ULEX Message Framework Overview

This document briefly describes the ULEX (Universal Lexical Exchange) Message Framework. ULEX provides an extensible framework for consistent packaging of information for the purposes of information sharing. This message exchange format is based on a plug-in architecture, so different programs can provide program-specific implementations of the exchange standard concepts. The framework defines major message exchange objects, hierarchy, and structure of the elements. A brief explanation for the architecture of the framework is provided below.

ULEX message framework:

- Defines top level objects (such as Message, Data Item, Digest, and Attachment).
- Provides conceptual business rules that connect ULEX structures together.
- Uses W3C XML Schema abstract and group substitution constructs to provide extension points for domain specific extensions and implementations.

Figure 1 illustrates the top level structure for the PublishMessage submission:

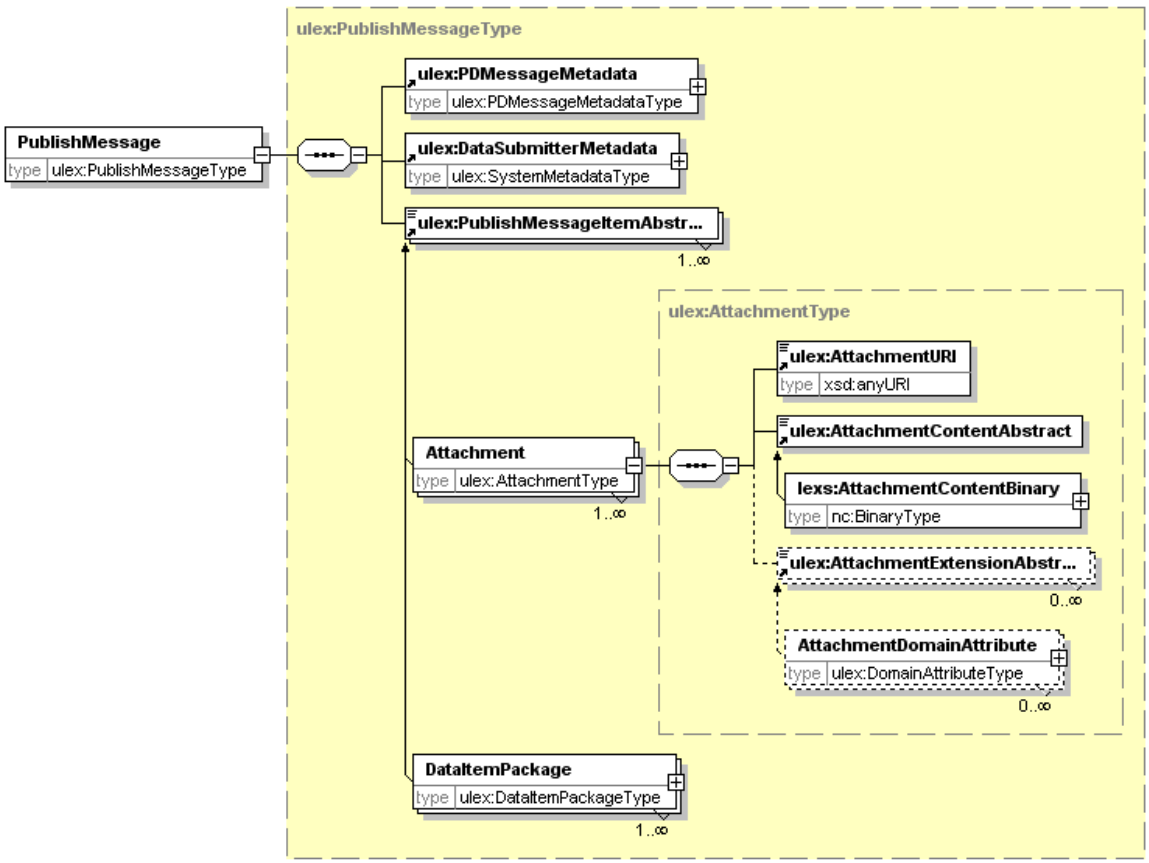

Figure 1. ULEX Message Framework

ULEX message framework defines message structures and major components common to all ULEX based exchanges. Framework also defines specific points where the definition and structure of the content is determined by the particular implementation. For example, for all ULEX based exchanges, PublishMessage will contain PDMessageMetadata and DataSubmitterMetadata elements, plus one or more publish message items, each of which can be either a DataItemPackage or an Attachment element. On the other hand, a ULEX Attachment contains AttachmentURI and AttachmentContent elements, but the format for the AttachmentContent element is not defined in the ULEX Message Framework. AttachmentContent is declared as an abstract element and is an example of an extension point left for concrete domain implementation.

