Activity Streams Working Group

M. Atkins
SAY Media
W. Norris
Google
C. Messina
Citizen Agency, Google
M. Wilkinson
MySpace, Facebook, VMware
R. Dolin
Microsoft
February 13, 2011

TOC

Atom Activity Streams 1.0

Abstract

This document presents an XML format that allows activities on social objects to be expressed within the Atom Syndication Format.

Table of Contents

1. Introduction

TOC

The Atom Syndication Format, as defined in **[RFC4287]**, is widely used to transmit various types of web content such as weblog posts, news headlines, as well as user activities within social sites. In the case of user activities, Atom lacks the ability to express much of the activity-specific metadata in a machine-parseable format. Activity Streams is an XML format designed to allow this additional activity metadata to be expressed within existing Atom entries and feeds.

It is a goal of this specification to provide sufficient metadata about an activity such that a consumer of the data can present it to a user in a rich human-friendly format. This may include constructing readable sentences about the activity that occurred, visual representations of the activity, or combining similar activities for display.

1.1. Namespace and Version

TOC

The XML Namespaces **[xml-names]** URI for the XML data format described in this specification is:

```
http://activitystrea.ms/spec/1.0/
```

For convenience, this data format may be referred to as "Atom Activities 1.0".

1.2. Notational Conventions

TOC

This specification uses the namespace prefix "activity:" for the Namespace URI identified in **Section 1.1**, and the namespace prefix "atom:" for the Namespace URI identified in Section 1.2 of **[RFC4287]**. Note that the choice of namespace prefixes is arbitrary and not semantically significant.

This specification uses a shorthand form of terms from the XML Infoset **[xml-infoset]**. The term "element" is used to refer to an element information item, and "attribute" is used to refer to an attribute information item.

This specification allows the use of IRIs **[RFC3987]**. Every URI **[RFC3986]** is also an IRI, so a URI may be used wherever an IRI is named. When an IRI that is not also a URI is given for dereferencing, it MUST be mapped to a URI using the steps in Section 3.1 of **[RFC3987]**. When an IRI is serving as an identifier, it MUST NOT be so mapped.

The text of this specification provides the sole definition of conformance. Examples in this specification are non-normative.

This specification uses "Atom" to refer to **[RFC4287]**.

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]**.

2. Activity Concepts

TOC

In its simplest form, an activity consists of an actor, a verb, and an object. It tells the story of a person performing an action on or with an object -- "Geraldine posted a photo" or "John shared a video". In most cases these elements will be explicitly declared, but they may also be implied.

2.1. The Activity Construct

TOC

An activity consists of the following logical components:

2.1.1. Time

TOC

A Date Construct, as defined in section 3.3 of **[RFC4287]**, that identifies the time at which the activity occurred. It is important to note that this is not necessarily the same as the time at which the activity was published. An Activity construct MUST have exactly one Time value.

2.1.2. Actor

An **Object Construct** that identifies the entity that performed the activity. An Activity construct MUST have exactly one actor.

2.1.3. Verb

An IRI reference that identifies the action of the activity. This value MUST be an absolute IRI, or a IRI relative to the base IRI of `http://activitystrea.ms/schema/1.0/`. An Activity construct MUST have exactly one verb.

2.1.4. Object

This **Object Construct** identifies the primary object of the activity. An Activity construct MUST have exactly one object.

2.1.5. Target

The target of an activity is an **Object Construct** that represents the object to which the activity was performed. The exact meaning of an activity's target is dependent on the verb of the activity, but will often be the object of the English preposition "to". For example, in the activity "John saved a movie to his wishlist", the target of the activity is "wishlist". The activity target MUST NOT be used to identify an indirect object that is not a target of the activity. An Activity construct MAY have a target but it MUST NOT have more than one.

2.1.6. Title

An HTML representation of the natural language title for this activity. Consumers MAY use the value of this field, if set, as a fallback for when the provided verb is not recognized. An Activity Construct MAY have a title but it MUST NOT have more than one.

