PSV Aggregator/Distributor Message Package
An Implementation of PSV

Version 1.2

Guide to the PSV Aggregator/Distributor Message Package

August 18, 2014

Table of Contents

1 STATUS

1.1 Document Status

The status of this document is:

✓	Draft for Public Comment	June 29, 2012
✓	Specification Release	September 27, 2012
✓	Specification Release	May 31, 2013
✓	Specification Release	August 18, 2014

1.2 Document Location

The location of this document
is:1http://www.prismstandard.org/specifications/psv/1.2/pamp_guide_1.1.pdf

The location of the previous versions of this document is
http://www.prismstandard.org/specifications/psv/1.0/pamp_guide_1.1.pdf
http://www.prismstandard.org/specifications/psv/1.0/pamp_guide_1.0.pdf

1.3 Version History

Version Number	Release Date	Editor	Description
PAMP 1.0	June 30, 2012	Kennedy	Public Draft
PAMP 1.0	September 27, 2012	Kennedy	Version 1.0 Release
PAMP 1.1	May 31, 2013	Ichikawa /Kennedy	Version 1.1 Release
PAMP 1.2	August 18, 2014	Kennedy	Version 1.2 Release

2 INTRODUCTION

2.1 Background of the Aggregator/Distributor Message

This implementation of the PRISM Source Vocabulary was developed by the Japanese Magazine Publishing Association in collaboration with members of the IDEAlliance nextPub Initiative and the PRISM Working Group. This PSV Implementation is being published as a component of the PRISM Source Vocabulary Specification documentation package and is open for international comment and adoption.

The PSV Aggregator/Distributor Message Package was developed to meet three use cases documented in this Guide. These use cases represent the magazine and news publishing model found today in Japan. However use of this implementation of PSV is not limited to Japan.

2.1.1 What is an Aggregator/Syndicator?

In North America aggregators and syndicators are synonymous. The aggregator/syndicator purchases rights to content from multiple publishers for the purpose of combining content across magazine titles and offering the collection of content to the public as a distinct product offering. In this business model, the aggregator/syndicator serves as a secondary content publisher.

2.1.2 What is an Aggregator/Distributor?

In Japan, an aggregator is synonymous with a distributor. The business model in Japan is for the publisher to create and produce magazine and news content and for an aggregator/distributor to transform and deliver the magazine or newspaper (including tablet editions) to the public.

> **Note:** An aggregator/syndicator serves a very different function from an aggregator/distributor. This Guide documents the PSV-based XML message used to deliver content from a publisher to an aggregator/distributor. This PSV implementation is not intended to be used to implement a use case based on an aggregation/syndication business model.

2.2 About The Guide to the PSV Aggregator/Distributor Message Package

The PSV Aggregator/Distributor Message Package Guide [PAMPGUIDE] document describes how to use the PSV metadata fields and pamP XML messaging tags to deliver content to content aggregators/distributors. The Guide documents the pamP XML message structure and provides the pamP XSD and document samples.

2.3 How Can I Use this Guide?

This guideline document describes how to apply PSV-based XML tagging and PRISM 3.0 metadata fields to deliver content to aggregator/distributors.

The *pamP Guide* documents the XML tagging to support three distinct use cases or implementation models (IMs):

- **IM-1:** Magazine and news publishers wish to deliver a PSV-based XML message containing publication/issue and optionally article metadata with an external link to a downloadable issue PDF from a server or CMS.
- **IM-2:** Magazine and news publishers wish to deliver a PSV-based XML message containing publication/issue metadata with metadata for each article of the issue along with an external link to a downloadable PDF for each article from a server or CMS.
- **IM-3:** In the future, magazine and news publishers wish to deliver a PSV-based XML message containing publication/issue metadata with metadata for each article of the issue along with PSV-tagged article content.

2.4 Supporting Documentation

Both the PRISM and PRISM Source Vocabulary Specifications are required to support the JMPA PSV Aggregator Message Package. The PRISM Documentation Package consists of:

2.4.1 PRISM Source Vocabulary Documentation

nextPub has developed a series of specifications collectively known as the PRISM Source Vocabulary. The use case for PSV is to encode semantically rich content for transformation and delivery to any platform. This Specification is made up of a modular documentation package that builds on PRISM 3.0 and HTML5. Over time new modules may be added to the documentation package. The documentation package for the nextPub PRISM Source Vocabulary Specification Version 1.0 consists of:

Document	Description
PRISM Source Vocabulary Specification Overview [PSVSO] http://www.prismstandard.org/specifications/psv/1.0/PSV_overview.pdf or http://www.prismstandard.org/specifications/psv/1.0/PSV_overview.htm	The Introduction to the PRISM Source Vocabulary provides an introduction and a non-technical overview of the PRISM Source Vocabulary.
PRISM Source Vocabulary Specification [PSVS] http://www.prismstandard.org/specifications/psv/1.0/PSV.pdf or http://www.prismstandard.org/specifications/psv/1.0/PSV.htm	The PRISM Source Vocabulary Specification defines semantically rich for source metadata and content markup that can be transformed and served to a wide variety of output devices including eReaders, mobile tablet devices, smartphones and print.
PRISM Source Vocabulary Markup Specification [PSVMS] http://www.prismstandard.org/specifications/psv/1.0/PSV_markkup.pdf or http://www.prismstandard.org/specifications/psv/1.0/PSV_markup.htm	The PSV Markup Specification documents the XML tags in the PSV namespace that are used to encode XML Source Content.
PAM to PSV_Guide [PAMPSVGUIDE] http://www.prismstandard.org/specifications/psv/1.0/PAM_PSV.pdf or http://www.prismstandard.org/specifications/psv/1.0/PAM_PSV.htm	This Guide documents mappings from PAM XML to PSV XML. It is normative only.

2.4.2 PRISM 3.0 General Documents

This is a set of general or overview documents that apply to PRISM.

Document	Description
PRISM Introduction [PRISMINT] http://www.prismstandard.org/specifications/3.0/PRISM_introduction_3.0.pdf or http://www.prismstandard.org/specifications/3.0/PRISM_introduction_3.0.htm	Overview, background, purpose and scope of PRISM; examples; contains no normative material.
PRISM Compliance [PRISMCOMP] http://www.prismstandard.org/specifications/3.0/	Describes three profiles of PRISM compliance for content and systems; includes normative

PRISM_compliance_3.0.pdf or http://www.prismstandard.org/specifications/3.0/ PRISM_compliance_3.0.htm	material.

2.4.3 PRISM 3.0 Metadata Specifications

This is the set of documents that outline the prism metadata fields and values by PRISM metadata category. PRISM has modularized its metadata specification by namespace so users may pick those modules that meet their unique business requirements without having to implement the entire PRISM specification.

Document	Description
The PRISM Basic Metadata Specification [PRISMBMS] http://www.prismstandard.org/specifications/3.0/ PRISM_Basic_Metadata_3.0.pdf or http://www.prismstandard.org/specifications/3.0/ PRISM_Basic_Metadata_3.0.htm	Describes the basic metadata elements contained in the PRISM namespace to describe article content; includes normative material.
PRISM Advertising Metadata Specification [PRISMADMS] http://www.prismstandard.org/specifications/3.0/ PRISM_Advertising_Metadata_3.0.pdf or http://www.prismstandard.org/specifications/3.0/ PRISM_Advertising_Metadata_3.0.htm	Describes advertising metadata elements including those drawn from AdsML, GWG and Ad-ID; includes normative material.
The PRISM Subset of Dublin Core Metadata Specification [PRISMDCMS] http://www.prismstandard.org/specifications/3.0/ PRISM_Dublin_Core_Metadata_3.0.pdf or http://www.prismstandard.org/specifications/3.0/ PRISM_Dublin_Core_Metadata_3.0.htm	Describes the metadata elements from the Dublin Core namespace that are included in PRISM; includes normative material.
The PRISM Image Metadata Specification [PRISMIMS] http://www.prismstandard.org/specifications/3.0/ PRISM_Image_Metadata_Specification_3.0.pdf or http://www.prismstandard.org/specifications/3.0/ PRISM_Image_Metadata_Specification_3.0.htm	Describes the metadata elements contained in the PRISM Metadata for Images Namespace and other related image namespaces, includes normative material.
The PRISM Recipe Metadata Specification [PRISMRMS] http://www.prismstandard.org/specifications/3.0./ PRISM_Recipe_Metadata_3.0.pdf or http://www.prismstandard.org/specifications/3.0./ PRISM_Recipe_Metadata_3.0.htm	Describes the metadata elements contained in the PRISM Recipe Metadata Namespace, includes normative material
The PRISM Usage Rights Metadata Specification [PRISMURMS] http://www.prismstandard.org/specifications/3.0/ PRISM_Usage_Rights_Metadata_3.0.pdf or http://www.prismstandard.org/specifications/3.0/ PRISM_Usage_Rights_Metadata_3.0.htm	Describes the metadata elements contained in the PRISM Usage Rights Namespace; includes normative material. This namespace will supersede elements in both the prism: and prl: namespaces in version 3.0 of the specification.