An example of a concrete, domain-specific implementation is shown below. Note that the DataItemPackage structure shown in Figure 2 contains a DigestAbstract.

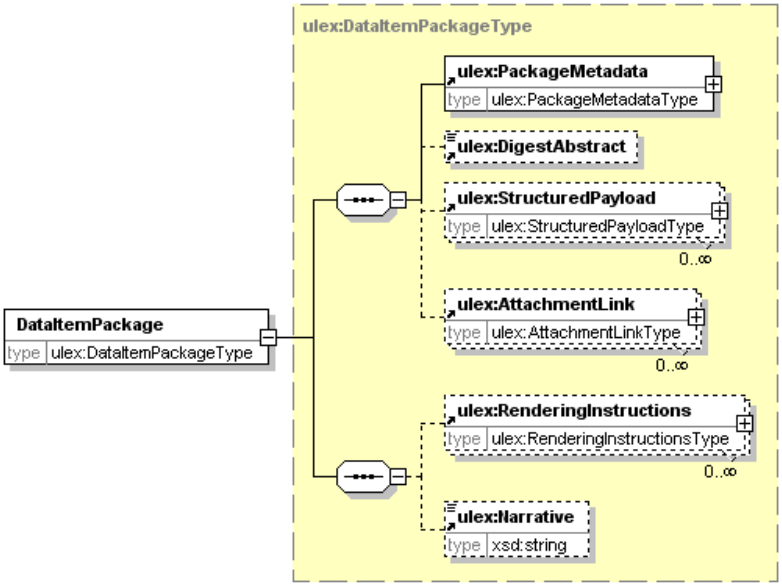

Figure 2. ULEX DataItemPackage

The DigestAbstract element is declared in the ULEX Message Framework schema as shown below:

```
<xsd:element name="DigestAbstract" abstract="true" nillable="false"/>
```

ULEXS Message Framework Code Snippet

Each program will create its own implementation of the DigestAbstract as part of its ULEX-based standard. For example, the Department of Justice Law Enforcement Information Sharing Program (LEISP) has an LEISP Exchange Specification (LEXS). LEXS exchanges could be based on ULEX and could implement the DigestAbstract element as shown below:

```
<xsd:element name="Digest" type="lexs:DigestType" nillable="false" substitutionGroup="ulex:DigestAbstract"/>
 <xsd:complexType name="DigestType">
      <xsd:annotation>
        <xsd:documentation>A structure that describes LEXS Digest format. Structure contains LEXS Entities and
Associations between Entities. While LEXS Entities are children of Digest element and don't follow any particular order
, all LEXS Associations are wrapped in an element "Associations" and follow strict sequence.</xsd:documentation>
        <xsd:appinfo>
          <i:Base i:namespace="http://niem.gov/niem/structures/2.0" i:name="Object"/>
        </xsd:appinfo>
      </xsd:annotation>
      <xsd:complexContent>
        <xsd:extension base="s:ComplexObjectType">
          <xsd:sequence>
            <xsd:element ref="lexsdigest:Entity" maxOccurs="unbounded"/>
            <xsd:element ref="lexsdigest:Associations" minOccurs="0"/>
          </xsd:sequence>
        </xsd:extension>
      </xsd:complexContent>
</xsd:complexType>
```

LEXS Implementation

The base ULEX framework does not contain any references to LEXS structures; however it allows inclusion of LEXS-based content, or content defined by other programs. It is important to point out that messages constructed from implementations defined by different programs will not necessarily be compatible with each other since, for example, a Person object definition used in the message Digest defined by the LEXS program will possibly differ from a Person object definition in UCore. However, since the generic structure of the message is the same, the interoperability task could be significantly simplified in the future. Figure 3 is a high level diagram that reflects the correlation between the ULEX framework schema, its concrete LEXS program-specific implementation, and the corresponding Publish Discover Message. Namespaces code snippets illustrate schemas and building blocks dependencies required for ULEX message implementation.