2.1.7. Summary

An HTML representation of the activity, including visual elements such as thumbnails. Consumers MAY use the value of this field to present the activity to a user, in lieu of constructing a representation from each of the activity's components. Publishers SHOULD include a summary for activities in order to provide a fallback for parsers that are not Activity Streams aware. An Activity Construct MAY have a summary but it MUST NOT have more than one.

2.2. The Object Construct

An object construct is a thing, real or imaginary, which participates in an activity. It may be the entity performing the activity, or the entity on which the activity was performed. An object consists of the logical components defined in the following sections. Certain object types may further refine the meaning of these components, or they may define additional components. If an object type defines an additional component then it SHOULD also define the representation of that component in one or more serialization formats.

2.2.1. ID

The id of an object construct is an IRI that uniquely identifies the object. Note that the definition of "IRI" excludes relative references. An Object construct SHOULD have an ID value, and MUST NOT have more than one.

If an object construct does not have an ID value consumers MAY use the **Permalink URL** as a weaker identifier, but must in this case allow for the fact that Permalink URL is not defined to be unique across all objects and be prepared to handle duplicates.

2.2.2. Name

TOC

This string value provides a human readable display name for the object, if the object has a name. An Object construct MAY have a name, but MUST NOT have more than one.

2.2.3. Summary

TOC

This string value provides a human readable description or summary of the Object. An Object construct MAY have a summary, but MUST NOT have more than one.

2.2.4. Representative Image

TOC

This IRI reference identifies an image resource which provides a visual representation of the object, intended for human consumption. An Object construct MAY have a representative image, but MUST NOT have more than one.

2.2.5. Permalink URL

TOC

This IRI reference identifies a resource which provides an HTML representation of the object. An Object construct MAY have a Permalink URL, but MUST NOT have more than one.

2.2.6. Object Type

TOC

An IRI reference that identifies the type of object. This value MUST be an absolute IRI, or a IRI relative to the base IRI of `http://activitystrea.ms/schema/1.0/`. An Object construct MAY have a type, but MUST NOT have more than one.

If no object type is present, the object has no specific type. Consumers SHOULD refer to such objects only by their names. For example, when forming an activity sentence a consumer might say "Johan posted 'My Cat'" rather than "Johan posted a photo: 'My Cat'".

---

3. Atom Representation

TOC

Activities can be represented in an Atom document using a combination of conventions and custom extension elements. This specification defines two ways that an activity can be represented using `atom:entry` elements, as a Full Activity Entry or using Implied Activity Shorthand. Other specifications may define additional representations.

3.1. Activity Representations

TOC

Any valid Atom entry as defined by section 4.1.2 of **[RFC4287]** is a representation of an activity as defined in **Section 2.1**.

If the `atom:entry` contains an `activity:object` element then it is a Full Activity Entry and MUST be interpreted as described in **Section 3.1.1**. Otherwise the entry is an Implied Activity Shorthand Entry and MUST be interpreted as described in **Section 3.1.2**.

3.1.1. Full Activity Entry

TOC

A Full Activity Entry is one in which the `atom:entry` represents the full **Activity Construct**. The components of the activity are represented by child elements of the `atom:entry`. In this section, "the entry" refers to the `atom:entry` element that represents the Full Activity Entry.

The components of the Activity Construct are represented in a Full Activity Entry as follows:

Actor
: The **Actor** of the Activity Construct is represented by the `atom:author` element that applies to the entry, as defined by section 4.2.1 of **[RFC4287]**. The representation of an Object Construct as an `atom:author` element is defined in **Section 3.2.3**.

Object
: The **Object** of the Activity Construct is represented by the `activity:object` child element of the entry, as defined in **Section 5.2**.

Target
: The **Target** of the Activity Construct is represented by the `activity:target` child element of the entry, as defined in **Section 5.4**. If no such element is present, the Activity Construct has no Target.

Verb
: The **verb IRI** of the Activity Construct is represented by an `activity:verb` child element as defined in **Section 5.1**. If the entry has no `activity:verb` child element, then the Activity Construct has the "Post" verb as defined in **Section 6**.

