Schema.org Actions

A taxonomy of verbs

goto@google.com
Requirements

- model past, present and future activities
- model discovery and execution of activities
Use cases: Gmail
Use cases: Google+

- Prefilled text
- Call-to-action button to a URL or deep link to action in your mobile app
- Target URL can also deep link to content in mobile apps
- Preview snippet is rendered from the target page
Use cases: Search

Google search for "my hotel reservation next week"

Hotel reservations

Only you can see this result

From Fairfield Inn

Fairfield Inn Atlantic City North

Check in on May 15, 2013

Check in

Wednesday, May 15, 2013

Check out

Thursday, May 16, 2013

Location

405 East Absecon Boulevard
Absecon, NJ 08201
United States
Get directions

View email for this reservation
Use case: Now

Virgin America
flight 25

Status: Delayed / Mon, 29 Oct 2012
Depart New York
JFK 12:30 PM (12:00 PM)
Terminal 2, Gate 54B
Arrive San Francisco
SFO 1:56 PM (1:31 PM)
Terminal 4

Navigate to JFK / 32 min
View email

Le Gavroche
43 Upper Brook Street, London, W1K 7QR

Reservation in 1 hour
Journey time walking 45 minutes

Get directions
View email
Design

Goals

- **Rich set of nouns**: how do you describe a Hotel?
- **Disambiguation**: what does it mean to receive a gift? is the gift mine now?
- **Structure for Synonyms**: what’s the difference between writing and creating?
- **Contextual Arguments**: what arguments does buying have?
- **Execution and Handling**: how can actions be fulfilled?
- **Structured Composition**: can the result of an action serve as an argument of another?

Principles

- Verbs are top level types
- Verb arguments refer to semantic roles
- Hierarchy specialization
- Allow Reciprocals
- Allow Antagonyms
Implementation
**Thing > Action > UpdateAction > AddAction > InsertAction > AppendAction**

The act of inserting at the end of an ordered collection.

<table>
<thead>
<tr>
<th>Property</th>
<th>Expected Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>additionalType</td>
<td>URL</td>
<td>An additional type for the item, typically used for adding more specific types from external vocabularies in microdata syntax. This is a relationship between something and a class that the thing is in. In RDFa syntax, it is better to use the &quot;typeof&quot; attribute - for multiple types. Schema.org tools may have only weaker understanding of extra types, in particular those defined externally.</td>
</tr>
<tr>
<td>url</td>
<td>URL</td>
<td>URL of the item.</td>
</tr>
<tr>
<td>description</td>
<td>Text</td>
<td>A short description of the item.</td>
</tr>
<tr>
<td>image</td>
<td>URL</td>
<td>URL of an image of the item.</td>
</tr>
<tr>
<td>name</td>
<td>Text</td>
<td>The name of the item.</td>
</tr>
</tbody>
</table>

**Properties from Action**

<table>
<thead>
<tr>
<th>Property</th>
<th>Expected Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>agent</td>
<td>Person or Organization</td>
<td>The direct performer or driver of the action (animate or inanimate). e.g. <em>John</em> wrote a book.</td>
</tr>
<tr>
<td>participant</td>
<td>Person or Organization</td>
<td>Other co-agents that participated in the action indirectly. e.g. John wrote a book with <em>Steve</em>.</td>
</tr>
<tr>
<td>object</td>
<td>Thing</td>
<td>The object upon the action is carried out, whose state is kept intact or changed. Also known as the semantic roles patient, affected or undergoer (which change their state) or theme (which doesn’t). e.g. John read <em>a book</em>.</td>
</tr>
<tr>
<td>result</td>
<td>Thing</td>
<td>The result produced in the action. e.g. John wrote <em>a book</em>.</td>
</tr>
<tr>
<td>location</td>
<td>Place or PostalAddress</td>
<td>The location where the action occurred. e.g. John wrote a book at <em>his house</em>.</td>
</tr>
<tr>
<td>instrument</td>
<td>Thing</td>
<td>The object that helped the agent perform the action. e.g. John wrote a book with <em>a pen</em>.</td>
</tr>
<tr>
<td>startTime</td>
<td>DateTime</td>
<td>When the Action was performed: start time. This is for actions that span a period of time. e.g. John wrote a book from <em>January</em> to <em>December</em>.</td>
</tr>
<tr>
<td>endTime</td>
<td>DateTime</td>
<td>When the Action was performed: end time. This is for actions that span a period of time. e.g. John wrote a book from January to <em>December</em>.</td>
</tr>
</tbody>
</table>

**Properties from UpdateAction**

<table>
<thead>
<tr>
<th>Property</th>
<th></th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>collection</td>
<td>Thing</td>
<td>A sub property of object. The collection target of the action.</td>
</tr>
</tbody>
</table>

**Properties from AddAction**

<table>
<thead>
<tr>
<th>Property</th>
<th></th>
<th>Description</th>
</tr>
</thead>
</table>

**Properties from InsertAction**

<table>
<thead>
<tr>
<th>Property</th>
<th></th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>tLOccurrence</td>
<td>Place or Number</td>
<td>A sub property of location. The final location of the object or the agent after the action.</td>
</tr>
</tbody>
</table>

**Properties from AppendAction**

<table>
<thead>
<tr>
<th>Property</th>
<th></th>
<th>Description</th>
</tr>
</thead>
</table>

// John bought a book using amazon.com
<script type="application/ld+json">
{
    "@context": "http://schema.org",
    "@type": "BuyAction",
    "agent": {
        "@type": "Person",
        "name": "John"
    },
    "object": {
        "@type": "Book",
        "name": "Outliers"
    },
    "instrument": {
        "@type": "Product",
        "name": "Amazon"
    }
}
</script>
Example #2: Action handling

// Confirms a table reservation.
<script type="application/ld+json">
{
  "@context": "http://schema.org",
  "@type": "ConfirmAction",
  "object": {
    "@type": "Reservation",
    "name": "Bar Tartine"
  },
  "handler": {
    "@type": "HttpActionHandler",
    "url": "http://opentable.com/1234"
  }
}
</script>

<!-- Buys products -->
<form itemprop="http://schema.org/BuyAction">
  <input type="text" itemprop="object"></input>
</form>

<!-- Confirms a table reservation. -->

Questions?

goto@google.com

http://lists.w3.org/Archives/Public/public-vocabs/

http://sdo-actions.appspot.com/Action