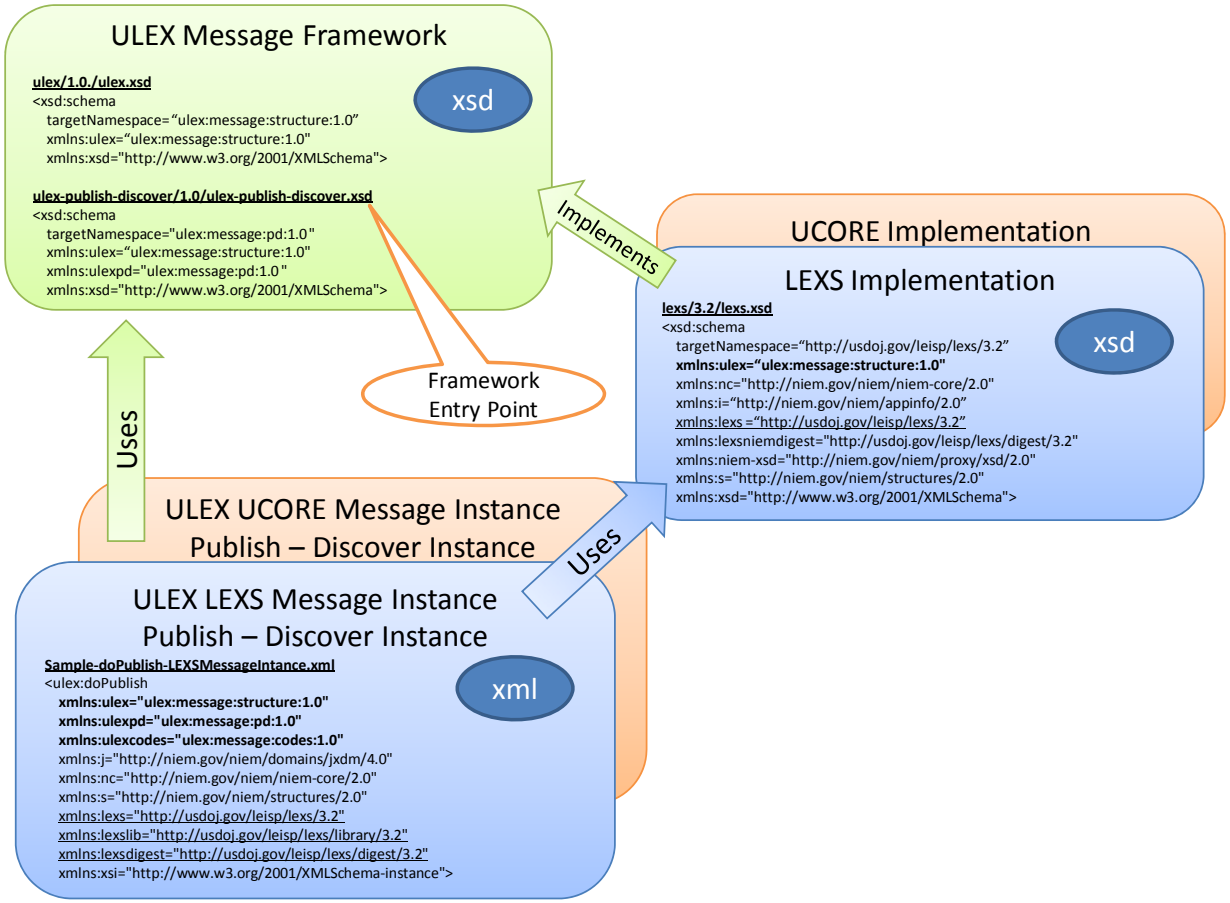

Figure 3. ULEX Message Framework Implementation

The resulting diagram for a data item package corresponding to the LEXS implementation of the ULEX data item package from Figure 2 is shown below.