Time
: The **Time** of the Activity Construct is represented by the `atom:published` child element of the entry, as defined in section 4.2.9 of **[RFC4287]**.

Title
: The **Title** of the Activity Construct is represented by the `atom:title` child element of the entry, as defined in section 4.2.14 of **[RFC4287]**. Consumers MUST convert the value into HTML if necessary.

Summary
: The **Summary** of the Activity Construct is represented either by the `atom:content` or `atom:summary` child element of the entry, as defined in section 4.1.3 and section 4.2.13 of **[RFC4287]** respectively. If `atom:summary` is present it MUST contain the summary of the Activity Construct. If `atom:summary` is not present then the `atom:content` element MUST contain the summary of the Activity Construct. Consumers MUST convert the value into HTML if necessary.

TOC

3.1.2. Implied Activity Shorthand

An implied activity shorthand entry represents an Activity Construct in terms of its object. In this mode, the `atom:entry` primarily represents the Object Construct that is the object of the Activity Construct, but also represents certain additional components of the Activity Construct. In this section, "the entry" refers to the `atom:entry` element.

The components of the Activity Construct are represented in an Implied Activity Shorthand Entry as follows:

Actor
: The **Actor** of the Activity Construct is represented by the `atom:author` element that applies to the entry, as defined by section 4.2.1 of **[RFC4287]**. The representation of an Object Construct as an `atom:author element` is defined in **Section 3.2.3**.

Object
: The **Object** of the Activity Construct is represented by the entry itself. The representation of an Object Construct as an `atom:entry` element uses the child elements defined in **Section 3.2.2**.

Target
: The Activity Construct does not have a **Target**.

Verb
: The **verb IRI** of the Activity Construct is represented by an `activity:verb` child element as defined in **Section 5.1**. If the entry has no `activity:verb` child element, then the Activity Construct has the "Post" verb as defined in **Section 6**.

Time
: The **Time** of the Activity Construct is represented by the `atom:published` child element of the entry, as defined in section 4.2.9 of **[RFC4287]**.

Title
: The Activity Construct does not have a **Title**.

Summary
: The Activity Construct does not have a **Summary**.

The Implied Activity Shorthand is an accommodation to allow activity metadata to be added to a feed that is considered by traditional Atom consumers to be a list of content objects rather than a list of activities, without changing the meaning of that feed. Such feeds can often be recognized by the fact that the `atom:title` element of each entry is the title of an object rather than a sentence describing an activity.

3.2. Object Representations

TOC

Object constructs are used to identify a number of different components in an activity, including the **Actor** and **Object** of an activity, among others. Depending on what an object is used to identify, it is rendered in Atom using a different root element, as defined in **Section 3.2.1**. The components of the object are represented using child elements as defined in **Section 3.2.2**.

3.2.1. Object Construct Representations

TOC

Object constructs can be represented using the following elements:

atom:entry
: When used to describe the object of an Implied Activity as defined in **Section 3.1.2**, an Object Construct is represented using the `atom:entry` element, using the elements defined in **Section 3.2.2**.

atom:author
: When used to describe the **Actor of an Activity Construct**, an Object Construct is represented using the `atom:author` element, as defined in **Section 3.2.3**.

activity:object
: When used to describe the **Object of an Activity Construct**, an Object Construct is represented using the `activity:object` element as defined in **Section 5.2**.

activity:target
: When used to describe the **Target of an Activity Construct**, an Object Construct is represented using the `activity:target` element as defined in **Section 5.4**.

other elements
: Other specifications or object types may define additional elements that can be used to represent objects.

The element used to represent the Object MAY have additional attributes and child elements that represent properties defined by the Object Type of the Object or by other extensions. These extensions are defined by the Atom representation provided for the Object Type or extension.