2.4.4 PRISM 3.0 Controlled Vocabulary Specifications

These modules are new with PRISM 3.0. All controlled vocabularies and their terms are documented in this publication set.

Document	Description
The PRISM Controlled Vocabulary Markup Specification [PRISMCVMS] http://www.prismstandard.org/specifications/3.0/ PRISM_Controlled_Vocabulary_Markup_3.0.pdf or http://www.prismstandard.org/specifications/3.0/ PRISM_Controlled_Vocabulary_Markup_3.0.htm	Describes the metadata fields in the PRISM Controlled Vocabulary Namespace that can be used to describe a controlled vocabulary. Actual PRISM controlled vocabularies are now placed in the PRISM Controlled Vocabularies Specification [PRISMCVS]
The PRISM Controlled Vocabularies Specification [PRISMCVS] http://www.prismstandard.org/specifications/3.0/ PRISM_CV_Spec_3.0.pdf or http://www.prismstandard.org/specifications/3.0/ PRISM_CV_Spec_3.0.htm	The PRISM Controlled Vocabularies are now documented in this document.

2.4.5 Access to Supporting Documentation

The PRISM Source Vocabulary Documentation Package is publicly and freely available on the PSV website, www.psvstandard.org. Documentation for PRISM is available on the PRISM website at www.prismstandard.org.

2.5 Legend for Diagrams

In this guide, the XML model is often illustrated by a model diagram. Each diagram was produced with the XML Spy product. These diagrams show the elements and attributes that make up a model and their order and frequency.

The legend for reading XML model diagrams is shown in Figure 2.1. Elements that are required by the model are shown in a solid box. Elements that are optional are shown in a dotted box. Likewise attributes may be required (solid box) or optional (dotted box). A repeatable occurrence of elements is indicated by numbers below each element box to the right.

The diagrams also indicate how elements are assembled. When building some models, elements may occur in a sequence with a specified order. Other models provide a choice from among a number of elements. The legend in Figure 2.1 shows the connectors for sequence and choice.
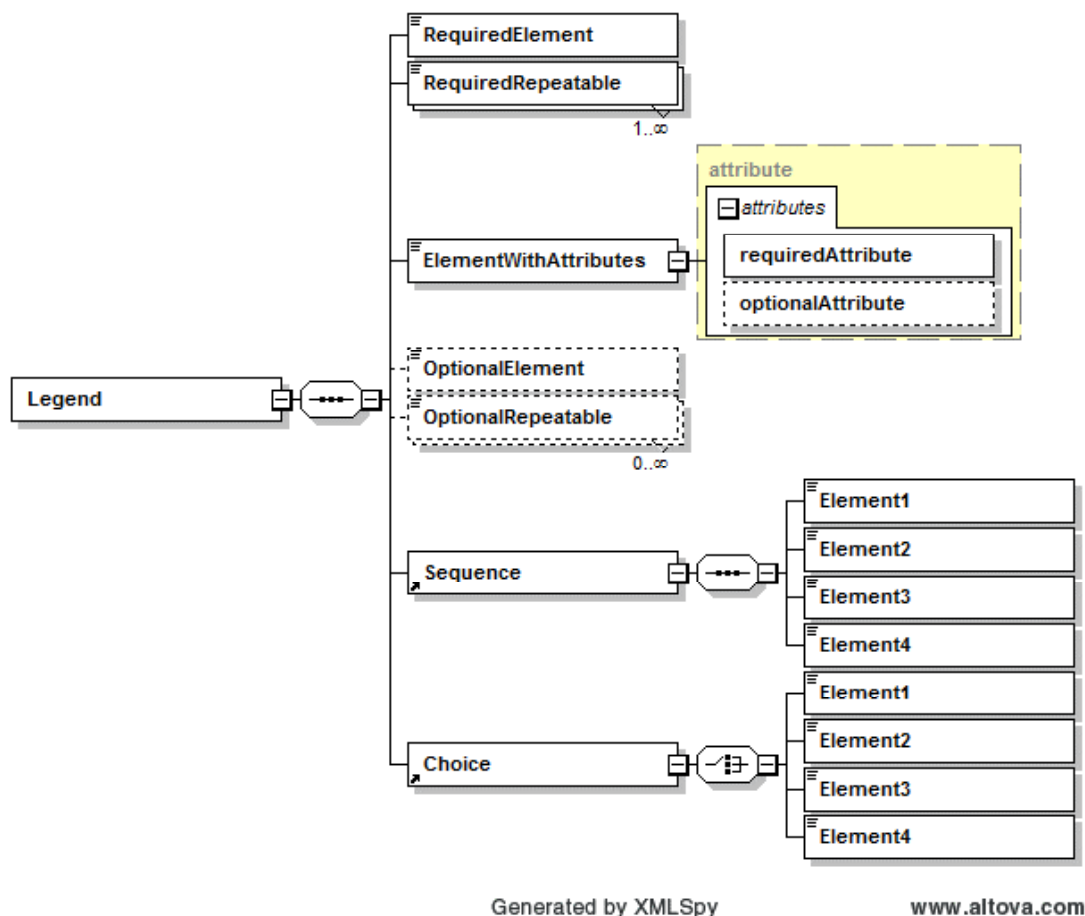
Figure 2.1 Legend for XML Diagrams

2.6 Relationship to PRISM Source Vocabulary

The PSV Aggregator/Distributor Message Package builds upon the PRISM Source Vocabulary, an XML tag set for encoding semantically rich, platform-agnostic source publication content and metadata for delivery across platforms and channels. The PSV tag set includes mechanisms for encoding source magazine content and its metadata. See the PRISM Source Vocabulary Specification [PSVS] for more information.

2.7 Relationship to HTML5

HTML5 is gaining credibility as a delivery platform for publications, not only on the Web but as an alternative to the publication apps that we see today on tablets and smartphones. The IDEAlliance PRISM Source Vocabulary Specification Working Group has decided to come into as close compliance with the XML Serialization of HTML5 as is possible. Hence HTML5 serves as the foundation for tagging PSV content.

2.8 The Relationship to PRISM

The PSV Aggregator/Distributor Message Package builds upon PSV that builds upon the foundation of PRISM 3.0 Specifications and relies on the metadata fields and controlled vocabularies defined by PRISM. pamP defines an XML message for delivering metadata, links to issue and article PDFs and for tagging source content, but it *does not* define its own metadata fields or controlled vocabularies.

