

Network Working Group
Internet-Draft
Intended status: Experimental
Expires: March 17, 2011

A. Forte
H. Schulzrinne
Columbia University
September 13, 2010

Location-to-Service Translation Protocol (LoST) Extensions
draft-forte-lost-extensions-02.txt

Abstract

An important class of location-based services answer the question "What instances of this service are closest to me?" Examples include finding restaurants, gas stations, stores, automated teller machines, wireless access points (hot spots) or parking spaces. Currently, the Location-to-Service Translation (LoST) protocol only supports mapping locations to a single service based on service regions. This document describes an extension that allows queries of the type "N nearest", "within distance X" and "served by".

Status of this Memo

This Internet-Draft is submitted to IETF in full conformance with the provisions of [BCP 78](#) and [BCP 79](#).

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

The list of current Internet-Drafts can be accessed at <http://www.ietf.org/ietf/lid-abstracts.txt>.

The list of Internet-Draft Shadow Directories can be accessed at <http://www.ietf.org/shadow.html>.

This Internet-Draft will expire on March 17, 2011.

Copyright Notice

Copyright (c) 2010 IETF Trust and the persons identified as the document authors. All rights reserved.

Internet-Draft

LoST Extensions

September 2010

This document is subject to [BCP 78](http://trustee.ietf.org/license-info) and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the BSD License.

Table of Contents

1.	Introduction	3
2.	Requirements Notation	3
3.	Service Regions	3
4.	New Query Types: "N nearest", "within distance X" and "served by"	5
5.	LoST Extensions	5
5.1.	New Use of Shapes in Queries	5
5.2.	Queries Based on Service Regions	6
5.3.	Difference Between "within distance X" and "served by" Queries	8
5.4.	Limiting the Number of Returned Service URIs	8
5.5.	The <serviceLocation> Element in Responses	10
6.	Relax NG Schema	12
7.	Security Considerations	14
8.	IANA Considerations	14
8.1.	LoST Extensions Relax NG Schema Registration	14
8.2.	LoST Extensions Namespace Registration	15
9.	Non-Normative RELAX NG Schema in XML Syntax	15
10.	References	18
	Authors' Addresses	18

[1.](#) Introduction

The Location-to-Service Translation (LoST) protocol [[RFC5222](#)] maps service identifiers (URNs) and civic or geospatial information to service URIs, based on service regions. While motivated by mapping locations to the public safety answering point (PSAP) serving that location, the protocol has been designed to generalize to other location mapping services.

However, the current LoST query model assumes that each service URI has a service region and that service regions do not overlap. This fits the emergency services model, where the service region of a PSAP is given by jurisdictional boundaries, but does not work as well for other services that do not have clearly defined boundaries. For example, any given location is likely served by a number of different restaurants, depending on how far the prospective customer is willing to walk or drive.

We extend LoST with three additional queries, giving the protocol the ability to find the N nearest instances of a particular service, all services within a given radius and all services whose service region includes the client's current location.

The LoST extensions defined in this document do not apply to emergency services.

[2.](#) Requirements Notation

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [[RFC2119](#)].

[3.](#) Service Regions

Generally speaking, service regions apply only to a subset of services.

In [Section 1 of \[RFC5222\]](#) a service region is defined as follows:

"To minimize round trips and to provide robustness against network failures, LoST supports caching of individual mappings and indicates the region for which the same answer would be returned ("service region")."

and in [Section 5.5 of \[RFC5222\]](#):

"A response MAY indicate the region for which the service URL returned would be the same as in the actual query, the so-called service region."

Given that for non-emergency services for each query we could get a large number of service URLs, a service region as per definition above would be the region within which a user would get the same set of service URLs. If one or more of the URLs in the set changes, the set of URLs changes that is, the service region changes. Therefore, for non-emergency services we can distinguish two types of service regions. One is the region served by the single POI (e.g., pizza delivery), the other one is according to the definition in [\[RFC5222\]](#) the region within which the client would always get the same set of service URLs. The second type of service region includes multiple service regions of the first type. Because of this, the service region as defined in [\[RFC5222\]](#) would change very frequently, making the use of service regions as described in [\[RFC5222\]](#) not that useful.

This difference between emergency and non-emergency services is due to the fact that PSAPs have service regions that do not overlap significantly with one another and one service region is served by a single PSAP. As explained above, this is not the case for non-emergency services.

Generally speaking, we can divide location-based services into two main categories.

