Articulation = the action of putting into words an idea or feeling of a specified type

Ontology Articulation Guidelines

* What is an ontology? What can it be used for?
	+ Is it only in the “eye of the beholder”?
	+ What “should an ontology BE”? (There is no one answer)
	+ Appls – alone vs coupled with ML, for NLP/NLU, for conceptual modeling, for improved search, for QA, …
* Before development … Define questions to be asked related to:
	+ Purpose and usage (taxonomy, data integration/fusion, use in application, ?)
		- What are the problems to be solved?
	+ Stakeholders in development?
	+ Audience (other computers, computers and people, only techies, only SMEs, business persons, lay persons, ?)
		- Formal/logical definitions
		- NL definitions
		- Different communities need different types of communications. They even may use different vocabularies.
	+ Scope/domain (broad or narrow, focused on changes/events or things, ?)
		- Including timeframe to be addressed
		- Can discuss how this overlaps with “systems thinking” (What is your 'system'? What are its boundaries? What depth do you need to go to? Temporal aspects of an entity are important to know to understand what needs to be represented.)
		- Who, what, where, when, why, how is good approach that avoids philosophical debate
		- But, there needs to be some appreciation for the philosophical in order to avoid mistakes (This needs to be directly relevant to development and not a broad philosophy discussion.)
		- If in an application, what is dictated by that environment or by the dev team?
	+ Physical/real scope or conceptual/mental?
		- One world or Possible worlds
		- Remember that each person’s view of the world is unique, with their own biases and preconceptions. Even scientific worlds/rules change as more knowledge becomes available.
		- Is argumentation/defeasible logic needed? => Probabilities and strong focus on provenace (Who said what, when? And, what triples came from where and when and how?)
			* Possible worlds can be based on evidence, interpretation, the future and its scenarios, ...
	+ Need to understand cause-effect? Then need events/states/actions.
	+ Multi-lingual?
	+ Needing logic/reasoning/inference?
	+ Needing probabilities?
	+ Intended for reuse?
	+ How will it be maintained?
	+ How will it be populated by instances?
	+ …
* Starting development based on the different answers to the above questions, with examples
	+ Summary of the different schools of thought, methods, tools and techniques that different people use and recommend, as well as the overlap and underlap
	+ Language options (RDFS, OWL, OWL w reasoning, CL, …)
		- Pros/cons of each choice
	+ Inspiration or direct reuse from existing designs
	+ Different answers to the above -> different development and design choices
	+ Tool choices
		- Text edit (not recommended for beginners), Protégé, TopBraid, …
	+ Version numbering and management
	+ Creating definitions
	+ Provenance of the concepts
	+ Defining IRIs/naming schemes for the concepts
		- Never assume that you can name something and the name never changes.
	+ …
* Things to understand:
	+ Make context explicit (these are the answers to the “before development” questions)
	+ Nice to find common abstractions, but don’t assume that every use of a word will necessarily point to a common concept
	+ Combination of top-down and bottom-up
	+ Use of TLOs and ODPs
		- "To a person with a hammer, every problem looks like a nail." A person who has made a psychological commitment to a particular tool or TLO will change the problem to suit their tool rather than the other way around.
	+ Generalization/specialization and “how deep down the rabbit hole do you go?”
	+ Modeling relationships/properties
	+ Writing rules/axioms
	+ Don’t “boil the ocean”
	+ …
* Evaluation, feedback and evolution
	+ Evaluation metrics
	+ Getting feedback from lay persons
		- Spreadsheet review
		- Protégé review
		- Online and offline feedback
		- …
	+ Change management
* Publishing and reuse
	+ Reuse at what level? (whole, part, inspirational)
	+ How to reuse? (Reference, copy, …)
	+ What happens when your underlying (reused) ontology changes?
	+ …
* Mapping and Harmonization