2.9 Relationship to the PRISM Aggregator Message (PAM)

In 2004, The PRISM Working Group defined an XML tag set for encoding metadata and article content for delivery to aggregators/syndicators. This tag set, known as the PRISM Aggregator Message, or PAM has been widely implemented in North America. This PSV Aggregator/Distributor Message Package Specification is designed to deliver content to aggregator/distributors in Japan. It has a different use case than the original PAM and is based on newer technology, including PRISM Source Vocabulary (PSV), PRISM 3.0 and HTML5. The pamP should not be confused with the original PRISM Aggregator Message. See Figure 2.1.
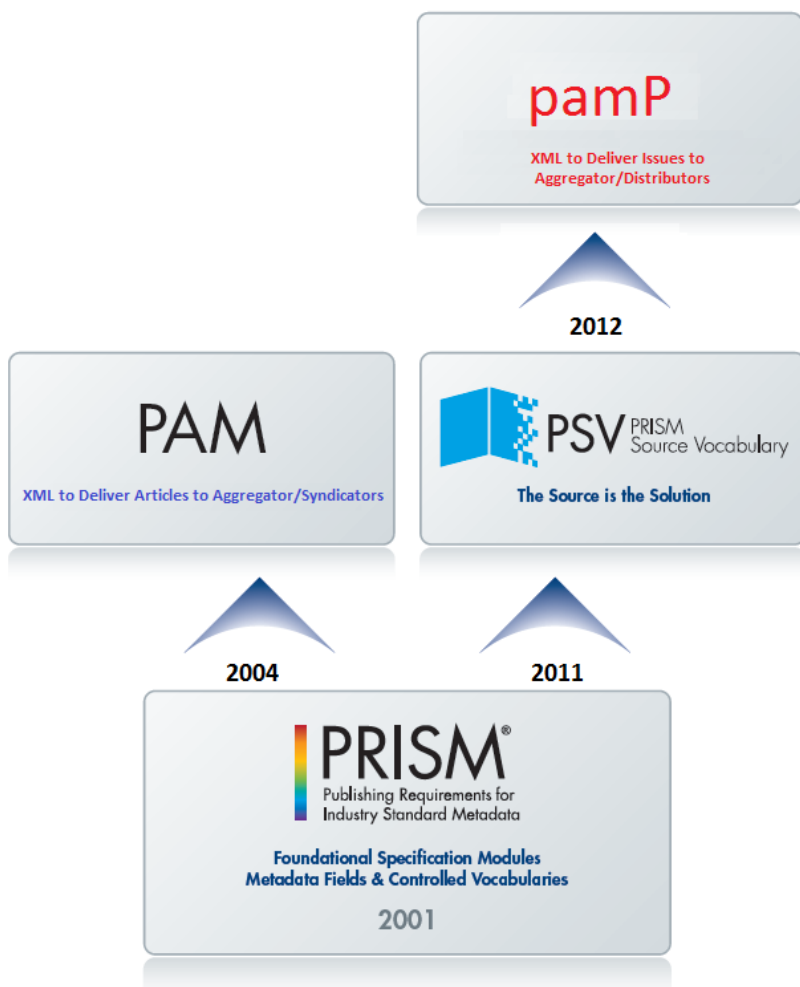
Figure 2.1 pamP Origins

3 THE PAMP XML MESSAGE MODEL

3.1 PSV Aggregator/Syndicator Message Package Use Cases

Three use cases, or implementation models, have been identified for The PSV Aggregator/Distributor Message Package. The use cases are documented below. Note that the XML Model has been defined to support all three use cases.

3.1.1 Use Case 001: Deliver a Complete PDF Issue

Title	Deliver a Complete PDF Issue to Aggregators
Summary:	Magazine and news publishers wish to deliver a PSV-based XML message containing publication/issue and optionally article metadata with an external link to a downloadable issue PDF on a server or CMS. In addition, the publisher wants to deliver additional supporting content such as the cover, table of contents or sample pages from articles
Actors:	Publishers, Aggregator/Distributor
Prerequsites:	
Scenario:	1. Publishers send a message designed to "deliver" a complete PDF magazine/newspaper issue along with supporting PDFs such as the cover, contents or sample article pages along with metadata to aggregators. 2. Aggregators process the message and download a complete PDF issue and supporting PDFs from a server or CMS.
-Requirements	1. The message must identify the publisher, publication and issue of content being delivered. 2. The issue PDF must be available on a server or CMS for download. 3. The message must be based on the PRISM Source Vocabulary 1.0..

3.1.2 Use Case 002: Deliver a Complete Issue Made Up of Single-Article PDFs

Title	Deliver a Complete Issue Made Up of single-article PDFs
Summary:	Magazine and news publishers wish to deliver a PSV-based XML message containing publication/issue metadata with metadata for each article of the issue along with an external link to a downloadable PDF for each article from a server or CMS.
Actors:	Publisher, Aggregator/Distributor
Prerequsites:	
Scenario:	1. Publishers send a message designed to "deliver" a complete magazine/newspaper issue made up of single-article PDFs with accompanying metadata to aggregators.

	2. Aggregators process the message and download sigle-article PDFs making up a complete issue from a server or CMS for distribution to subscribers.
-Requirements	1. The message must identify the publisher, publication and issue of content being delivered. 2. The single-article PDFs must be available on a server or CMS for download. 3. The message must be based on the PRISM Source Vocabulary 1.0.

3.1.3 Use Case 003: Deliver a Complete PSV Tagged Issue

Title	Deliver a Complete PSV Tagged Issue to Aggregators
Summary:	In the future, magazine and news publishers wish to deliver a PSV-based XML message containing publication/issue metadata with metadata for each article of the issue along with PSV-tagged article content.
Actors:	Publishers, Aggregator/Distributors
Prerequsites:	
Scenario:	1. Publishers send a message designed to "deliver" a complete PSV tagged magazine/newspaper issue to aggregators. 2. Aggregator/Distributors process the message and accept delivery of the PSV tagged issue for distribution to subscribers.
-Requirements	1. The message must identify the publisher, publication and issue of content being delivered. 2. The issue components must be available on a server or CMS for download. 3. The message must be based on the PRISM Source Vocabulary 1.0..

3.2 pamP High Level Model

The high-level model expressed by the pamP schema supports each of the use cases. See Figure 3.1.

The root element of the pamP model is the <message element. The <message carries an im= attribute where the Implementation Model number is to be referenced. You can learn more about Implementation Models in the next section.

The <message is made up of issue-level metadata followed by either <issuelinks to PDF's that support Use Case 001 and 002. Following this path, metadata about each issue article may also be transmitted. A separate path of this schema allows the publisher to support Use Case 003, delivering PSV-tagged issue content.
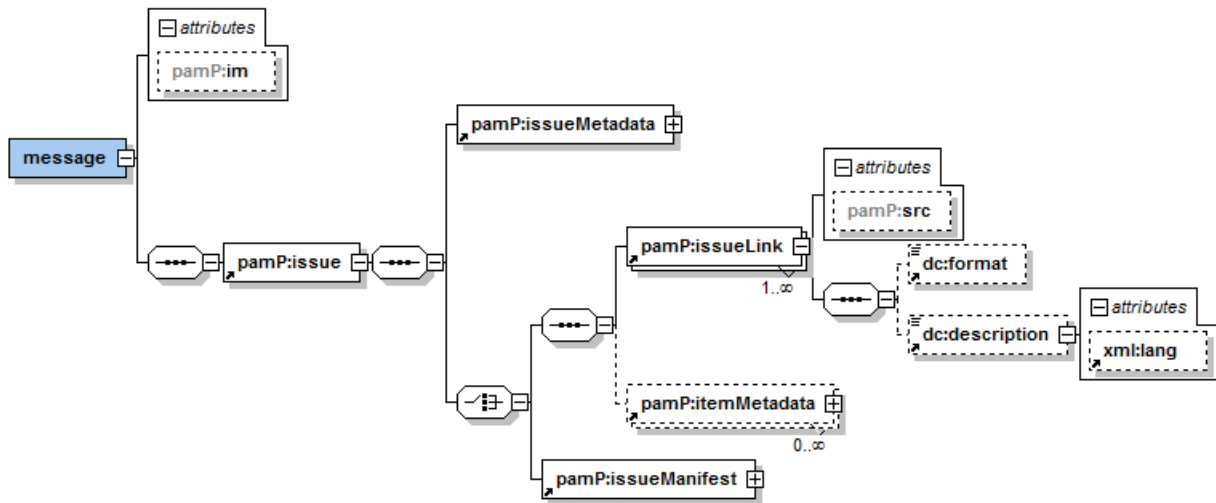
Figure 3.1 The message model for pamP.

4   IMPLEMENTATION MODELS

The use cases driving the development of the PSV Aggregator/Distributor Message Package may be implemented in a number of ways. Implementation Models for these use cases are documented in this section of the Guide.

4.1   Implementation Model 1-1

The first implementation model for Use Case #1 is a model where issue metadata is delivered to aggregators/distributors along with a link to a PDF of the issue, ready for distribution to subscribers. Figure 4.1 shows the XSD path to support this implementation model.

PAMP Implementation 1-1
Issue metadata + External link for issue data

Figure 4.1 IM 1-1 Schema View

```
<message im="1-1">
<issue><issueMetadata> . . . </issueMetadata>
<issueLink scr=http://www.***.pdf>
            <dc:format>application/pdf</dc:format>
</issueLink>
</issue>
</message>
```

4.2   Implementation Model 1-2

The second implementation model for Use Case #1 is a model where issue metadata is delivered to aggregators/distributors along with a link to a PDF of the issue, ready for distribution to subscribers. In

13

addition metadata about each article within the issue is provided to the aggregator/distributors. Figure 4.2 shows the XSD path to support this implementation model.