Services based on:

- o how far they are from the user (e.g., ATM machines, takeout)

- o whether or not their service region includes the client's current location (e.g., pizza delivery, NG9-1-1)

For services included in the first category service regions are not relevant while they are important for services included in the second category. This distinction becomes obvious if we consider the difference between takeout (first category) and delivery (second category), for example. In the case of takeout, the user wants to go to a particular restaurant and buy dinner regardless of his location falling in the restaurant service region or not. For delivery the user cares about the restaurant service region as the restaurant will deliver food to him only if the user's location falls within the restaurant service region.

There is a clear distinction between services that require service regions and services that do not. The LoST extensions defined in this document take this into account by using the service classification mentioned above.

[4.](#) New Query Types: "N nearest", "within distance X" and "served by"

We introduce three new types of queries, "N nearest", "within distance X" and "served by". The first query returns the N points of interest closest to the client's physical location, the second query discovers all those points of interest located within a given distance from the client's physical location, the third query returns all those points of interest whose service region includes the client's current location.

[5.](#) LoST Extensions

For queries "within distance X", the LoST client needs to specify to the server the range within which instances of a particular service should be searched. In order to do this, we make use of various shapes [[PIDF-LO](#)] in LoST queries.

For queries "served by", the LoST client needs to let the server know that it MUST return only those services whose service region includes the client's current location. In order to do this we introduce the <region> element in <findService> queries. Service region boundaries

MAY be returned in a LoST <findServiceResponse> as described in [\[RFC5222\]](#).

For queries "N nearest", the LoST client needs to let the server know N, that is, the maximum number of service URIs to be returned in a response. In order to specify this, we introduce the <limit> element in <findService> queries.

Also, we introduce a new element in LoST responses, namely <serviceLocation>. This new element is used by the server to indicate to the client the physical location of points of interest. In doing so, the client can compute the distance and other metrics between its current location and the points of interest.

The new elements <region>, <limit> and <findService> are defined in the "lostNe" namespace. This new namespace is defined in [Section 6](#).

[5.1](#). New Use of Shapes in Queries

In [\[PIDF-LO\]](#) different shapes are defined in order to represent a point and an area of uncertainty within which the user might be situated. While this remains true for "served by" queries, for "within distance X" queries such shapes represent the area within which we want to find a service.

For example, Figure 1 shows a <findService> geodetic query using the

circular shape. In particular, with the query shown in Figure 1, we are asking the LoST server to send us a list of service URNs for pizza places within 200 meters from our approximate position specified in <p2:pos>.

Aside from the circular shape, other shapes are also useful. In particular, there are situations in which it is useful to query for services in a certain direction of movement rather than in an exact physical location. For example, if a user is driving north from New York City to Boston, it would be useful for this user to be able to query for services north of where he currently is that is, not at his current physical location nor at his final destination.

In order to do this, we use shapes such as an ellipse. The ellipse has a major and a minor dimension thus allowing for defining a

"privileged" direction by having the major dimension in the direction of movement. In the present context the circular shape allows a device to search for services in any direction surrounding its physical location while shapes such as the ellipse allow the device to search for services in a more specific direction. The ellipse shape is defined in [[PIDF-LO](#)].

```
<?xml version="1.0" encoding="UTF-8"?>
<findService
  xmlns="urn:ietf:params:xml:ns:lost1"
  xmlns:p2="http://www.opengis.net/gml"
  serviceBoundary="value"
  recursive="true">
  <location id="6020688f1ce1896d" profile="geodetic-2d">
    <p2:Circle srsName="urn:ogc:def:crs:EPSG::4326">
      <p2:pos>37.775 -122.422</p2:pos>
      <p2:radius uom="urn:ogc:def:uom:EPSG::9001">
        200
      </p2:radius>
    </p2:Circle>
  </location>
  <service>urn:service:local.pizza</service>
</findService>
```

Figure 1: A 'within distance X' <findService> geodetic query using the circular shape

[5.2](#). Queries Based on Service Regions

As mentioned in [Section 1](#), we can divide location-based services into two main categories.

Services based on:

- o how far they are from the user
- o whether or not their service region includes the client's current location

A "within distance X" query addresses services included in the first category while a "served by" query addresses services included in the second category.

When querying LoST regarding a specific service, we need to specify if such service belongs to either the first or the second category. This is necessary since depending on the category the service belongs to, the LoST server has to follow a different metric in selecting the results to include in the response.