3.2.2. Common Representation for Object Construct Components

TOC

Unless specified otherwise, the components of an Object Construct are represented using child elements of the representative element as follows, regardless of the type of element used to describe the Object Construct:

ID
: The **ID** of the Object Construct is represented using the `atom:id` element, as defined in section 4.2.6 of **[RFC4287]**.

Name
: The **name** of the Object Construct is represented using the `atom:title` element, as defined in section 4.2.14 of **[RFC4287]**. Consumers MUST convert the value into HTML if necessary.

Summary
: The **summary** of the Object Construct is represented using the `atom:summary` element, as defined in section 4.2.13 of **[RFC4287]**. Consumers MUST convert the value into HTML if necessary.

Representative Image
: The **representative image** of the Object Construct is represented by the first `atom:link` element whose rel attribute value is "preview" and a whose type attribute value is an image media type.

Permalink URL
: The **permalink URL** of the Object Construct is represented by the href attribute of an `atom:link` element, whose rel attribute value is "alternate" and whose type attribute value is "text/html".

Object Type
: The **object type URL** of the Object Construct is represented by an `activity:object-type` element as defined in **Section 5.3**.

3.2.3. Object represented as atom:author

TOC

When used to describe the **Actor of an Activity**, an Object Construct is represented using the `atom:author` element (see section 4.2.1 of **[RFC4287]**). The components of the Object Construct are represented using child elements as defined in **Section 3.2.2**, with the following exceptions:

Name
: The name of the Object Construct is represented using the `atom:name` element, as defined in section 3.2.1 of **[RFC4287]**.

Permalink URL
: The URL of the Object Construct is represented by the href attribute of an `atom:link` element, whose rel attribute value is "alternate" and a whose type attribute value is "text/html". If no such `atom:link` element is present, consumers SHOULD accept the content of the `atom:uri` element as representing the Object Permalink URL.

4. RSS 2.0 Representation

TOC

Activities can be represented in an RSS 2.0 document, as defined by **[RSS2]**, using a combination of conventions and custom extension elements. An activity is represented using an RSS "item" element. The RSS serialization only supports a subset of the activity data model allowing descriptions of simple activities that have only a verb and an object. This serialization is optimized for posting activities describing the creation of new objects. Publishers with a need for representing other fields that are not catered for by this serialization are recommended to use the Atom serialization instead for best results. This RSS serialization is defined only as a means for those already publishing streams of objects as RSS to produce minimally useful lists of activities for activity-aware consumers.

A specific RSS document has a single activity actor that applies to all activities it represents. This actor has no known id, name, url or object type but it is assumed that all activities in the feed share the same actor and therefore consumers MAY use information obtained from another source, such as the user account which is assumed to be the subject of the feed, to populate these properties.

4.1. Activity Construct Represented as RSS "item"

TOC

In this section, "the item" refers to the RSS "item" element which is representing the activity.

An RSS "item" element represents exactly one Activity Construct in terms of its object. The item actually directly represents both the Object Construct that is the object of the Activity Construct and the Activity Construct itself via different subsets of the child elements.

The components of the Activity Construct are represented in an RSS item as follows:

Actor
: The **Actor** of the Activity Construct is the feed-wide actor as described in **Section 4**.

Object
: The **Object** of the Activity Construct is represented by the item itself. The representation of an Object Construct as an RSS item element is defined in **Section 4.2**.

Target
: The Activity Construct does not have a **Target**.

Verb
: The **verb IRI** of the Activity Construct is represented by an `activity:verb` child element as defined in **Section 5.1**. If the item has no `activity:verb` child elements then the Activity Construct has only the "Post" verb as defined in **Section 6**.

Time
: The **Time** of the Activity Construct is represented by the `pubDate` child element of the item, as defined in **[RSS2]**.

Title
: The Activity Construct does not have a **Title**.

Summary
: The Activity Construct does not have a **Summary**.

4.2. Object Construct Represented as RSS "item"

TOC

The components of an Object Construct are represented using child elements of an RSS `item` element as follows:

ID
: The **ID** of the Object Construct is represented using the RSS `guid` element, as defined in **[RSS2]**, if and only if its value conforms to the syntax of an IRI as defined in **[RFC3987]**. If the value is not a valid IRI then the Object Construct has no ID.