Figure 4.2 IM 1-2 Schema View

```
<message im="1-2">
<issue><issueMetadata> . . . </issueMetadata>
<issueLink scr=http://www.***.pdf>
            <dc:format>application/pdf</dc:format>
</issueLink>
<itemMetadata> . . .  </itemMetadata>
</issue>
</message>
```

4.3 Implementation Model 1-3

The third implementation model for Use Case #1 is a model where issue metadata is delivered to aggregators/distributors along with multiple links to the issue in a number of formats, ready for distribution to subscribers. Figure 4.3 shows the XSD path to support this implementation model.

Figure 4.3 IM 1-3 Schema View

```
<message im="1-3">
<issue><issueMetadata> . . . </issueMetadata>
<issueLink scr=http://www.***.pdf>
            <dc:format>application/pdf</dc:format>
</issueLink>
<issueLink scr=http://www.***.zip>
            <dc:format>application/zip+EPUB3</dc:format>
</issueLink>

</issue>
</message>
```

4.4 Implementation Model 1-4

The fourth implementation model for Use Case #1 is a model where issue metadata is delivered to aggregators/distributors along with multiple links to the issue in a number of formats, ready for distribution to subscribers. In addition sample pages such as the cover and contents will be delivered to subscribers. Metadata about each article in the issue may also be delivered. Figure 4.4 shows the XSD path to support this implementation model.

PAMP Implementation 1– 4
Issue metadata＋External links for multiple issue data
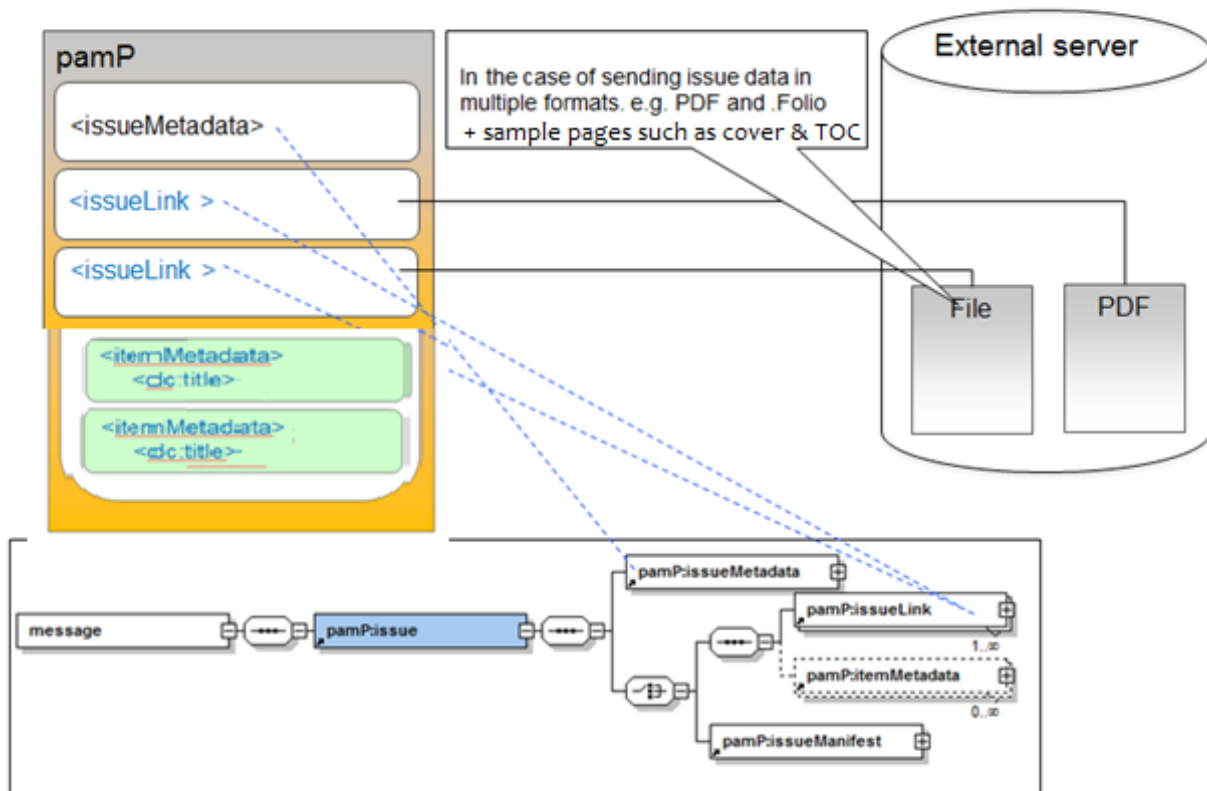

```
<message im="1-4">
<issue><issueMetadata> . . . </issueMetadata>
<issueLink scr=http://www.***.pdf>
      <dc:format>application/pdf</dc:format>
      <dc:description>full PDF issue</dc:description>
</issueLink>
<issueLink scr=http://www.***.zip>
      <dc:format>application/zip+EPUB3</dc:format>
      <dc:description>full issue EPUB 3 format</dc:description>
</issueLink>
<issueLink scr=http://www.***cover.pdf>
      <dc:format>application/pdf</dc:format>
      <dc:description>Issue Cover PDF</dc:description>
</issueLink>
<issueLink scr=http://www.***toc.pdf>
      <dc:format>application/pdf</dc:format>
      <dc:description>Issue Tableof Contents PDF</dc:description>
</issueLink>
```

```
<itemMetadata>
      <prism:contentType>article</prism:contentType>
      <dc:identifier>article01</dc:identifier>
      <dc:title>Inside The Game of Thrones</dc:title>
</itemMetadata>
<itemMetadata>
      <prism:contentType>article</prism:contentType>
      <dc:identifier>article02</dc:identifier>
      <dc:title>The Climate Battle </dc:title>
</itemMetadata>
<itemMetadata> . . . </itemMetadata>
</issue>
</message>
```

4.5 Implementation Model 2-1

The first implementation model for Use Case #2 is a model where issue metadata is delivered to aggregators/distributors along with a manifest made up with multiple links to the single-article PDFs that make up the issue, ready for distribution to subscribers. Figure 4.4 shows the XSD path to support this implementation model.

Figure 4.4 IM 2-1 Schema View

```
<message im="2-1">
<issue><issueMetadata> . . . </issueMetadata>
```

```
     <issueManifest>
        <item sequence="1">
                <itemMetadata> . . .  </itemMetadata>
                <itemLink scr=http://www.***.pdf>
                    <dc:format>application/pdf</dc:format>
                </itemLink>
         </item>
        <item sequence="2">
                <itemMetadata> . . .  </itemMetadata>
                <itemLink scr=http://www.***.pdf>
                    <dc:format>application/pdf</dc:format>
                </itemLink>
         </item>
     </issueManifest>
</issue>
</message>
```
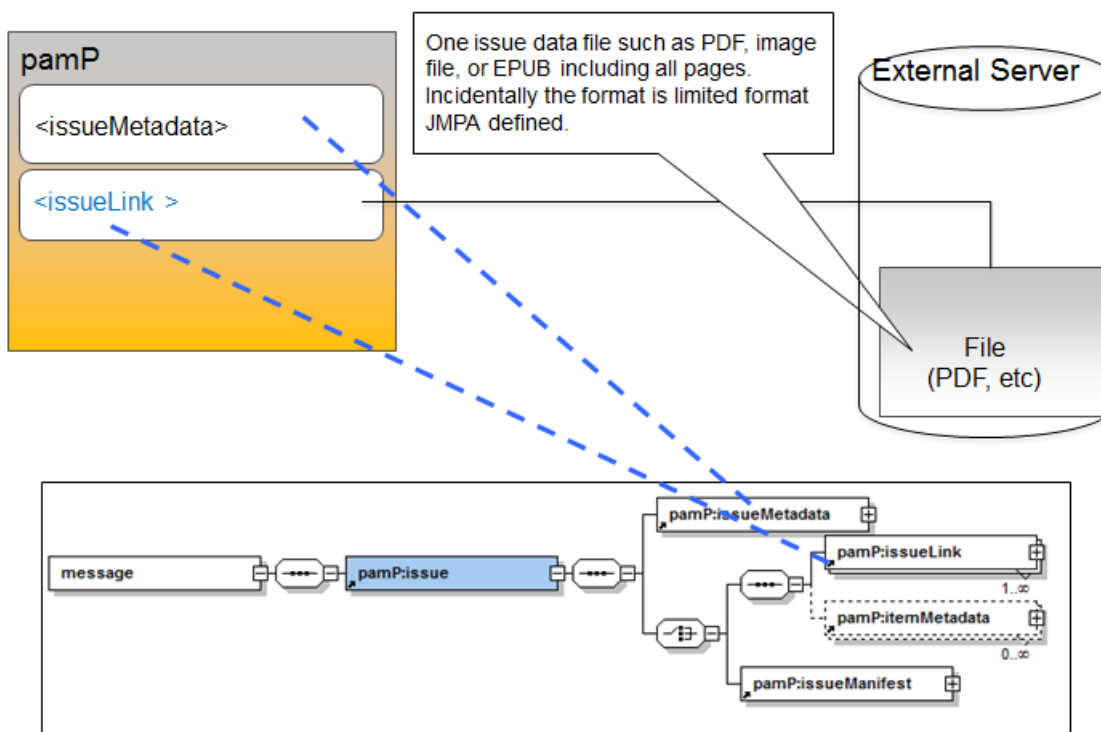
4.6 Implementation Model 2-2

