" type="RejectAction"
                substitutionGroup="CPL:action"/>
  </xs:schema>
```

4 IANA considerations

4.1 URN Sub-Namespace Registration

This section registers two new XML namespaces, as per the guidelines in [7]

URI: urn:ietf:params:xml:ns:cpl

Registrant Contact: Xiaotao Wu <xiaotaow@cs.columbia.edu>

XML:

```
BEGIN
<?xml version="1.0"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML Basic 1.0//EN"
          "http://www.w3.org/TR/xhtml-basic/xhtml-basic10.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
  <meta http-equiv="content-type"
     content="text/html;charset=iso-8859-1"/>
  <title>Call Processing Language Namespace</title>
</head>
<body>
  <h1>Namespace for Call Processing Language</h1>
  <h2>application/cpl+xml</h2>
  <p>See <a href="[[[URL of published RFC]]]">RFCXXXX</a>.</p>
</body>
```

```
              </html>
              END
```

URI: urn:ietf:params:xml:ns:cpl:nserver

Registrant Contact: Xiaotao Wu <xiaotaow@cs.columbia.edu>

XML:

```
              BEGIN
              <?xml version="1.0"?>
              <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML Basic 1.0//EN"
                        "http://www.w3.org/TR/xhtml-basic/xhtml-basic10.dtd">
              <html xmlns="http://www.w3.org/1999/xhtml">
              <head>
                <meta http-equiv="content-type"
                   content="text/html;charset=iso-8859-1"/>
                <title>Call Processing Language Extensions for Network Servers
Namespace</title>
              </head>
              <body>
                <h1>Namespace for Call Processing Language Extensions for
Network Servers</h1>
                <h2>application/cpl+xml</h2>
                <p>See <a href="[[[URL of published RFC]]]">RFCXXXX</a>.</p>
              </body>
              </html>
              END
```

5 Changes from Earlier Version

5.1 Changes from Draft -01

- Split the original CPL schema into two parts. One is the base schema with common elements applicable to all the entities, such as user agents or presence agents. The other is the schema specifically for network servers, such as proxy or redirect servers.
- Add three abstract elements, namely trigger, switch and action. Trigger is mapped to the top-level call processing action in the original CPL standard. All the new triggers MUST be the substitutionGroup of the abstract trigger element. Switch and action are the same as those defined in the original CPL standard. All the new switches MUST be the substitutionGroup of the abstract switch element. All the new actions MUST be the substitutionGroup of the abstract

action element.

- Add IANA considerations for URN Sub-Namespace registration.
- Remove useless type CommaDelimiterType.
- Separate normal and informal references.

5.2 Changes from Draft -00

- Bug fix, in xs:schema tag, change 'xmlns:tns' to 'xmlns'
- Add example CPL scripts with using the schema in this specification

6 Authors' Addresses

Xiaotao Wu
Dept. of Computer Science
Columbia University
1214 Amsterdam Avenue, MC 0401
New York, NY 10027
USA
electronic mail: xiaotaow@cs.columbia.edu

Henning Schulzrinne
Dept. of Computer Science
Columbia University
1214 Amsterdam Avenue, MC 0401
New York, NY 10027
USA
electronic mail: schulzrinne@cs.columbia.edu

Jonathan Lennox
Dept. of Computer Science
Columbia University
1214 Amsterdam Avenue, MC 0401
New York, NY 10027
USA
electronic mail: lennox@cs.columbia.edu

7 Normative References

[1] J. Lennox and H. Schulzrinne, "CPL: a language for user control of Internet telephony services," internet draft, Internet Engineering Task Force, Nov. 2001. Work in progress.

[2] T. Bray, J. Paoli, C. M. Sperberg-McQueen, and E. Maler, "Extensible markup language (xml) 1.0 (second edition)," W3C Recommendation, World Wide Web Consortium (W3C), Oct. 2000. http://www.w3.org/TR/2000/REC-xml-20001006.

[3] D. C. Fallside, "XML schema part 0: Primer," W3C Recommendation, World Wide Web Consortium (W3C), May 2001. http://www.w3.org/TR/xmlschema-0/.

[4] F. Dawson and D. Stenerson, "Internet calendaring and scheduling core object specification (icalendar)," RFC 2445, Internet Engineering Task Force, Nov. 1998.

[5] R. Moats, "URN syntax," RFC 2141, Internet Engineering Task Force, May 1997.

[6] R. Moats, "A URN namespace for IETF documents," RFC 2648, Internet Engineering Task Force, Aug. 1999.

[7] M. Mealling, "The IETF XML registry," internet draft, Internet Engineering Task Force, July 2002. Work in progress.

Full Copyright Statement

Table of Contents