Name
: The **name** of the Object Construct is represented using the RSS `title` element, as defined in **[RSS2]**.

Summary

The Object Construct has no **summary**.

Representative Image
: The **representative image** of the Object Construct is represented by the href attribute of an `atom:link` element, whose rel attribute value is "preview" and a whose type attribute value is an appropriate image media type.

Permalink URL
: The **permalink URL** of the Object Construct is represented by the first RSS `link` child element, as defined in **[RSS2]**. If no such element is present, the value of the first `guid` child element that is marked as a permalink as defined in **[RSS2]** represents the Permalink URL. If neither element is present, the Object Construct has no Permalink URL.

Object Type
: Each **object type URL** of the Object Construct is represented by an `activity:object-type` element as defined in **Section 5.3**.

---

5. Elements used in XML Representations

TOC

This specification defines several elements in the XML namespace defined in **Section 1.1**. These elements are used in both the Atom and RSS serializations and may be used by other XML-based serializations of the activity streams data model defined by other specifications.

5.1. The activity:verb Element

TOC

The `activity:verb` element represents the Verb of an Activity Construct as defined in **Section 2.1.3**. Its content is an IRI reference that identifies the verb.

5.1.1. Comparing activity:verb

TOC

The value of `activity:verb` elements can be used to determine whether an Activity construct contains a recognized verb. Processors MUST compare `activity:verb` elements on a character-by-character basis (in a case-sensitive fashion). Comparison operations MUST be based solely on the IRI character strings and MUST NOT rely on dereferencing the IRIs or URIs mapped from them. As a result, two IRIs that resolve to the same resource but are not character-for-character identical will be considered different for the purposes of identifier comparison.

Processors SHOULD resolve relative IRIs to their absolute form using the base `http://activitystrea.ms/schema/1.0/` prior to comparison. For example, processors SHOULD consider the following IRIs as equivalent for the purpose of comparison:

- `http://activitystrea.ms/schema/1.0/post`
- `post`

5.2. The activity:object Element

TOC

An `activity:object` element represents the Object Construct that is the Object of an Activity Construct as defined in **Section 2.1.4**.

The content model of an `activity:object` element includes the elements indicated in **Section 3.2.2**, with the meanings defined in that section, as well as any extension elements defined by the Atom serializations of the object type listed for the object.

5.3. The activity:object-type Element

TOC

An `activity:object-type` element represents the Object Type of an Object Construct as defined in **Section 2.2.6**. Its content is an IRI reference that identifies the object type.

5.3.1. Comparing activity:object-type

TOC

The value of `activity:object-type` elements can be used to determine whether an Object construct is of a recognized type. Processors MUST compare `activity:object-type` elements using the same comparison rules described in **Section 5.1.1**.

5.4. The activity:target Element

TOC

An `activity:target` element represents the Object Construct that is the Target of an Activity Construct as defined in **Section 2.1.5**.

The content model of an `activity:target` element includes the elements indicated in **Section 3.2.2**, with the meanings defined in that section, as well as any extension elements defined by the Atom serializations of the object type listed for the object.

6. The "Post" Verb

TOC

This specification defines one initial verb and defers to other specifications to define the full schema of verbs and object types.

The "Post" verb describes the act of authoring an object and then publishing it online. The IRI reference for the "Post" verb is:

```
post
```

The actor of an Activity Construct using the "Post" verb is an Object Construct representing the person or object that authored and posted the item represented by the object of the Activity Construct.

The target of the Activity Construct, if present, represents the item into which the object is posted. For example, this could represent a blog that the author has posted in.

7. Security Considerations

TOC

As this specification defines an extension to the Atom Syndication Format, it is subject to the same security considerations defined in **[RFC4287]**.

Publishers or Consumers implementing Activity Streams as a stream of public data may also want to consider the potential for unsolicited commercial or malicious content and should take preventative measures to recognize such content and either identify it or not include it in their stream implementations.