The second implementation model for Use Case #2 is a model where issue metadata is delivered to aggregators/distributors along with a manifest made up with multiple links to the single-article PDFs that make up the issue and links to PSV/HTML5 for each article. Figure 4.5 shows the XSD path to support this implementation model.

Figure 4.5 IM 2-2 Schema View

18

```
<message im="">-1
<issue><issueMetadata> . . . </issueMetadata>
   <issueManifest>
      <item sequence="1">
             <itemMetadata> . . .  </itemMetadata>
             <itemLink scr=http://www.***.pdf>
                 <dc:format>application/pdf</dc:format>
             </itemLink>
             <itemLink scr=http://www.***.htm >
                 <dc:format>text/html</dc:format>
             </itemLink>

       </item>
      <item sequence="2">
             <itemMetadata> . . .  </itemMetadata>
             <itemLink scr=http://www.***.pdf>
                 <dc:format>application/pdf</dc:format>
             </itemLink>
             <itemLink scr=http://www.***.htm >
                 <dc:format>text/html</dc:format>
             </itemLink>
        </item>
   </issueManifest>
</issue>
</message>
```
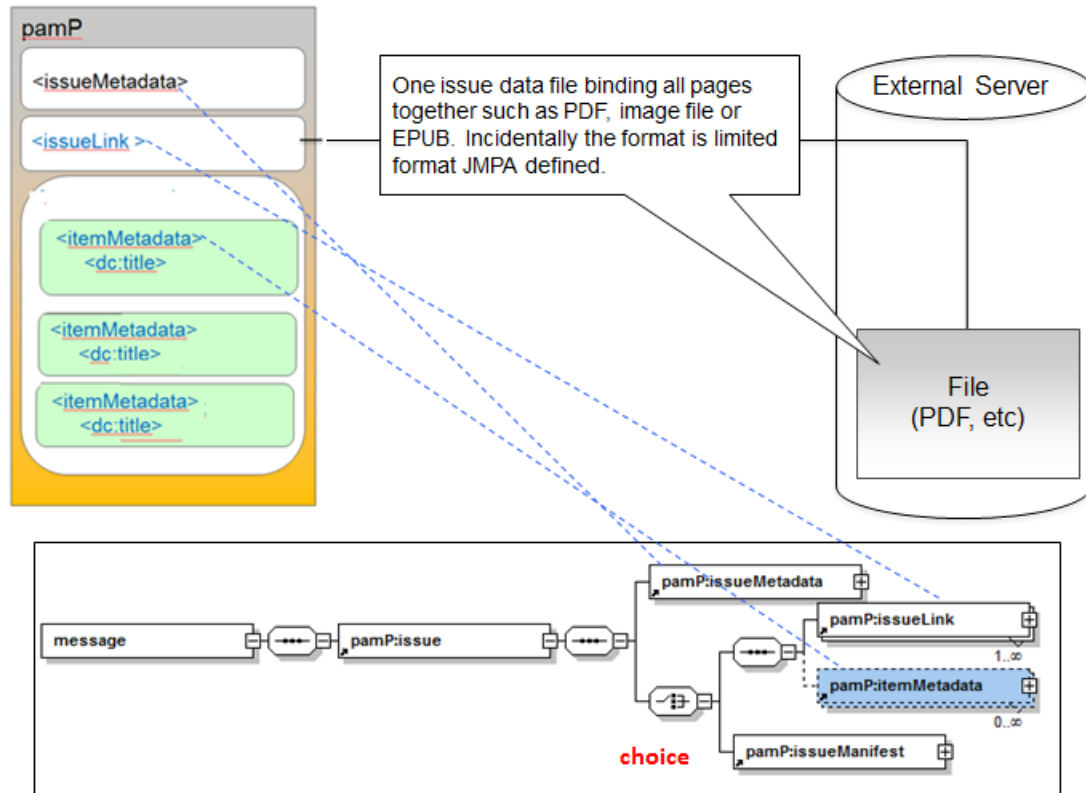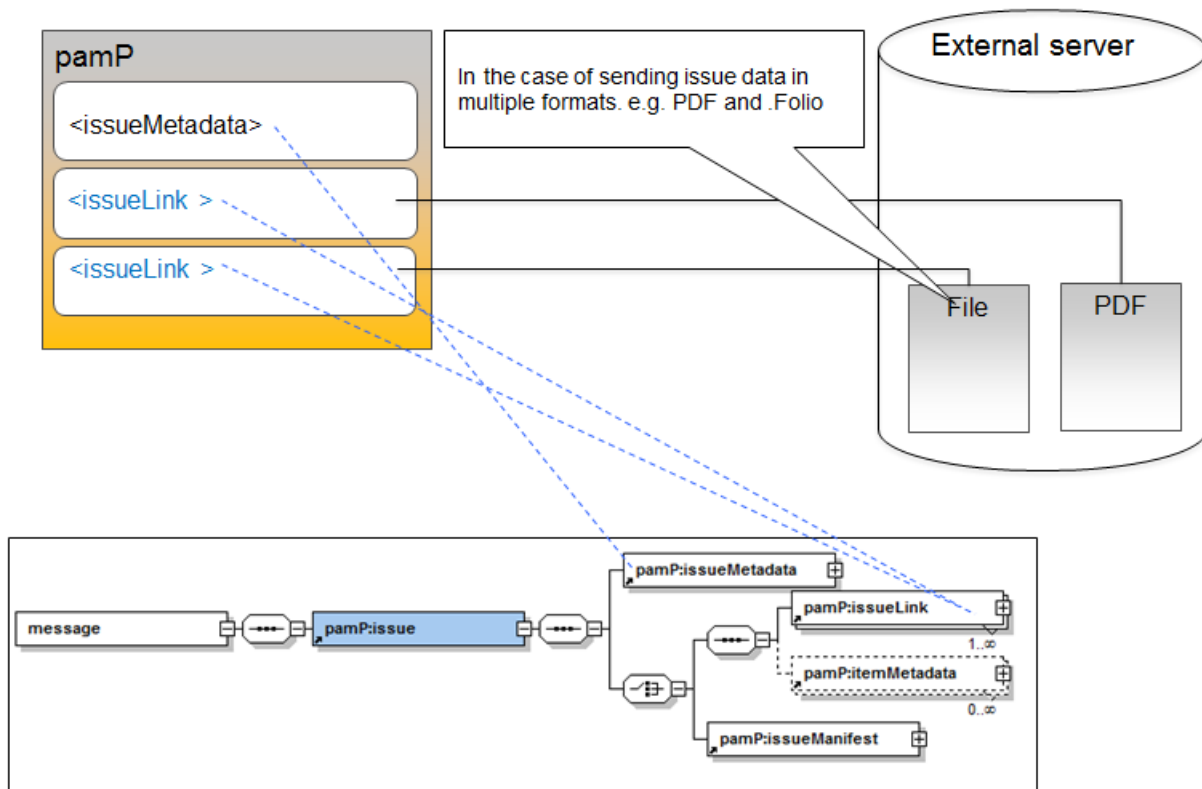
4.7 Implementation Model 3-1

The first implementation model for Use Case #3 is a model where each articl eis represented by an XML file with its content encoded in PSV HTML5 for distribution to subscribers. Figure 4.6 shows the XSD path to support this implementation model.

