

## **About**

The project centers around tracking and representing the various personal offenses experienced by Emma Wormwood and the various others in her diary entries. The goal is to provide schemata, example markup, and an example XSLT transformation which allows individuals to document and display the offenses that occur in a diary, or for that matter, any sort of text. The core elements are designed to be flexible enough to apply to many different kinds of offenses (though only two appear in Emma Wormwood's diaries), but rigid enough to ensure that marked-up documents can be presented in a uniform and structured manner.

Marking up the offenses contained in a diary may aid in understanding social norms across societies and times. It also might serve as a companion to other writing about the same people, to elucidate the social dynamics implicit in a text. For example, perhaps an individual writes in their diary that they suffered a severe offense at the hands of person X. But person X's diary recounts the event as a trivial, lighthearted offense. The same action may be perceived in different ways. Furthermore, investigating the offenses of old can be fun, and humorous.

## **Elements**

The core elements of the project are the <offense-event> element, the <offending-action> element, and the <response> element.

**offense-event:** an event which results in offense being taken.

Attributes:

- Belligerents: A list of all the people involved in the offense
- offenseType: Specification of what sort of offense it was (physical blows, verbal insults, etc)
- Severity (Choose one: Trivial, Moderate, or Severe): Specifies how severe the offense was.
- May contain offending-action or response elements as descendant

Contents: text

**offending-action:** a single action within an offense event

Attributes:

- Offender: the name(s) of the offending party.
- Offended: the name(s) of the offended party.
- Provoked (Boolean): whether or not the action was provoked
- Presently may contain an insult element or a laugh-at element as descendant

Contents: text

**response:** a responding action within an offense event

May only contain an offending-action

The general structure of markup will be that the sentences describing an event which causes offense is contained in an offense-event element. Each action which contributed to this offense is marked as an offending-action, and where relevant, a response element. Within the offending-action element will be a particular element, which at present is limited to an insult or a laugh-at, but this list may be expanded should individuals happen across other kinds of offenses.

**insult:** a particular name or insult used in an offending-action

Attributes:

- nameCaller: who said the name/insult
- Target: who was the target of the name/insult
- nameKind: the type of insult the name/insult is (e.g. religious insult, name with disparaging implications)
- modernEquivalent: where relevant, modern insults with similar meanings (eg 'backwoodser' has a modern equivalent of hillbilly, etc.)
- Explanation: where relevant, explains what the name calling instance means (e.g. Emma calls Jesse a Heathen, which in view of the frequent mentions of going to church, is more severe than if someone called another a Heathen in an atheistic setting).

Contents: text

**laugh-at:** a particular instance of someone laughing at another's expense

Attributes:

- Laugher: the name of the person laughing

- Laughee: the name of the person being laughed at
- Reason: the reason that the laugher is laughing at the laughee.

Contents: text

### **Encoding Guide:**

As encoders mark up a document, they should contain a passage that recounts an offense with the offense-event element. Then, within the event, they should isolate the key action(s) and mark them as offending-action. Within the offending-action element, they should specify the kind of action it was. If need be, they may edit the schemata and XSLT to expand the possible offensive actions to include whatever they are marking up.

Expanding the schemata involves defining a new element for a new kind of offense. Then, reference that element within the offending-action element. The remainder of the schemata should remain unchanged. Schematron rules may be written if need be. In the XSLT file, encoders may add lines to template match onto their new element, and to the 'list of offenses' code, add an `xsl:if` statement, putting their new element within the 'test' attribute, and then list the attributes of their new element. For example, if I am encoding a document and I encounter a physical prank that causes offense, then I would first add to the relaxNG schema:

```

<define name="prank-element">
  <element name="prank">
    <attribute name="prankster"><text/></attribute>
    <attribute name="prankVictim"><text/></attribute>
    <attribute name="prankName"><text/></attribute>
    <text/>
  </element>
</define>

```

Then, to the offending-action-element definition, add the emboldened line:

```

<define name="offending-action-element">
  [Extra lines removed for brevity]
  <oneOrMore>
    <choice>
      <text/>
      <ref name="insult-element"/>
      <ref name="laugh-at-element"/>
      <ref name="prank-element"/>
    </choice>
  </oneOrMore>
</element>
</define>

```

And to the XSLT document, add the following chunk:

```

<xsl:template match="d:prank">
    <span class="prank">
        <xsl:apply-templates/>
    </span>
</xsl:template>

```

And among the list of offenses, add the following chunk in the ‘offending actions’ section, and ‘responses’ section:

```

<xsl:if test="d:prank">
    <br/><strong>Prank:</strong> <xsl:value-of select="d:prank"/>
    <br/><strong>Prank Type:</strong> <xsl:value-of select="d:insult/@prankName"/>
</xsl:if>

```

Do this only when a particular offending action does not fit into any predefined categories. If desired, an encoder may enhance the schematron rules with the emboldened lines:

```

<pattern>
    <rule context="wd:offense-event/wd:offending-action">
        <assert test="not(wd:prank) or @offender = wd:prank/@prankster">
            In an offending-action element, if the offense is a prank, the
prankster must match the offender attribute. Check your
capitalisation, spelling, and order of names if multiple people are
laughing.
        </assert>
    </rule>
</pattern>

```

Among the other checks for offender/nameCaller/laugher rules, and the same for the offended/target/laughee rules.