Encoding Rationale for Chapter 17 XML Annotation

Group: #12

Editor: Zijun Ye, Aditya Kandel

The encoding of Chapter 17 from Mary Shelley's *Frankenstein* follows the Text Encoding Initiative (TEI) guidelines, and the encoding methodology presented in Chapter 7 of *Doing Digital Humanities: Practice, Training, Research* by Julia Flanders, Syd Bauman, and Sarah Connell. The XML file was created using Oxygen XML Editor and VS Code. This encoding supports our future critical apparatus. Specifically, it examines the emotional dynamics between Victor and the creature, how their internal states are influenced by justice and ethics, and how the author uses nature to reflect and underline these emotions. This rationale explains the choices made regarding annotations, elements, and attributes.

Overall Structure

The XML file is organized into three main segments: <teiHeader>, <body>, and <back>. Each serves a specific purpose:

1. **<teiHeader>**: Contains metadata, including the book title, author, editor, publication details, and source. This ensures the encoding is well-documented and easily referenceable.
2. **<body>**: This section contains the full text of Chapter 17, accompanied by annotations. It begins with a <head> element to indicate the chapter title, distinguishing it from other chapters.

   The content is structured as either paragraphs or dialogues. Paragraphs are encoded using <p>, while dialogues are marked with <said> tags that include a who attribute to link speech to specific characters. The use of <said who=""> clearly separates dialogue from narrative text, making it easy to identify which character—such as Victor or the creature and assist for future critical analysis.

   To improve readability, annotations(<note>) are placed close to the relevant text rather than at the end of the chapter. We observed from previous encoding projects that placing annotations at the end made reading difficult, requiring excessive scrolling to access notes.
3. **<back>**: This section includes a character list (<listPerson>) and a bibliography (<listBibl>), with each list clearly separated by a <head> tag for better organization.

The character list uses <person> tags with corresponding xml:id attributes to link characters to their utterances in <said> within the text. This setup enables character analysis and provides insights into the narrator's role.

The bibliography catalogues citations, with each entry enclosed in <bibl> tags. These include elements such as <title>, <author>, <date>, <publisher>, and <pubPlace>, ensuring citation details. <note> are added to clarify where and how each citation is used.

Annotation and Element Choices

Annotations

All annotations are implemented using the <note> element, but they serve different purposes. To differentiate these purposes, the “type” attribute is used:

1. **Wording Explanation (type="gloss")**: This is used for definitions or clarifications of specific words or phrases. For instance, words like "Base” and "Detestation" are explained in context, often with references to dictionary definitions to clarify meanings relevant to the text. This ensures accurate interpretation where words have multiple possible meanings.
2. **Interpretative Commentary (type="annotation")**: This general-purpose type is used for interpretive insights or to highlight varying interpretations of terms. For example, the term "sea of ice" is annotated to explore its symbolic relevance in the narrative.
3. **Metaphorical Analysis (type="metaphor")**: This is dedicated to identifying literary devices like metaphors, particularly when nature reflects Victor’s emotions. For example, the phrase "eternal twinkling of the stars" is annotated to illustrate how Shelley uses natural imagery to mirror Victor's internal state.

Terms

The <term> element, paired with a unique xml:id attribute, connects specific text parts to their corresponding <note> annotations. This semantic tagging highlights keywords or phrases, indicating to users that additional explanations are available.

Troubleshooting Strategies

Managing overlapping

Some terms, like "justice," appear multiple times in the content, often with distinct nuances. In the beginning, two same ‘ids’ throw an error in our XML file. By searching through TEI documentation and community forums, we found out the requirement of ‘xml:id’ attributes must be uniquely referenced. We applied the solution to assign separate <note> annotations with distinct ‘ids’, differentiated by numbering (e.g., #justice01, #justice02) for each context. This approach ensures that each occurrence of "justice" is correctly identified and avoids potential confusion.

Proofread and debug

During the encoding process, we initially used VS Code for most tasks but encountered challenges in detecting syntax errors. We switched to using the Oxygen XML editor and found out that the editor could help us detect syntax errors. We also discovered the feature of Oxygen - ‘Apply Transformation Scenario(s)’ that can transform our XML file into XHTML format. This preview function improved our ability to debug and proofread the encoded text.