Representing Change Tracking in XML Markup

Robin La Fontaine, Nigel Whitaker and Tristan Mitchell

DELTAXML
EXPERTS IN INFORMATION CHANGE

Background

- Generic change tracking format originally developed for OpenDocument
- Interest from DITA and other XML schema groups to have change tracking
- DITA viewed it as an XML problem not a DITA problem
- W3C Community Group formed Nov 2012 to develop this

Benefits of a change tracking standard

DELTAXML
EXPERTS IN INFORMATION CHANGE

- Documents with tracked changes could be moved from one XML editor to another
- XML editing applications could track changes in any XML document type
- Any XML document type could include a change history and the ability to roll back to previous versions
- Software designed to handle change in XML could be applied to many different XML document types

DELTAXML
EXPERTS IN INFORMATION CHANGE

Where we're headed

- Aims of the approach
- First steps - addition and deletion
- "That's not what I did!" - refining changes to hierarchy
- When two become one - adding merge capabilities
- Questions

DELTAXML
EXPERTS IN INFORMATION CHANGE

Aims of the approach

- Changes to XML are represented generically within the XML itself
- All changes can be represented
- Changes can be reversed so that any version of the document can be recreated
- Change markup can easily be converted into Processing Instructions

DELTAXML
EXPERTS IN INFORMATION CHANGE

First Steps

```
<insertion-example>
  <para delta:insertion-type="insert-with-content"
        delta:insertion-change-idref="ct1234">
    The weather in February in Prague is cold.
  </para>
</insertion-example>
```

```
<deletion-example>
  <delta:removed-content delta:removal-change-idref="ct5678">
    <para>
      It was raining when I left the UK.
    </para>
  </delta:removed-content>
</deletion-example>
```

Attribute Changes

```
<attribute-examples>

  <!-- adding a new attribute -->
  <para ac:change001="ct1,insert,style" style="my-new-paragraph-style">
    Weather discussion deserves a new paragraph style!
  </para>

  <!-- removing an attribute -->
  <para ac:change123="ct1,remove,status,draft">
    This paragraph is ready for viewing.
  </para>

   <!-- modifying an attribute -->
  <para ac:change456="ct1,modify,subject,weather" subject="meteorology">
    The study of weather patterns is known as meteorology.
  </para>

</attribute-examples>
```

Text Changes

```
<text-change-example>
  <!-- Changing:
       The weather in Prague is snowy.
       to:
       The weather in Prague is cold. -->
  <para>
    The weather in Prague is
    <delta:removed-content delta:removal-change-idref="ct1">
      snowy
    </delta:removed-content>
    <delta:inserted-text-start delta:inserted-text-id="it123"/>
    cold
    <delta:inserted-text-end delta:inserted-text-idref="it123"/>.
  </para>
</text-change-example>
```

Level 1

- Element addition / deletion
- Attribute addition / deletion / modification
- Text addition / deletion
- This gives basic change tracking capabilities
- More useful than simple element addition and deletion
- Demo at www.deltaxml.com/xmlprague2013

DELTAXML
EXPERTS IN INFORMATION CHANGE

“That’s not what I did!”

Using Level 1 to describe individual word style changes. Here, a word is made bold with the addition of a span:

```
<span-addition>
  <!-- Paragraph content changes from:
       The weather in Prague is snowy.
       to:
       The weather in Prague is <span style="bold">snowy</span>.
  -->
  <para>
    The weather in Prague is
    <delta:removed-content delta:removal-change-idref="ct1">
      snowy
    </delta:removed-content>
    <span delta:insertion-type="insert-with-content"
          delta:insertion-change-idref='ct1234'
          style="bold">
      snowy
    </span>.
  </para>
</span-addition>
```