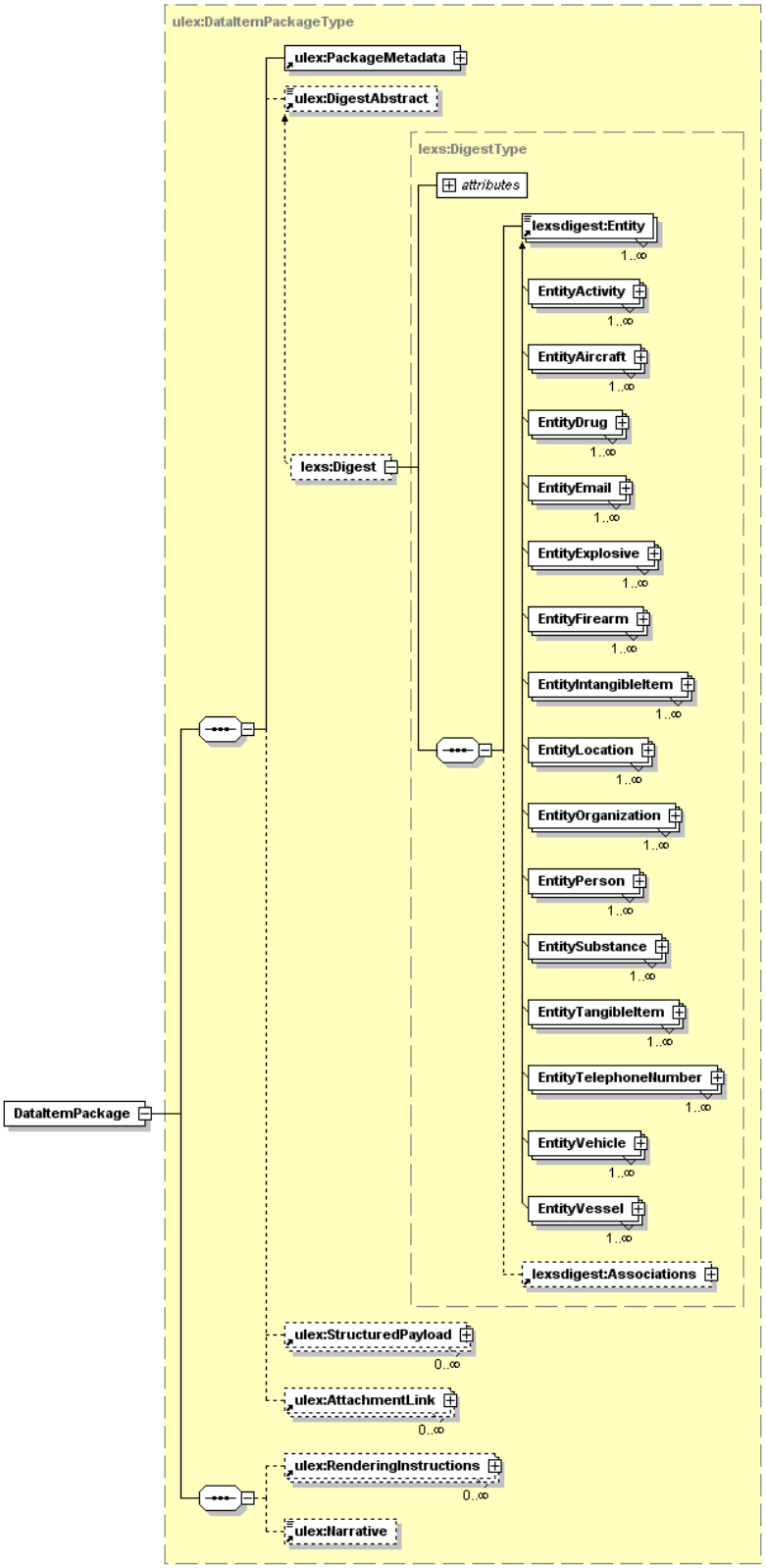

Figure 4. LEXS DataItemPackage

Key Framework Concepts

Extension Points

The framework includes a number of abstract elements containing the word "Extension", such as PDMessageMetadataExtensionAbstract. These extension points provide a mechanism for adding content beyond what is available in the ULEX framework. If a program exchange needs to add multiple elements to ULEX defined structures it may substitute ULEX extension points for its own elements. Extension points are defined as multiply occurring. ULEX-based implementations are required to use the structures provided by the program standard. These abstract substitutions are generally used by programs defining a standard for use by their constituencies. ULEX framework provides example of such an extension (PDMessageMetadataDomainAttribute) that can be used by communities or projects for content augmentation without predefined program-specific elements and extensions (see next section on Domain Attributes and Figure 5 for an example).