PAMP Implementation3
Issue Metadata + Article metadata + Content of each article
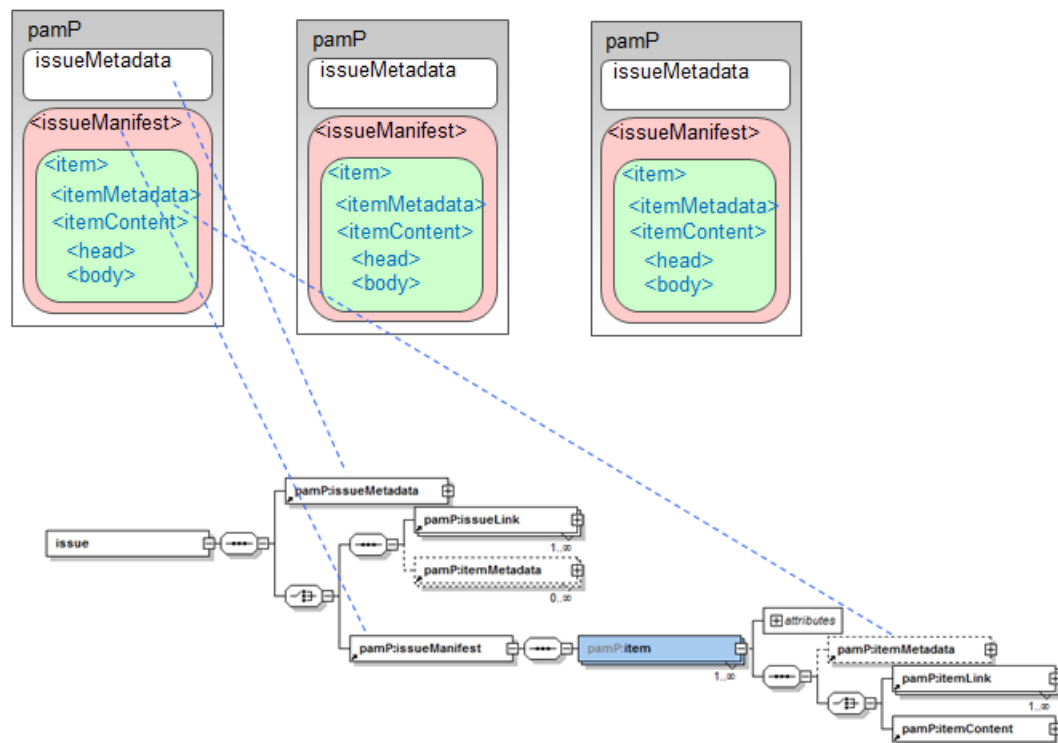
Figure 4.6 IM 3-1 Schema View

```
<message im=""">-1
<issue><issueMetadata> . . . </issueMetadata>
    <issueManifest>
        <item sequence="1">
                <itemMetadata> . . .  </itemMetadata>
                <itemContent>
                        <html5:head> . . . </html5:/head>
                        <html5:body> . . . </html5:/body>
                </itemContent>
         </item>
        <item sequence="2">
                <itemMetadata> . . .  </itemMetadata>
                <itemContent>
                        <html5:head> . . . </html5:/head>
                        <html5:body> . . . </html5:/body>
                </itemContent>
        </item>
    </issueManifest>
</issue>
</message>
```

5   ISSUE METADATA

One of the key structures in the PSV Aggregator/Distributor Message Package is issue metadata. This block was based on the PSV Issue Information block in the WhereUsed structure. Only issue metadata determined to be required by the three use cases documented in this guide were included. See Figure 5.1 for the complete <issueMetadata model.
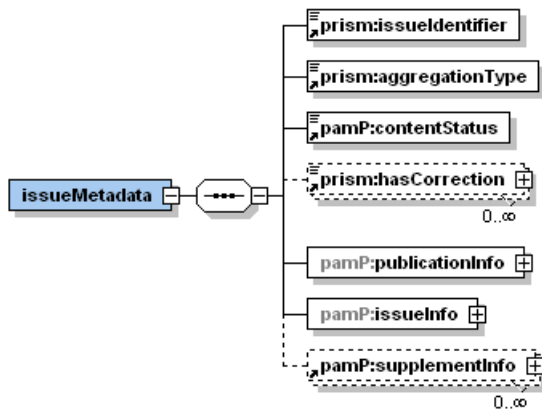
Figure 5.1 Issue Metadata Structure

5.1.1 Required Fields

Required fields in the PSV Aggregator/Distributor Message Package are a unique identifier for the issue, the prism:issueIdentifier, prism:aggregationType and the pamP:contentStatus.

5.1.2 pamP:contentStatus

Because corrected content may need to be sent to aggregator/distributors just as it may be sent to aggregator/syndicators in the existing PRISM Aggregator Message, tracking the status is important. JMPA has based its contentStatus element on the pam:status element. Values for the content status will be defined by each market employing this schema. However common fields will include "original" "update" and "correction".

> **Note:** to communicate the reason for a change in status, the best practice is to use the prism:hasCorrection field to indicate the purpose of the status change.

5.1.3 Publication Information

The PSV Aggregator/Distributor Message Package includes fields from the PSV Publication Information block. See Figure 5.2.
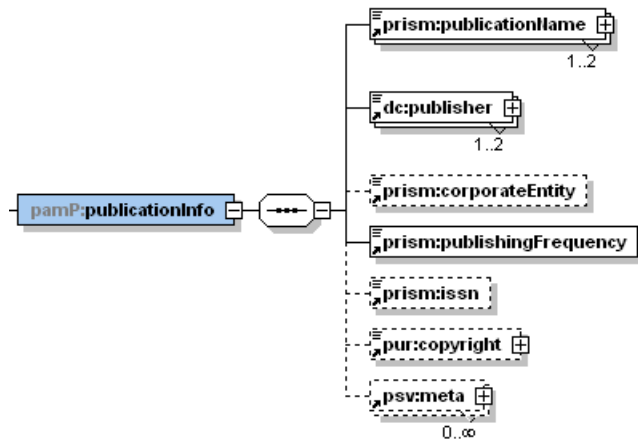
Figure 5.2 Publication Information included in pamP

5.1.4 Fields from PSV Issue Information

The majority of the fields in the PSV Aggregator/Distributor Message Package come from the PSV Issue Information block. See Figure 5.3.
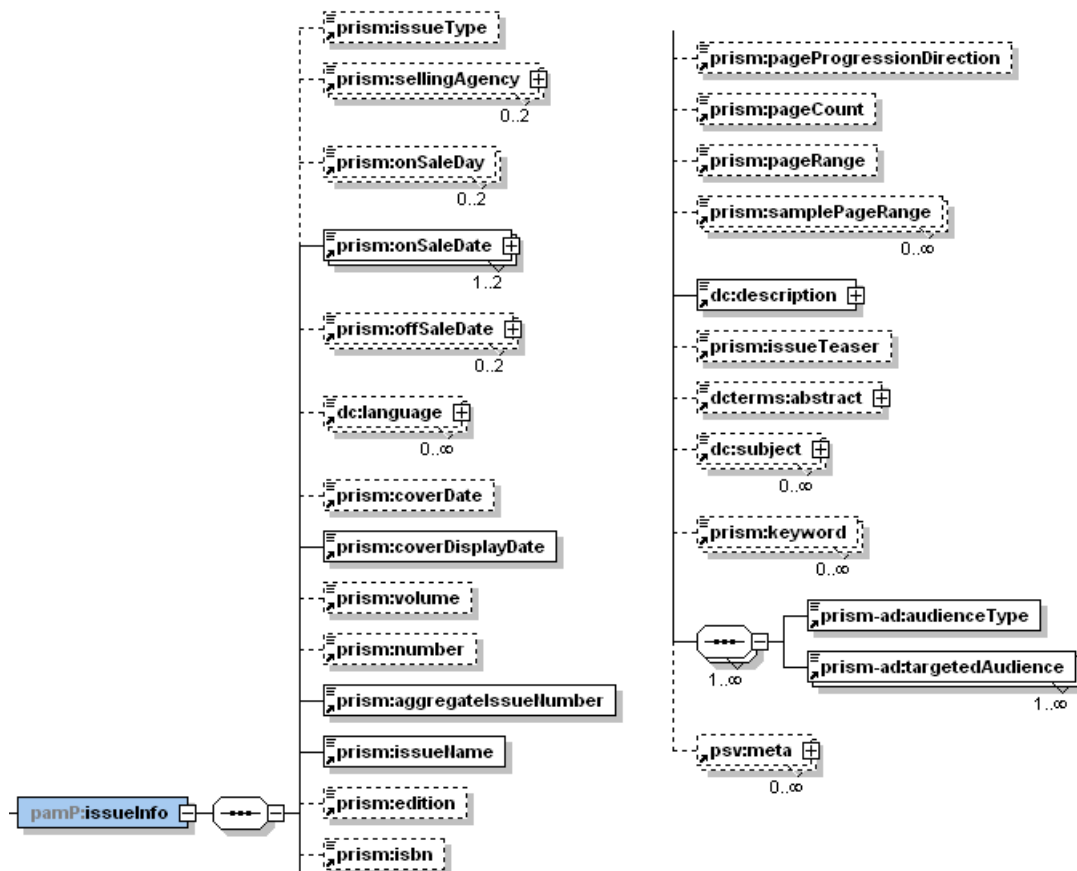
Figure 5.3 PSV Issue Information included in pamP