Publishers should take reasonable measures to make sure potentially malicious user input such as cross-site scripting attacks are not included in the Activity Streams data they publish.

Consumers that re-emit ingested content to end-users MUST take reasonable measures if emitting ingested content to make sure potentially malicious ingested input is not re-emitted.

Consumers that re-emit ingested content for crawling by search engines should take reasonable measures to limit any use of their site as a Search Engine Optimization loophole. This may include converting un-trusted hyperlinks to text or including a rel="nofollow" attribute.

8. IANA Considerations

TOC

None.

9. License

TOC

10. Normative References

TOC

[RFC2119]	Bradner, S., “**Key words for use in RFCs to Indicate Requirement Levels**,” March 1997.
[RFC3986]	Berners-Lee, T., Fielding, R., and L. Masinter, “**Uniform Resource Identifiers (URI): Generic Syntax**,” January 2005.
[RFC3987]	Duerst, M. and M. Suignard, “**Internationalized Resource Identifiers (IRIs)**,” January 2005.
[RFC4287]	Nottingham, M. and R. Sayre, “**The Atom Syndication Format**,” December 2005.
[RSS2]	“**RSS 2.0 Specification**,” March 2009.
[xml-infoset]	Cowan, J., “**XML Information Set (Second Edition)**,” February 2004.
[xml-names]	Bray, T., “**Namespaces in XML 1.0 (Third Edition)**,” December 2009.

Appendix A. Acknowledgements

TOC

The authors wish to thank the Activity Streams community and implementers for their support, encouragement, and enthusiasm including but not limited to: Abdul Qabiz, Adina Levin, Adrian Chan, Adriana Javier, Alexandre Loureiro Solleiro, Ari Steinberg, Arne Roomann-Kurrik, Beau Lebens, Ben Hedrington, Ben Metcalfe, Ben Werdmuller, Bill de hÃ“ra, Bob Aman, Bob Wyman, Brynn Evans, Charlie Cauthen, Chris Chabot, Christian Crumlish, Dan Brickley, Danny Ayers, Dare Obasanjo, Darren Bounds, David Cramer, David Recordon, DeWitt Clinton, Douglas Pearce, Ed Summers, Elias Bizannes, Elisabeth Norris, Eric Woods, Evan Prodromou, Gee-Hsien Chuang, Greg Biggers, Hillary Madsen, Howard Liptzin, Ian Kennedy, Ivan Pulleyn, Jacob Kim, James Falkner, James Walker, Jason Kahn, Jeff Kunins, Jian Lin, John Panzer, Jon Paul Davies, Jonathan Coffman, Jonathan Dugan, Joseph Boyle, Joseph Hosten, Joseph Smarr, Jud Valeski, Julien Genestoux, Jyri EngestrÃ¶m, Kaliya Hamlin, Kevin Marks, Laurent Eschenauer, Manu Mukerji, Mark Weitzel, Marshall Kirkpatrick, Mary Hoder, Matt Leventi, Matt Wilkinson, Matthias Muller-Prove, Max Engel, Melvin Carvalho, Michael Richardson, Nate Benes, NeilFred Picciotto, Nick Lothian, Nissan Dookeran, Nitya Narasimhan, Pablo Martin, Pat G. Cappalaere, Peter Ferne, Peter Reiser, Peter Saint-Andre, Phil Wolff, Philip (flip) Kromer, Rick Severson, Robert Langbert, Todd Barnard, Ryan Boyd, Sam Sethi, Scott Raymond, Scott Seely, Simon Grant, Simon Wistow, Stephen Garcia, Steve Ivy, Steven Livingstone-Perez, Sylvain Carle, Sylvain Hellegouarch, Tantek Celik, Tatu Saloranta, Tim Moore, Timothy Young, Todd Barnard, Tosh Meston, and Tyler Gillies.

Appendix B. Examples

TOC

This appendix is non-normative.

