**Step-by-Step Guide for Ontology Modeling (G3-A2)**

| **Step** | **Action** | **Example/Code** | **Purpose** |
| --- | --- | --- | --- |
| **1. Create Data Property** | Define hasARIAState for toggle switch state. | **In Protégé:** - Go to Data Properties → Add. - Name: hasARIAState - **Domain**: ToggleSwitchComponent - **Range**: xsd:string | Model ARIA state (e.g., aria-checked). |
| **2. Add Restriction to**ToggleSwitchComponent | Restrict allowed values for hasARIAState. | **Manchester Syntax:** hasARIAState some { "aria-checked='true'", "aria-checked='false'" } | Ensure toggle switches only use valid ARIA states. |
| **3. Create an Instance** | Add a ToggleSwitchComponent instance. | **Turtle Syntax:** :DarkModeToggle a :ToggleSwitchComponent . | Represent a real-world toggle (e.g., dark mode switch). |
| **4. Add Data Property Assertion** | Assign ARIA state to the instance. | **Turtle Syntax:** :DarkModeToggle :hasARIAState "aria-checked='true'" . | Declare the toggle’s state for screen readers. |
| **5. Link to Accessibility Principles** | Associate with PerceivablePrinciple. | **Turtle Syntax:** :DarkModeToggle :requiresPrinciple :PerceivablePrinciple . | Explain compliance with accessibility standards. |
| **6. Run Reasoner** | Validate consistency in Protégé. | **Steps:** 1. Go to Reasoner → HermiT. 2. Click Start Reasoner. | Flag inconsistent instances (e.g., invalid hasARIAState). |
| **7. SPARQL Query for Validation** | Retrieve compliant toggle switches. | **SPARQL Query:** sparql<br> SELECT ?toggle ?ariaState<br> WHERE {<br> ?toggle a :ToggleSwitchComponent ;<br> :hasARIAState ?ariaState .<br> FILTER (?ariaState IN ("aria-checked='true'", "aria-checked='false'"))<br> }<br> | Verify which toggles have valid ARIA states. |