In order for the client to specify which of the two categories the service belongs to, we introduce the `<region>` element. This new element is of type boolean. When its value is true it means that the requested service belongs to the second category and a search based on service regions MUST be performed by the LoST server. When either its value is false or the `<region>` element is missing, the LoST server MUST perform a search based on the distance between the user and the points of service.

For a search based on service regions the LoST server MUST return only those services whose service region includes the client's current location. Service region boundaries MAY be returned in a LoST `<findServiceResponse>` as described in [[RFC5222](#)].

```
<?xml version="1.0" encoding="UTF-8"?>
<findService
  xmlns="urn:ietf:params:xml:ns:lost1"
  xmlns:p2="http://www.opengis.net/gml"
  xmlns:ext="http://www.thisisnotdoneyet.net/lostNe"
  serviceBoundary="value" recursive="true">
  <ext:region>true</ext:region>
  <location id="6020688f1ce1896d" profile="geodetic-2d">
    <p2:Circle srsName="urn:ogc:def:crs:EPSG::4326">
      <p2:pos>37.775 -122.422</p2:pos>
      <p2:radius uom="urn:ogc:def:uom:EPSG::9001">
        200
      </p2:radius>
    </p2:Circle>
  </location>
  <service>urn:service:local.pizza</service>
</findService>
```

Figure 2: A 'served by' `<findService>` geodetic query with the new

[5.3.](#) Difference Between "within distance X" and "served by" Queries

Figures 1 and 2 show an example of a "within distance X" query and a "served by" query, respectively. The two types of queries although very similar have three important differences.

- o A "served by" query can support all the shapes a "within distance X" query can support plus the point shape which does not make sense for a "within distance X" query.
- o In a "within distance X" query a shape represents the area within which to search for points of service. In all other types of queries including a "served by" query, a shape represents an uncertainty in the location of the client.
- o In a "within distance X" query the value of the <region> element, if present, MUST be false. If such element is not present, its value MUST be assumed equal to false. A "served by" query SHALL have the value of the <region> element always set to true.

[5.4.](#) Limiting the Number of Returned Service URIs

Limiting the number of results is helpful, particularly for mobile devices with limited bandwidth. For "N nearest" queries, the client needs to be able to tell the server to return no more than N service URIs. In order to specify such limit we introduce a new element, namely <limit>. This new element is OPTIONAL but when present, it MUST be conveyed inside the <findService> element defined in [\[RFC5222\]](#).

Figures 3, 4 and 5 show a <findService> geodetic query where the client asks the server to return no more than 20 service URIs. In particular, Figure 3 shows a 'N nearest' query, Figure 4 shows a query which is a combination of 'N nearest' and 'within distance X' while Figure 5 shows a query which is a combination of 'N nearest' and 'served by'. When receiving such queries, the LoST server will return a list of no more than 20 points of interest.

If the available points of interest are more than N, then the server has to identify among the points of interest to return, the N points of interest closest to the client's physical location and MUST return those in the response.

When the <limit> element is not present in a <findService> query then all available points of interest for the requested type of service SHALL be returned by the LoST server.

```
<?xml version="1.0" encoding="UTF-8"?>
<findService xmlns="urn:ietf:params:xml:ns:lost1"
  xmlns:p2="http://www.opengis.net/gml"
  xmlns:ext="http://www.thisisnotdoneyet.net/lostNe"
  serviceBoundary="value" recursive="true">
  <ext:limit>20</ext:limit>
  <location id="6020688f1ce1896d" profile="geodetic-2d">
    <p2:Point id="point1" srsName="urn:ogc:def:crs:EPSG::4326">
      <p2:pos>40.7128 -74.0092</p2:pos>
    </p2:Point>
  </location>
</service>urn:service:food.pizza</service>
</findService>
```

Figure 3: A 'N nearest' <findService> geodetic query with the new <limit> element

```
<?xml version="1.0" encoding="UTF-8"?>
<findService
  xmlns="urn:ietf:params:xml:ns:lost1"
  xmlns:p2="http://www.opengis.net/gml"
  xmlns:ext="http://www.thisisnotdoneyet.net/lostNe"
  serviceBoundary="value"
  recursive="true">
  <ext:region>false</ext:region>
  <ext:limit>20</ext:limit>
  <location id="6020688f1ce1896d" profile="geodetic-2d">
    <p2:Circle srsName="urn:ogc:def:crs:EPSG::4326">
      <p2:pos>37.775 -122.422</p2:pos>
      <p2:radius uom="urn:ogc:def:uom:EPSG::9001">
        200
      </p2:radius>
    </p2:Circle>
  </location>
  <service>urn:service:local.pizza</service>
</findService>
```