5.1.5 Supplement Information

PSV Aggregator/Distributor Message Package provides the capability to deliver magazine supplements with an issue or separately from an issue. Metadata specific to supplements can be included within the Supplement Info block. See Figure 5.4.
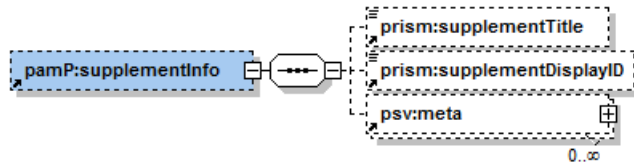
Figure 5.4 PSV Supplement Information included in pamP

6   ITEM METADATA

Implementation models for Use Case #2 add the inclusion of metadata about the components, or items, found within an issue. In this model metadata that could apply to either an article or a supplement are included at the item level. Metadata specific to an article or supplement are found in articleInfo or supplementInfo. See Figure6.1.
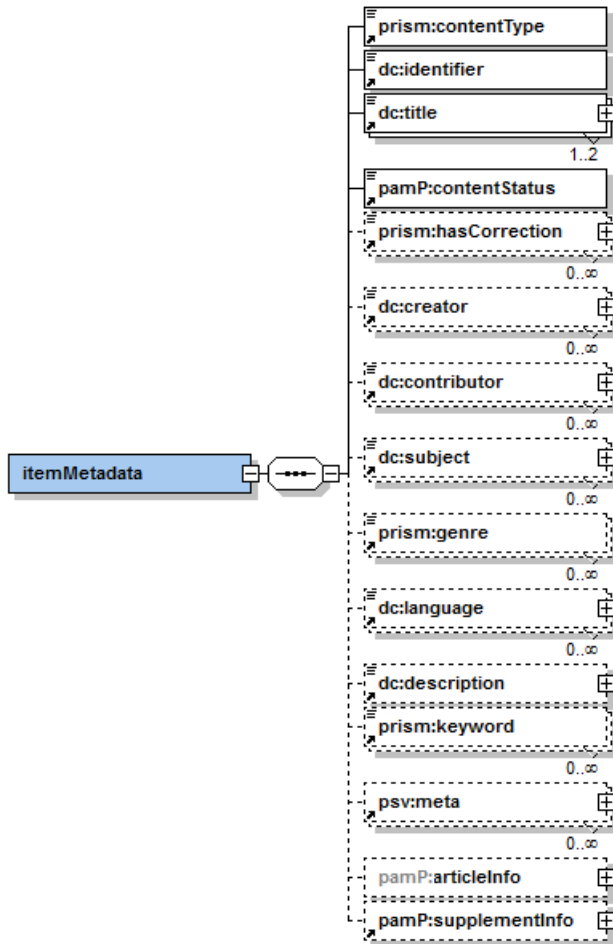
Figure 6.1 Item Metadata Structure

7 PAMP SCHEMA

The core of the PSV Aggregator/Distributor Message Package is found in the pamP.xsd.

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!-- Copyright IDEAlliance 2014  -->
<!--  August 2014 Version 1.2    -->
<!--  Added dc:description to issueLink to describe link target -->
<!-- Added local copy of xml.xsd -->


<!-- +++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
                                  pamP: XSD
     +++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++ -->
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
xmlns:dc="http://purl.org/dc/elements/1.1/" xmlns:dcterms="http://purl.org/dc/terms/"
xmlns:pamP="http://prismstandard.org/namespaces/pamP/1.2/"
xmlns:prism="http://prismstandard.org/namespaces/basic/3.0/" xmlns:prism-
ad="http://prismstandard.org/namespaces/prismad/3.0/"
xmlns:psv="http://prismstandard.org/namespaces/psv/1.0/"
xmlns:pur="http://prismstandard.org/namespaces/prismusagerights/3.0/"
xmlns:html5="http://www.w3.org/1999/html5"
targetNamespace="http://prismstandard.org/namespaces/pamP/1.2/"
elementFormDefault="qualified" attributeFormDefault="qualified" version="08182014">
       <xs:import namespace="http://purl.org/dc/elements/1.1/" schemaLocation="pamP-
dc.xsd"/>
       <xs:import namespace="http://purl.org/dc/terms/" schemaLocation="pamP-
dcterms.xsd"/>
       <xs:import namespace="http://prismstandard.org/namespaces/basic/3.0/"
schemaLocation="pamP-prism.xsd"/>
       <xs:import namespace="http://prismstandard.org/namespaces/prismad/3.0/"
schemaLocation="pamP-prism-ad.xsd"/>
       <xs:import namespace="http://prismstandard.org/namespaces/prismusagerights/3.0/"
schemaLocation="pamP-pur.xsd"/>
       <xs:import namespace="http://prismstandard.org/namespaces/psv/1.0/"
schemaLocation="pamP-psv.xsd"/>
       <xs:import namespace="http://www.w3.org/1999/html5" schemaLocation="pamP-
html5.xsd"/>
       <xs:import namespace="http://www.w3.org/XML/1998/namespace"
schemaLocation="xml.xsd"/>
       <!-- +++++++++ pamP:message Element +++++++++++++++++ -->
       <xs:element name="message">
             <xs:complexType>
                    <xs:sequence>
                           <xs:element ref="pamP:issue" minOccurs="1"/>
                    </xs:sequence>
                    <xs:attribute name="im" type="xs:string"/>
             </xs:complexType>
       </xs:element>
       <!-- +++++++++ pamP:issue Element +++++++++++++++++ -->
       <!-- dk added choice; first choice is IM-1 and second is IM-2 and IM-3 -->
       <xs:element name="issue">
             <xs:complexType>
                    <xs:sequence>
                           <xs:element ref="pamP:issueMetadata"/>
```

```
                              <xs:choice>
                                    <xs:sequence>
                                          <xs:element ref="pamP:issueLink"
maxOccurs="unbounded"/>
                                          <xs:element ref="pamP:itemMetadata"
minOccurs="0" maxOccurs="unbounded"/>
                                    </xs:sequence>
                                    <xs:element ref="pamP:issueManifest"/>
                              </xs:choice>
                        </xs:sequence>
                  </xs:complexType>
            </xs:element>
            <!-- ++++++++++ pamP:issueMetadata Element +++++++++++++++++ -->
            <xs:element name="issueMetadata">
                  <xs:complexType>
                        <xs:sequence>
                              <xs:element ref="prism:issueIdentifier"/>
                              <xs:element ref="prism:aggregationType"/>
                              <xs:element ref="pamP:contentStatus"/>
                              <xs:element ref="prism:hasCorrection" minOccurs="0"
maxOccurs="unbounded"/>
                              <xs:element name="publicationInfo">
                                    <xs:complexType>
                                          <xs:sequence>
                                                <xs:element ref="prism:publicationName"
minOccurs="1" maxOccurs="2"/>
                                                <xs:element ref="dc:publisher"
minOccurs="1" maxOccurs="2"/>
                                                <xs:element ref="prism:corporateEntity"
minOccurs="0"/>
                                                <xs:element
ref="prism:publishingFrequency"/>
                                                <xs:element ref="prism:issn"
minOccurs="0"/>
                                                <xs:element ref="pur:copyright"
minOccurs="0"/>
                                                <xs:element ref="psv:meta" minOccurs="0"
maxOccurs="unbounded"/>
                                          </xs:sequence>
                                    </xs:complexType>
                              </xs:element>
                              <xs:element name="issueInfo">
                                    <xs:complexType>
                                          <xs:sequence>
                                                <xs:element ref="prism:issueType"
minOccurs="0"/>
                                                <xs:element ref="prism:sellingAgency"
minOccurs="0" maxOccurs="2"/>
                                                <xs:element ref="prism:onSaleDay"
minOccurs="0" maxOccurs="2"/>
                                                <xs:element ref="prism:onSaleDate"
minOccurs="1" maxOccurs="2"/>
                                                <xs:element ref="prism:offSaleDate"
minOccurs="0" maxOccurs="2"/>
```