EXPERTS IN INFORMATION CHANGE

Hierarchy changes

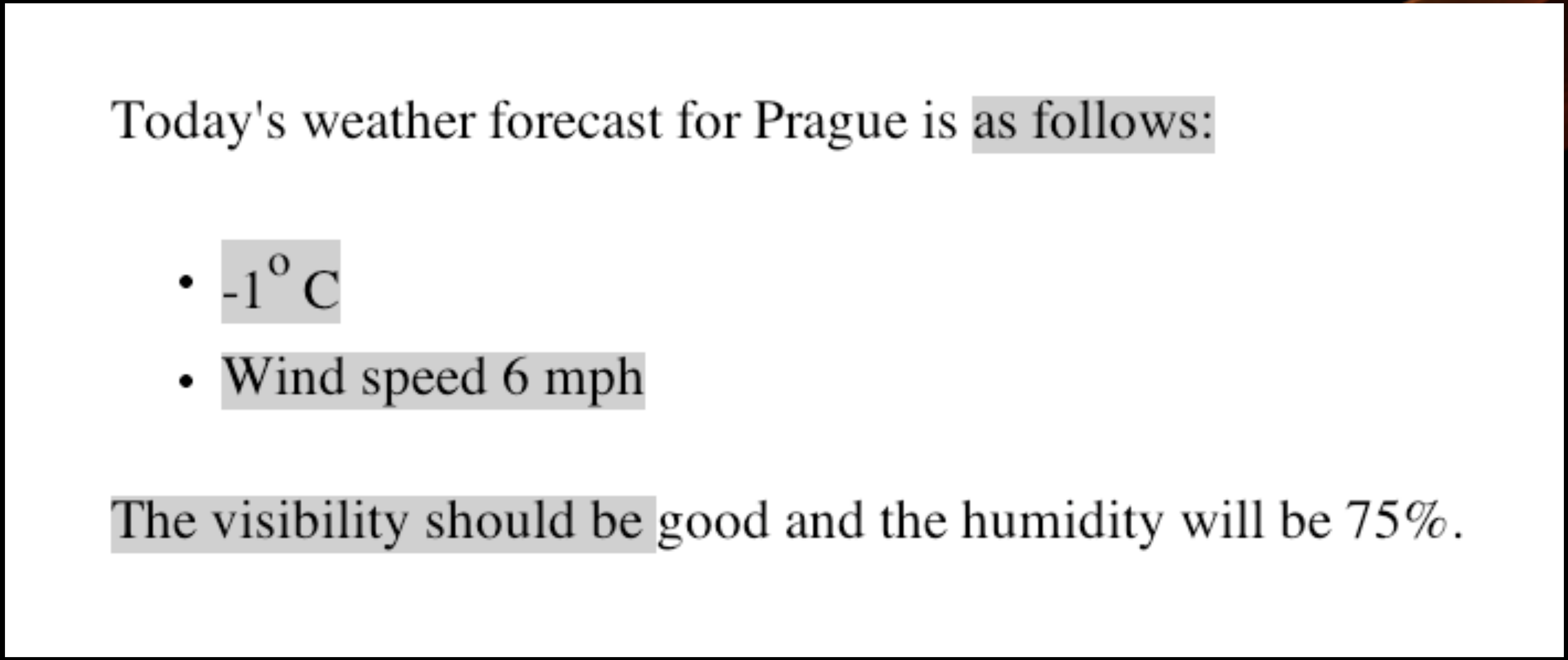
Defining a new insertion construct allows the tracking of structural changes without modifying their underlying content

```
<span-addition-level2>
  <!-- Paragraph content changes from:
       The weather in Prague is snowy.
       to:
       The weather in Prague is <span style="bold">snowy</span>.
  -->
  <para>
    The weather in Prague is
    <span delta:insertion-type="insert-around-content"
          delta:insertion-change-idref='ct1234'
          style="bold">
      snowy
    </span>.
  </para>
</span-addition-level2>
```

DELTAXML
EXPERTS IN INFORMATION CHANGE

When two become one

Merging elements together is such a common occurrence that the operation has its own construct in Level 2



Today's weather forecast for Prague is as follows:

- -1^{o} C
- Wind speed 6 mph

The visibility should be good and the humidity will be 75%.

DELTAXML
EXPERTS IN INFORMATION CHANGE

When two become one

```
<merge-example>
  <para>
    Today's weather forecast for Prague is
    <delta:merge delta:removal-change-idref="ct1">
      <delta:leading-partial-content>as follows:</delta:leading-partial-content>
      <delta:intermediate-content>
        <ul>
          <li>-1<sup>o</sup> C</li>
          <li>Wind speed 6 mph</li>
        </ul>
      </delta:intermediate-content>
      <delta:trailing-partial-content>
        <para>The visibility should be </para>
      </delta:trailing-partial-content>
    </delta:merge>
    good and the humidity will be 75%.
  </para>
</merge-example>
```

DELTAXML
EXPERTS IN INFORMATION CHANGE

Level 2

- Element addition around existing content
- Element deletion leaving content in place
- Merge elements together
- Split an element into two elements
- Gives finer granularity of change than Level 1

DELTAXML
EXPERTS IN INFORMATION CHANGE

Summary

- All changes are generically represented in XML
- Level 1 provides basic change tracking capabilities
- Level 2 improves the granularity and includes constructs for common operations
- Initial work has been completed and tested
- More work is needed on conflicting changesets and alternative representations
- Level 1 Demo: www.deltaxml.com/xmlprague2013

Questions

Remove leaving content

```
<span-removal>
  <!-- Paragraph content changes from:
       The weather in Prague is <span style="bold">snowy</span>.
       to:
       The weather in Prague is snowy.
   -->
  <para>
    The weather in Prague is
    <delta:remove-leaving-content-start delta:removal-change-idref="ct1234"
                                        delta:end-element-idref="ee567">
      <span style="bold"/>
    </delta:remove-leaving-content-start>
    snowy
    <delta:remove-leaving-content-end delta:end-element-id="ee567"/>.
  </para>
</span-removal>
```

Split example

```
<split-example>
  <!--
    Splitting:
    <para>The weather in Prague is cold. The weather in the UK is wet.</para>
    into:
    <para>The weather in Prague is cold. </para>
    <para>The weather in the UK is wet.</para>
  -->
  <para split:split01="sp1">The weather in Prague is cold. </para>
  <para delta:insertion-type="split"
        delta:insertion-change-idref="ct1"
        delta:split-id="sp1">The weather in the UK is wet.</para>
</split-example>
```