Figure 4: A <findService> geodetic query with the new <limit> and <region> elements. This query is a combination of 'N nearest' and 'within distance X' queries.

```
<?xml version="1.0" encoding="UTF-8"?>
<findService
  xmlns="urn:ietf:params:xml:ns:lost1"
  xmlns:p2="http://www.opengis.net/gml"
  xmlns:ext="http://www.thisisnotdoneyet.net/lostNe"
  serviceBoundary="value"
  recursive="true">
  <ext:region>true</ext:region>
  <ext:limit>20</ext:limit>
  <location id="6020688f1ce1896d" profile="geodetic-2d">
    <p2:Circle srsName="urn:ogc:def:crs:EPSG::4326">
      <p2:pos>37.775 -122.422</p2:pos>
      <p2:radius uom="urn:ogc:def:uom:EPSG::9001">
        100
      </p2:radius>
    </p2:Circle>
  </location>
  <service>urn:service:local.pizza</service>
</findService>
```

Figure 5: A <findService> geodetic query with the new <limit> and <region> elements. This query is a combination of 'N nearest' and 'served by' queries.

[5.5.](#) The <serviceLocation> Element in Responses

It is important for the LoST client to know the location of a point of interest so that distance, route and other metrics can be computed. We introduce a new element, namely <serviceLocation>. The <serviceLocation> element contains the geodetic coordinates of a point of service and SHOULD be used for all non-emergency services. When it is used, it MUST be contained in a <mapping> element. In responses such as <findServiceResponse> [[RFC5222](#)], a list of service URIs, each with its own <serviceLocation> element, SHOULD be returned. The order of service URIs in the list is not relevant.

The <serviceLocation> element has one single attribute, 'profile', in order to specify the profile used. Only geodetic profiles SHOULD be

used as the computation of the distance, route and other metrics would at some point require geocoding of the civic address in geodetic coordinates. Because of this, the position specified in `<serviceLocation>` SHOULD be represented by using the `<Point>` element. The `<Point>` element is described in [Section 12.2 of \[RFC5222\]](#) and in Section 5.2.1 of [\[PIDF-LO\]](#). Figure 6 shows a `<findServiceResponse>` answer containing two location-to-service-URI mappings.

[NOTE: The `<locationUsed>` element cannot be extended for this purpose

as it is defined outside of the `<mapping>` element. In particular, in a response the `<locationUsed>` element is always one, while the number of service URIs is typically more than one.]

There are situations, however, in which it is helpful to include a civic address together with the geodetic coordinates of a point of service. Usually, databases already contain the civic address of points of interest and for devices with limited capabilities it is not always possible to perform decoding of geocoordinates in order to determine the civic address. Because of this, including also the civic address in a response can be useful. In order to do this, it is RECOMMENDED to include the `<civicAddress>` element as defined in [\[RFC5139\]](#) in each `<mapping>` element. Figure 6 shows a `<findServiceResponse>` answer with the `<civicAddress>` element.

```
<?xml version="1.0" encoding="UTF-8"?>
<findServiceResponse xmlns="urn:ietf:params:xml:ns:lost1"
  xmlns:p2="http://www.opengis.net/gml"
  xmlns:ext="http://www.thisisnotdoneyet.net/lostNe">
  <mapping
    expires="2007-01-01T01:44:33Z"
    lastUpdated="2006-11-01T01:00:00Z"
    source="authoritative.example"
    sourceId="7e3f40b098c711dbb6060800200c9a66">
    <displayName xml:lang="it">
      Che bella pizza e all' anima da' pizza da Toto'
    </displayName>
    <service>urn:service:local.pizza</service>
    <uri>sip:chebella@example.com</uri>
    <uri>xmpp:chebella@example.com</uri>
    <serviceNumber>2129397040</serviceNumber>
```

```

<ext:serviceLocation profile="geodetic-2d">
  <p2:Point id="point1" srsName="urn:ogc:def:crs:EPSG:4326">
    <p2:pos>33.665 -112.432</p2:pos>
  </p2:Point>
</ext:serviceLocation>
<civicAddress
  xmlns="urn:ietf:params:xml:ns:pidf:geopriv10:civicAddr">
  <country>US</country>
  <A1>New York</A1>
  <A3>New York</A3>
  <A6>Broadway</A6>
  <HNO>321</HNO>
  <PC>10027</PC>
</civicAddress>
</mapping>
<mapping