B.1. Full Activity Entry Examples

TOC

A typical activity entry:

```
<entry xmlns="http://www.w3.org/2005/Atom" xmlns:activity="http://activitystrea.ms/spec/1.0/">
  <id>tag:photopanic.example.com,2009:activity/4859/4352</id>
  <title>Geraldine posted a Photo on PhotoPanic</title>
  <published>2009-11-02T15:29:00Z</published>
  <link rel="alternate" type="text/html" href="http://example.com/geraldine/activities/4352" />
  <activity:verb>post</activity:verb>
  <activity:object>
    <id>tag:photopanic.example.com,2009:photo/4352</id>
    <title>My Cat</title>
    <published>2009-11-02T15:29:00Z</published>
    <link rel="alternate" type="text/html" href="http://example.com/geraldine/photos/4352" />
    <activity:object-type>photo</activity:object-type>
  </activity:object>
```

```
    <content type="html">
      &lt;p&gt;Geraldine posted a Photo on PhotoPanic&lt;/p&gt;
      &lt;img src="/geraldine/photos/4352.jpg"&gt;
    </content>
  </entry>
```

This example shows a typical activity entry using the post verb defined by this specification and a hypothetical "photo" object type. It also demonstrates the use of `atom:title` and `atom:content` to provide fallback content for feed processors that do not support activity extensions. An activity stream application might render this entry into the sentence "Geraldine posted a photo titled 'My Cat'", with the photo title presented as a link.

Figure 1

A slightly more complex activity entry:

```
  <entry xmlns="http://www.w3.org/2005/Atom" xmlns:activity="http://activitystrea.ms/spec/1.0/">
    <id>tag:photopanic.example.com,2009:/activity/4859/1643</id>
    <title>Geraldine posted a photo to the My Pets album.</title>
    <published>2010-06-21T00:28:35Z</published>
    <link rel="alternate" type="text/html" href="http://example.com/geraldine/activities/1643" />
    <author>
      <name>Geraldine</name>
      <uri>http://example.com/geraldine</uri>
      <id>tag:photopanic.example.com,2009:person/4859</id>
      <activity:object-type>person</activity:object-type>
      <link rel="alternate" type="text/html" href="http://example.com/geraldine" />
    </author>
    <activity:object>
      <id>tag:photopanic.example.com,2009:photo/1643</id>
      <title>My Cat</title>
      <link rel="alternate" type="text/html" href="/geraldine/photos/1643" />
      <link rel="preview" type="image/jpeg" href="/geraldine/photos/1643/thumb.jpg" />
      <link rel="enclosure" type="image/jpeg" href="/geraldine/photos/1643/full.jpg" />
      <activity:object-type>photo</activity:object-type>
    </activity:object>
    <activity:target>
      <id>tag:photopanic.example.com,2009:photo-album/2519</id>
      <title>My Pets</title>
      <link rel="alternate" type="text/html" href="/geraldine/albums/pets" />
      <activity:object-type>photo-album</activity:object-type>
    </activity:target>
    <content type="xhtml">...</content>
  </entry>
```

This example demonstrates the use of the actor, verb, object and target elements.

Figure 2

B.2. Implied Activity Entry Example

TOC

A simple implied activity entry:

```
  <entry xmlns="http://www.w3.org/2005/Atom" xmlns:activity="http://activitystrea.ms/spec/1.0/">
    <id>tag:photopanic.example.com,2009:photo/4352</id>
    <title>My Cat</title>
    <published>2010-11-02T15:29:00Z</published>
    <link rel="alternate" type="text/html" href="http://example.com/geraldine/photos/4352" />
    <activity:object-type>photo</activity:object-type>
  </entry>
```

This example shows the same activity from **Appendix B.1**, but represented using the implied activity shorthand. Note that some of the components of the Activity Construct, such as title and summary, are not supported by this representation.

Authors' Addresses

Martin Atkins
SAY Media

Will Norris
Google

Chris Messina
Citizen Agency, Google

Monica Wilkinson
MySpace, Facebook, VMware

Rob Dolin
Microsoft