```
                                             <xs:element ref="dc:language"
minOccurs="0" maxOccurs="unbounded"/>
                                             <xs:element ref="prism:coverDate"
minOccurs="0"/>
                                             <xs:element
ref="prism:coverDisplayDate"/>
                                             <xs:element ref="prism:volume"
minOccurs="0"/>
                                             <xs:element ref="prism:number"
minOccurs="0"/>
                                             <xs:element
ref="prism:aggregateIssueNumber"/>
                                             <xs:element ref="prism:issueName"
minOccurs="1"/>
                                             <xs:element ref="prism:edition"
minOccurs="0"/>
                                             <xs:element ref="prism:isbn"
minOccurs="0"/>
                                             <xs:element
ref="prism:pageProgressionDirection" minOccurs="0"/>
                                             <xs:element ref="prism:pageCount"
minOccurs="0"/>
                                             <xs:element ref="prism:pageRange"
minOccurs="0"/>
                                             <xs:element ref="prism:samplePageRange"
minOccurs="0" maxOccurs="unbounded"/>
                                             <xs:element ref="dc:description"
minOccurs="1"/>
                                             <xs:element ref="prism:issueTeaser"
minOccurs="0"/>
                                             <xs:element ref="dcterms:abstract"
minOccurs="0"/>
                                             <xs:element ref="dc:subject"
minOccurs="0" maxOccurs="unbounded"/>
                                             <xs:element ref="prism:keyword"
minOccurs="0" maxOccurs="unbounded"/>
                                             <xs:sequence minOccurs="1"
maxOccurs="unbounded">
                                                 <xs:element ref="prism-
ad:audienceType" minOccurs="1"/>
                                                 <xs:element ref="prism-
ad:targetedAudience" minOccurs="1" maxOccurs="unbounded"/>
                                             </xs:sequence>
                                             <xs:element ref="psv:meta" minOccurs="0"
maxOccurs="unbounded"/>
                                         </xs:sequence>
                                     </xs:complexType>
                             </xs:element>
                             <xs:element ref="pamP:supplementInfo" minOccurs="0"
maxOccurs="unbounded"/>
                     </xs:sequence>
             </xs:complexType>
     </xs:element>
     <!-- +++++++++ pamP:supplementInfo Element +++++++++++++++++ -->
     <xs:element name="supplementInfo">
             <xs:complexType>
```

```
                    <xs:sequence>
                            <xs:element ref="prism:supplementTitle" minOccurs="0"/>
                            <xs:element ref="prism:supplementDisplayID" minOccurs="0"/>
                            <xs:element ref="psv:meta" minOccurs="0"
maxOccurs="unbounded"/>
                    </xs:sequence>
            </xs:complexType>
    </xs:element>
    <!-- ++++++++++ pamP:src Element ++++++++++++++++ -->
    <xs:element name="src" type="xs:anyURI"/>
    <!-- ++++++++++ pamP:issueLink Element ++++++++++++++++ -->
    <xs:element name="issueLink">
            <xs:complexType>
                    <xs:sequence>
                            <xs:element ref="dc:format" minOccurs="0"/>
                            <xs:element ref="dc:description" minOccurs="0"/>
                    </xs:sequence>
                    <xs:attribute name="src" type="xs:anyURI"/>
            </xs:complexType>
    </xs:element>
    <!-- ++++++++++ pamP:issueManifest Element ++++++++++++++++ -->
    <!-- dk added choice; first choice is IM-2 and second choice is IM-3 -->
    <xs:element name="issueManifest">
            <xs:complexType>
                    <xs:sequence>
                            <xs:element name="item" maxOccurs="unbounded">
                                    <xs:complexType>
                                            <xs:sequence>
                                                    <xs:element ref="pamP:itemMetadata"
minOccurs="0"/>
                                                    <xs:choice>
                                                            <xs:element ref="pamP:itemLink"
minOccurs="1" maxOccurs="unbounded"/>
                                                            <xs:element
ref="pamP:itemContent"/>
                                                    </xs:choice>
                                            </xs:sequence>
                                            <xs:attribute name="sequence"
type="xs:string"/>
                                    </xs:complexType>
                            </xs:element>
                    </xs:sequence>
            </xs:complexType>
    </xs:element>
    <!-- ++++++++++ pamP:itemMetadata Element ++++++++++++++++ -->
    <xs:element name="itemMetadata">
            <xs:complexType>
                    <xs:sequence>
                            <!-- put all item level metadata here    -->
                            <xs:element ref="prism:contentType"/>
                            <xs:element ref="dc:identifier"/>
                            <xs:element ref="dc:title" minOccurs="1" maxOccurs="2"/>
```

```
                              <xs:element ref="pamP:contentStatus"/>
                              <xs:element ref="prism:hasCorrection" minOccurs="0"
maxOccurs="unbounded"/>
                              <xs:element ref="dc:creator" minOccurs="0"
maxOccurs="unbounded"/>
                              <xs:element ref="dc:contributor" minOccurs="0"
maxOccurs="unbounded"/>
                              <xs:element ref="dc:subject" minOccurs="0"
maxOccurs="unbounded"/>
                              <xs:element ref="prism:genre" minOccurs="0"
maxOccurs="unbounded"/>
                              <xs:element ref="dc:language" minOccurs="0"
maxOccurs="unbounded"/>
                              <xs:element ref="dc:description" minOccurs="0"/>
                              <xs:element ref="prism:keyword" minOccurs="0"
maxOccurs="unbounded"/>
                              <xs:element ref="psv:meta" minOccurs="0"
maxOccurs="unbounded"/>
                              <xs:element name="articleInfo" minOccurs="0">
                                      <xs:complexType>
                                              <xs:sequence>
                                                      <xs:element ref="prism:section"
minOccurs="0" maxOccurs="1"/>
                                                      <xs:element ref="prism:seriesTitle"
minOccurs="0"/>
                                                      <xs:element ref="prism:seriesNumber"
minOccurs="0"/>
                                                      <xs:element
ref="prism:pageProgressionDirection" minOccurs="0"/>
                                                      <xs:element ref="prism:pageCount"
minOccurs="0"/>
                                                      <xs:element ref="prism:startingPage"/>
                                                      <xs:element ref="prism:endingPage"/>
                                                      <xs:element ref="prism:pageRange"
minOccurs="0"/>
                                                      <xs:element ref="prism:samplePageRange"
minOccurs="0" maxOccurs="unbounded"/>
                                                      <xs:element ref="psv:meta" minOccurs="0"
maxOccurs="unbounded"/>
                                              </xs:sequence>
                                      </xs:complexType>
                              </xs:element>
                              <xs:element ref="pamP:supplementInfo" minOccurs="0"/>
                      </xs:sequence>
              </xs:complexType>
      </xs:element>
      <!-- ++++++++++ pamP:itemLink Element ++++++++++++++++ -->
      <xs:element name="itemLink">
              <xs:complexType>
                      <xs:sequence>
                              <xs:element ref="dc:format" minOccurs="0"/>
                      </xs:sequence>
                      <xs:attribute name="src" type="xs:anyURI"/>
              </xs:complexType>
```

```
        </xs:element>
    <!-- ++++++++++ pamP:itemContent Element ++++++++++++++++ -->
    <xs:element name="itemContent">
            <xs:complexType>
                    <xs:sequence>
                            <!--xs:element ref="html5:head" minOccurs="0"/-->
                            <xs:element ref="html5:body" minOccurs="0"/>
                    </xs:sequence>
            </xs:complexType>
    </xs:element>
    <!-- ++++++++++ pamP:status Element ++++++++++++++++ -->
    <xs:element name="contentStatus" type="xs:string"/>
</xs:schema>
```