```

```

expires="2007-01-01T01:44:33Z"
lastUpdated="2006-11-01T01:00:00Z"
source="authoritative.example"
sourceId="7e3f40b098c711dbb6060800200c9b356">
<displayName xml:lang="en">
  King Mario's Pizza
</displayName>
<service>urn:service:local.pizza</service>
<uri>sip:marios@example.com</uri>
<uri>xmpp:marios@example.com</uri>
<serviceNumber>2129397157</serviceNumber>
<ext:serviceLocation profile="geodetic-2d">
  <p2:Point id="point1" srsName="urn:ogc:def:crs:EPSG:4326">
    <p2:pos>33.683 -112.412</p2:pos>
  </p2:Point>
</ext:serviceLocation>
<civicAddress
  xmlns="urn:ietf:params:xml:ns:pidf:geopriv10:civicAddr">
  <country>US</country>
  <A1>New York</A1>
  <A3>New York</A3>
  <A6>Amsterdam Avenue</A6>
  <HNO>123</HNO>
  <PC>10027</PC>
</civicAddress>

```

```

    </mapping>
    <path>
      <via source="resolver.example"/>
      <via source="authoritative.example"/>
    </path>
    <locationUsed id="6020688f1ce1896d"/>
  </findServiceResponse>

```

Figure 6: A <findServiceResponse> answer

6. Relax NG Schema

This section provides the Relax NG schema of LoST extensions in the compact form. The verbose form is included in [Section 9](#).

```

namespace a = "http://relaxng.org/ns/compatibility/annotations/1.0"
default namespace ns1 = "urn:ietf:params:xml:ns:lostNe"

```

```

##
##   Extensions to the Location-to-Service Translation (LoST)

```

```

##   Protocol

##
##   LoST Extensions define three new elements: limit, region and
##   serviceLocation.
##
start =
  limit
  | region
  | serviceLocation

##
##   A limit to the number of returned results.
##
div {
  limit=
    element limit {
      xsd:positiveInteger

```

```

    }
}

##
##   A boolean variable to request a search
##   based on either service regions or distance.
##
div {
    region=
    element region {
        xsd:boolean
    }
}

##
##   Location Information
##
div {
    locationInformation =
    extensionPoint+,
    attribute profile { xsd:NMTOKEN }?
}

##
##   Location Information about the returned point
##   of service.
##
div {
    serviceLocation=
    element serviceLocation { locationInformation }+

```

```

}

##
##   Patterns for inclusion of elements from schemas in
##   other namespaces.
##
div {

    ##
    ##   Any element not in the LoST Extensions
    ##   namespace.

```

```

##
notLostExt = element * - (ns1:* | ns1:*) { anyElement }

##
##      A wildcard pattern for including any element
##      from any other namespace.
##
anyElement =
  (element * { anyElement }
   | attribute * { text }
   | text)*

##
##      A point where future extensions
##      (elements from other namespaces)
##      can be added.
##
extensionPoint = notLostExt*
}

```

[7.](#) Security Considerations

The same security considerations as in [\[RFC5222\]](#) apply.

[8.](#) IANA Considerations

[8.1.](#) LoST Extensions Relax NG Schema Registration

URI: urn:ietf:params:xml:schema:lostNe

Registrant Contact: Andrea G. Forte, andrea@cs.columbia.edu, Henning Schulzrinne, hgs@cs.columbia.edu

Relax NG Schema: The Relax NG schema to be registered is contained in

[Section 6.](#) Its first line is

```
default namespace ns1 = "urn:ietf:params:xml:ns:lostNe"
```


and its last line is

```
}
```

[8.2.](#) LoST Extensions Namespace Registration

URI: urn:ietf:params:xml:ns:lost1:ext

Registrant Contact: Andrea G. Forte, andrea@cs.columbia.edu, Henning Schulzrinne, hgs@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>LoST Extensions Namespace</title>
</head>
<body>
  <h1>Namespace for LoST Extensions</h1>
  <h2>urn:ietf:params:xml:ns:lostNe</h2>
<p>See <a href="http://www.rfc-editor.org/rfc/rfcXXXX.txt">
  RFCXXXX</a>.</p>
</body>
</html>
END
```

[9.](#) Non-Normative RELAX NG Schema in XML Syntax

