# A Common Machine-Readable Vocabulary and Knowledge Base Supporting Multiple Programming Models

Chunhua Liao
Lawrence Livermore National Laboratory, liao6@llnl.gov

**Topic**: programming systems, emerging technologies, codesign methodologies

**Challenge**: Designing and implementing HPC programming systems requires a wide range of diverse knowledge about multiple layers of software/hardware stack, including science disciplines, application domains, compilers, runtime systems and hardware. For example, understanding and communicating high-level domain and application semantics related to data dependence can enable a wide range of compiler and runtime optimizations aimed for improving parallelism and reducing unnecessary data movement. However, such knowledge has been either implicit or expressed in an ad-hoc manner in each domain. People from difficult domains may not even talk in the same vocabulary for semantically identical concepts. Different programming systems and compilers have to develop different interfaces or annotations to redundantly encode the same information. The ad-hoc management of heterogeneous information from multiple layers of the HPC stack causes unnecessary burden for both developers and users of the programming systems. It also makes co-design of HPC systems difficult in general.

**Opportunity**: What if as a community, we can collectively and systematically accumulate, share, and reuse formally defined, machine-readable, and human-friendly knowledge across multiple layers of software/hardware stack? If such a vision is realized, it will enable stakeholders from different backgrounds to easily collaborate. Different software and hardware components in the HPC stack will also be interoperable. A variety of programming systems can be designed and tested on top of a reusable knowledge base of different science disciplines, application domains, software packages, and hardware platforms.

Advances in the knowledge representation community may already generate sufficient techniques and tools to help HPC researchers create a formal and machine-readable knowledge base across different layers of software/hardware stack. For example, a starting point for any group of people to collaborate is to have a common vocabulary, taxonomy and properties to describe one or more domains. Ontology techniques define a systematic approach to capture and represent concepts, instances and their relations for a domain. The Resource Description Framework (RDF) data model[6] encodes knowledge in the form of subject-predicate-object expressions. These techniques are very relevant for building HPC programming systems since a key driven factor for HPC optimization is the extracted software and hardware properties.

On the other hand, this vision of course has its unique challenges. For one, no single person can understand all the domains of HPC. It requires organized efforts to start from a smaller

domain then incrementally aggregate smaller vocabularies and knowledge bases into more comprehensive ones.  Another challenge is knowledge engineering, including domain knowledge gathering and verification, is still a labor intensive process.

**Timeliness or maturity**:

Knowledge representations and knowledge bases have been studied for decades. They have increasingly been used in many domains as the related techniques such as Web Ontology Language[4], Resource Description Framework[6], and JSON-LD[5] mature. For example, Schema.org[2] is a collaborative vocabulary started by Google, Microsoft, Yahoo etc. to annotate the Internet with structured data. Wikidata[3] is another collaboratively edited multilingual knowledge graph managed by the Wikimedia Foundation (who runs Wikipedia). More recently, Yago 4[1] builds one of the most comprehensive knowledge bases on top of crowd-sourced wikipedia articles. Linked Open Vocabularies (LOV)[7] gathers more than 700 vocabularies from different domains and provides popularity statistics and a searchable interface for users to find the right choices.

With all the aforementioned efforts going on, It is a good time for the HPC community to investigate these techniques and build our own common vocabulary and shared knowledge base to facilitate co-design of programming systems and even the entire HPC systems.

**References**


1. Tanon, Thomas Pellissier, Gerhard Weikum, and Fabian Suchanek. "Yago 4: A reasonable knowledge base." In European Semantic Web Conference, pp. 583-596. Springer, Cham, 2020.
2. Guha, Ramanathan V., Dan Brickley, and Steve Macbeth. "Schema. org: evolution of structured data on the web." Communications of the ACM 59, no. 2 (2016): 44-51.
3. Vrandečić, Denny, and Markus Krötzsch. "Wikidata: a free collaborative knowledgebase." Communications of the ACM 57, no. 10 (2014): 78-85.
4. Motik, Boris, Bernardo Cuenca Grau, Ian Horrocks, Zhe Wu, Achille Fokoue, and Carsten Lutz. "OWL 2 web ontology language profiles." W3C recommendation 27 (2009): 61.
5. Sporny, Manu, Dave Longley, Gregg Kellogg, Markus Lanthaler, and Niklas Lindström. "JSON-LD 1.0." W3C recommendation 16 (2014): 41.
6. Pan, Jeff Z. "Resource Description Framework." In Handbook on ontologies, pp. 71-90. Springer, Berlin, Heidelberg, 2009.
7. Vandenbussche, Pierre-Yves, Ghislain A. Atemezing, María Poveda-Villalón, and Bernard Vatant. "Linked Open Vocabularies (LOV): a gateway to reusable semantic vocabularies on the Web." Semantic Web 8, no. 3 (2017): 437-452.



This work was performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under Contract DE-AC52-07NA27344. The work was also supported by the U.S. DOE Advanced Scientific Computing Research. LLNL-ABS-819592