|  |  |
| --- | --- |
| **To-do’s, issues, and tasks** | **Results, updates, and/or suggestions (non-immediate, future items)** |
| * To verify the original ampo ontology with its proposed cmap (<https://github.com/congruili/Additive-Manufacturing-Processing-Ontology>) as it appears that there are missing classes and properties in the ampo.ttl while shown in the cmap (<https://cmapscloud.ihmc.us/rid=1Q2LYB6N7-HKDXKF-82/AMOnto_v1.7.cmap>), and vice versa. Found following issues: * **For CMAP:**   + These properties are missing or not     - foaf:mbox     - foaf:name     - foaf:person     - reverse properties not shown (that’s okay)   + Fix sio class names to lower cases in cmap   + ampo:RawMaterial should be ampo:Material in cmap. * **For the ontology (ampo.ttl)**   + Ampo:Material’s literal definition in ampo.ttl could be improved. (Must it be “modified”?) Same applies to ampo:SupportMaterial, etc.   + Add ”ampo:hasPreviousStep” and “ampo:hasNextStep”   + SIO classes were not using prefixed naming   + Issues with importing SIO, not QUDT   + Need to add prefix for (<http://qudt.org/1.1/vocab/unit>)   + assert qudt:Quantity explicitly, just like prov:Activity and foaf:Person, as our classes from auxiliary ontologies   + add new individual participants | * *(Improvements done in* ***ampo-no-import.ttl****)* * Fix these 2 prefix definitions from  **@prefix sio: <http://semanticscience.org/ontology/sio.owl#> @prefix sio: <http://data.qudt.org/qudt/owl/1.0.0/qudt.owl#>** to **@prefix sio: <http://semanticscience.org/resource/> @prefix qudt: <http://data.nasa.gov/qudt/owl/qudt#>** * Changed the following references to entities from sio and qudt: **<http://semanticscience.org/resource/SIO\_000614>** 🡪 **sio:attribute <http://semanticscience.org/resource/SIO\_000776>** 🡪 **sio:object**   **<http://semanticscience.org/resource/SIO\_000006>** 🡪 **sio:process <http://data.nasa.gov/qudt/owl/qudt#Quantity> 🡪 qudt:Quantity**   * Added ”ampo:hasPreviousStep” and “ampo:hasNextStep” * Added **@prefix qudt: <http://qudt.org/1.1/vocab/unit#>** * QUDT:   + The qudt ontology used was version 1.0.0 developed with NASA (<http://data.qudt.org/qudt/owl/1.0.0/qudt.owl#>) with IRI prefixes “http://data.nasa.gov/qudt/owl/qudt#”.   + The ontology developed several new versions. We will need to revisit this when we need to define more detailed quantities. * “prov”startedAtTime” and “prov:endedAtTime” * “prov:Agent/Person/…” * ampo:downloadURL’s domain does not include all types of input and attributes. Why? * ampo:isRecurring is currently commented-out. Future attention needed. * asserted qudt:Quantity explicitly * added Hao, James, and Uduak |
| * **Improve and extends the ampo ontology\***   + ampo:Attribute can be attribute of ampo:Input to enable input as a collection rather than a single item   + ampo:Input should ALSO be linked with ampo:Step and ampo:Process as they are specific to to each instances   + Need to create property combinesToForm | * (Following improvements done in “ampo-1.5.ttl”) * Verified that ampo:Attribute can be attribute of ampo:Input * Defined the new ampo:Participation class and relating structure. Used an intermediate class to collect inputs:   ampo:EquipmentUsage rdfs:subClassOf prov:Usage .  ampo:MaterialUsage rdfs:subClassOf prov:Usage .  ampo:ModelUsage rdfs:subClassOf prov:Usage .  ampo:qualifiedEquipmentUsage rdf:type owl:ObjectProperty ;  ampo:qualifiedMaterialUsage rdf:type owl:ObjectProperty ;  ampo:qualifiedModelUsage rdf:type owl:ObjectProperty ;  ampo:equipment rdfs:subPropertyOf prov:entity  ampo:material rdfs:subPropertyOf prov:entity  ampo:model rdfs:subPropertyOf prov:entity  ampo:inputRequiredBy  ampo:requiresInput |
| * Create a new “**ampo-ink**” ontology which extends ampo for our inkjet printing use case   + Create classes (sub-classes especially for equipments, materials, attributes for equipments, input for equipments   + Create sub-classes of ampo:Attribute and ampo:Input   + Find solution for actuator’s 2 stages     1. This need to be considered together with input association with steps – do we need an intermediate class to govern the operational input and status of the equipment?     2. For the above, refer to prov-o   + (Optional) Create sub-properties of ampo:isInputOf and ampo:hasAttribute, and use their domain/range classes to make proper restrictions | (following improvements done in “ampo-ink.ttl”) |
| * Enabling a new experimental design in **ampo-ink**   + Droplet actuation should produce a droplet as a product, which is measured during dropping (well, after dropping it’s not a droplet any more) and the following data be collected:     1. Droplet ejected: Y/N     2. Diameter     3. Velocity     4. Satellites: Y/N     5. Tail: Y/N | Related classes and properties created. |