```
<?xml version="1.0" encoding="UTF-8"?>
  <grammar ns="urn:ietf:params:xml:ns:lostNe"
    xmlns="http://relaxng.org/ns/structure/1.0"
    xmlns:a="http://relaxng.org/ns/compatibility/annotations/1.0"
    datatypeLibrary="http://www.w3.org/2001/XMLSchema-datatypes">

    <start>
```

```
<a:documentation>
  Extensions to the Location-to-Service Translation (LoST)
  Protocol.
  LoST Extensions define three new elements: limit, region and
  serviceLocation.
</a:documentation>
<choice>
  <ref name="limit"/>
  <reference name="region"/>
  <ref name="serviceLocation"/>
</choice>
</start>

<div>
  <a:documentation>
    A limit to the number of returned results.
  </a:documentation>

  <define name="limit">
    <element name="limit">
      <data type="positiveInteger"/>
    </element>
  </define>
</div>

<div>
  <a:documentation>
    A boolean variable to request a search
    based on either service regions or distance.
  </a:documentation>

  <define name="region">
    <element name="region">
      <data type="boolean"/>
    </element>
  </define>
</div>

<div>
  <a:documentation>
    Location Information
  </a:documentation>

  <define name="locationInformation">
    <oneOrMore>
      <ref name="extensionPoint"/>
    </oneOrMore>
  </define>
</div>
```

<optional>

Internet-Draft

LoST Extensions

September 2010

```
        <attribute name="profile">
          <data type="NMTOKEN"/>
        </attribute>
      </optional>
    </define>
  </div>

  <div>
    <a:documentation>
      Location Information about the returned point of service.
    </a:documentation>

    <define name="serviceLocation">
      <element name="serviceLocation">
        <ref name="locationInformation"/>
      </element>
    </define>
  </div>

  <div>
    <a:documentation>
      Patterns for inclusion of elements from schemas in
      other namespaces.
    </a:documentation>

    <define name="notLostExt">
      <a:documentation>
        Any element not in the LoST Extensions namespace.
      </a:documentation>
      <element>
        <anyName>
          <except>
            <nsName ns="urn:ietf:params:xml:ns:lostNe"/>
            <nsName/>
          </except>
        </anyName>
        <ref name="anyElement"/>
      </element>
    </define>
```

```

<define name="anyElement">
  <a:documentation>
    A wildcard pattern for including any element
    from any other namespace.
  </a:documentation>
  <zeroOrMore>
    <choice>
      <element>

```

```

      <anyName/>
      <ref name="anyElement"/>
    </element>
    <attribute>
      <anyName/>
    </attribute>
    <text/>
  </choice>
</zeroOrMore>
</define>

<define name="extensionPoint">
  <a:documentation>
    A point where future extensions
    (elements from other namespaces)
    can be added.
  </a:documentation>
  <zeroOrMore>
    <ref name="notLostExt"/>
  </zeroOrMore>
</define>
</div>

</grammar>

```

10. References

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", [BCP 14](#), [RFC 2119](#), March 1997.
- [RFC5222] Hardie, T., Newton, A., Schulzrinne, H., and H.

Tschofenig, "LoST: A Location-to-Service Translation Protocol", [RFC 5222](#), August 2008.

- [RFC5139] Thomson, M. and J. Winterbottom, "Revised Civic Location Format for Presence Information Data Format Location Object (PIDF-LO)", [RFC 5139](#), February 2008.
- [PIDF-LO] Winterbottom, J., Thomson, M., and H. Tschofenig, "GEOPRIV PIDF-LO Usage Clarification, Considerations and Recommendations. IETF Internet Draft (Work in Progress)", February 2008.

Forte & Schulzrinne

Expires March 17, 2011

[Page 18]

Internet-Draft

LoST Extensions

September 2010

Authors' Addresses

Andrea G. Forte
Columbia University
Department of Computer Science
1214 Amsterdam Avenue, MC 0401
New York, NY 10027
USA

Email: andreaf@cs.columbia.edu

Henning Schulzrinne
Columbia University
Department of Computer Science
1214 Amsterdam Avenue, MC 0401
New York, NY 10027
USA

Email: hgs@cs.columbia.edu