Domain Attributes

ULEX includes Domain Attribute elements such as PDMessageMetadataDomainAttribute. Domain Attribute elements are of the generic ulex:DomainAttributeType and are placed at the object's extension point. For example, PDMessageMetadataDomainAttribute belongs to the substitution group for the PDMessageMetadataExtensionAbstract. These elements are intended for use by communities or projects in order to augment the contents of ULEX framework elements as well as ULEX-based standard elements. Since the program-specific schemas may not be altered to add content, the Domain Attribute elements provide a mechanism for an instance document to include additional content without requiring any modification to schemas. Domain Attribute elements includes a mandatory element used to indicate the domain, or community, or project for which this instance element applies. The Domain Attribute elements also include attribute/value pairs so instances can specify the name of the attribute and its value. For example, if an instance needs to indicate a region for which the information applies, the instance could include a PDMessageMetadataDomainAttribute with the AttributeName set to "Region" and the AttributeValue set to "Southeast". The Domain Attribute also includes an "xsd:any" element to handle cases where the attribute is complex and has its own structure.

For an example of the Extension Points and the Domain Attributes elements see Figure 5 below.

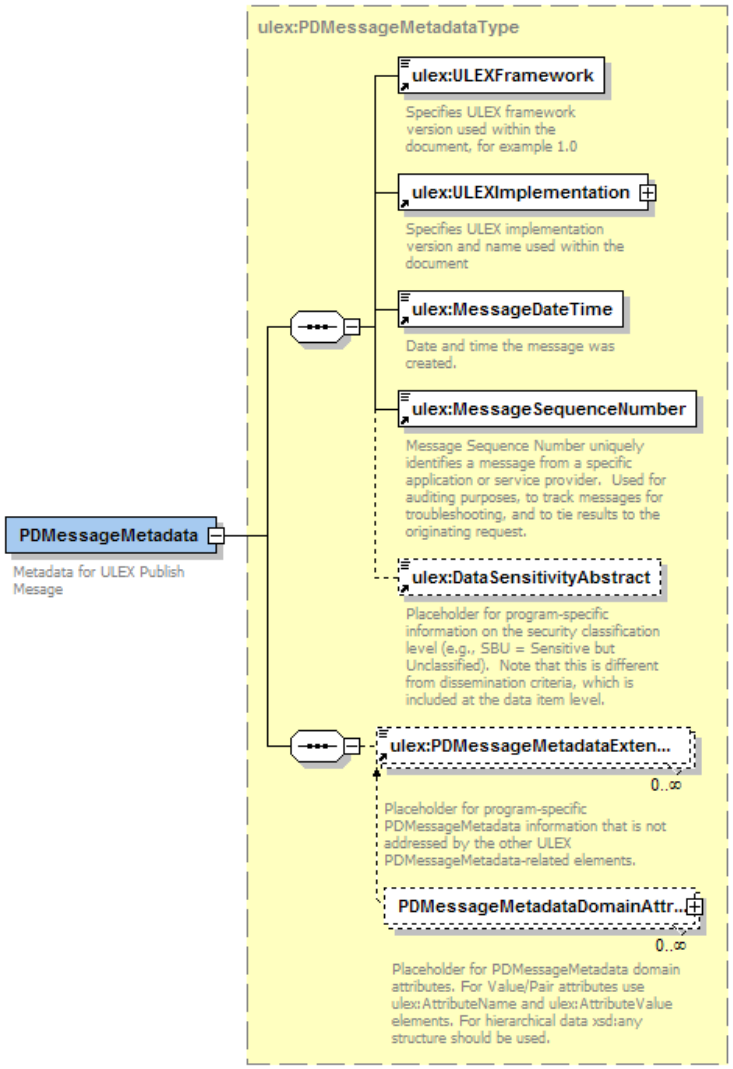

Figure 5 ULEX Extension Points and Domain Attributes.

Framework Inviolability

The framework schemas are defined by ULEX and may not be modified in any way by a program defining a ULEX-based standard, or by communities, or individual projects. The ULEX schemas must remain unchanged in order to maintain interoperability across ULEX implementations and ULEX-based standards. Abstract elements have been provided for programs to use to extend ULEX, however, the ULEX schemas must be used as defined in their entirety by any programs desiring to use ULEX.