[Slide 1] Using Vocabularies to Provide Consistency Across Structured Content

Joseph Busch, Principal Analyst, Taxonomy Strategies

(presentation at Taxonomy Bootcamp London on April 15, 2026 at 1:30 PM)

Application profiles provide a framework for structuring information related to a particular application. These can include two types of vocabularies – schema and value vocabularies. Schema vocabularies such as SKOS, RDFS, Dublin Core, and Schema.org specify the properties and attributes related to the application, while value vocabularies (ranging from simple controlled vocabularies to taxonomies and ontologies) specify the valid instances for those properties and attributes. This talk provides an accessible overview of application profiles and makes the case for why this type of documentation is important in the context of artificial intelligence.

[Slide 2] An application profile defines a specific set of requirements, settings, and metadata for a particular application to ensure compatibility and functionality.

[Slide 3] The profile adapts general standards or frameworks to meet the needs of a specific use case, for example, using Dublin Core for bibliographic description and FOAF for named entity properties rather than specifying a whole schema from scratch.

[Slide 4] "An Application Profile (AP) is a data specification aimed to facilitate the data exchange in a well-defined application context. It re-uses concepts from one or more

semantic data specifications, while adding more specificity, by identifying mandatory, recommended, and optional elements, addressing particular application needs, and providing recommendations for controlled vocabularies to be used."[1]

[Slide 5] In the context of application profiles, there are two kinds of vocabularies – concept schemes (like metadata schemas) and semantic schemes (which are controlled vocabularies, including taxonomies, thesauri, and ontologies). Concept schemes structure the how of information, defining the elements and rules for metadata, while semantic schemes, such as taxonomies and thesauri, provide a consistent set of terms to describe the what of information in a structured way.

[Slide 6] The Purpose of concept schemes is to define the types and structures of metadata used for describing a collection of resources. For example, Dublin Core is a well-known concept scheme that provides a standard set of metadata elements for describing digital resources. Concept schemes are prescriptive, outlining how information should be organized and presented to ensure interoperability.

[Slide 7] The Purpose of semantic schemes is to provide a controlled set of terms and relationships for describing the content itself, ensuring consistency and clarity. For example,

[1] Application Profiles: What are they and how to model and reuse them properly? A look through the DCAT-AP example. SEMIC Support Centre. [n.d.] https://interoperable-europe.ec.europa.eu/collection/semic-support-centre/application-profiles-what-are-they-and-how-model-and-reuse-them-properly-look-through-dcat-ap

- **Taxonomies** organize concepts into hierarchical "is a" or "part of" relationships, creating a structured tree for concepts.
- **Thesauri** are a type of controlled vocabulary that includes preferred terms, synonyms (variant terms), broader terms, narrower terms, and related terms.
- **Authority Files** are controlled vocabularies specifically used for authoritative lists of names of people, organizations, places, and other entities.
- **Ontologies** are more complex semantic schemes that define concepts and their relationships with greater specificity than a thesaurus or taxonomy.

Semantic schemes aim to unify meaning and improve information retrieval by ensuring that terms are used consistently and unambiguously across a collection.

[Slide 8] A W3C Application Profile is a specification that defines a particular way of using one or more base specifications to achieve a specific function or interoperability goal within a community of practice.

[Slide 9] An application profile doesn't create entirely new web standards but rather specifies how existing W3C Recommendations (like RDF, SKOS, or DCAT specifications) should be used in a particular context. This might involve limiting the use of certain features, requiring specific data types or vocabularies, or defining how different specifications interact.

[Slide 10] The primary goal of an application profile is to facilitate out-of-the-box interoperability between different systems or applications within a defined domain. By adhering to a common profile, developers can ensure that their implementations can seamlessly exchange information and interact with each other.

[Slide 11] Application profiles are often developed to address the specific needs of a particular community or industry sector. For example, a profile might be created for sharing learning resource metadata (LRMI), for creating Dublin Core applications (DCAP), for tabular Dublin Core application profiles (DC TAP), or for sharing knowledge organization systems (NKOS AP).

[Slide 12] The vision of the application profile is to enable interoperability but the practice of documenting the requirements may be just as important.

[Slide 13] This is particularly important in the context of current artificial intelligence (AI) and data management practices for a number of reasons:

1) In the current large language model (LLM) and generative AI paradigm “everything is data”. This means that unstructured content such as untagged text and rich media as well as structured content and data itself are all within processing scope.
2) Generative AI does not truly understand language (or see images or hear music). It is a complicated statistical model that generates a response to a prompt (what you submit) token-by-token, each token predicting the next. Even though AI responses

appear to be compelling, the reality is that the same prompt returns a different response each time. The results are difficult to refine because the mechanisms by which they were generated are not apparent.

3) The AI user interface is natural language which lowers the bar to use but makes it more difficult to define what is a good practice than with a structured language (like Java, SQL, etc.)

[Slide 14] The good news is that Gen AI is very easy to use and everyone is using it. The bad news is that the results are unreliable. I think it's similar to the experience with web search. Everyone is looking things up. The results are good enough. This search provides answers to simple questions, but it can be frustrating to get definitive answers, find a known result, or do any sort of in-depth authoritative research. To improve the reliability of search you need to use better search terms. One way to do this is to look for and choose search terms from vocabularies. The same is true for improving the reliability of AI.

[Slide 15] The approaches to integrating vocabularies into prompts are the same as for integrating vocabularies in information retrieval:

- Incorporate synonyms (and quasi-synonyms) in the prompt using a domain thesaurus
- Broaden or narrow the concepts in the prompt using a hierarchical taxonomy
- Provide specific contexts using concept schemes such as content type, geography, language, time, process, or other contexts

While it's possible to manually refer to vocabularies while writing prompts, LLMs integrate with other systems through the "context window". The context window is the input text that the LLM can process. LLMs predict the next token based on the input sequence, and by manipulating this input, one can affect the model's output. The ChatGPT context window is capped at 32,000 tokens. A token is roughly equivalent to 4 characters or about three quarters of a word. But processing large context windows can be computationally expensive and slow.

[Slide 16] An application programming interface (API) allows for development of applications that can **integrate additional resources such as vocabularies and search results into prompts**. The ChatGPT context window is up to a million tokens when accessed via the API. In this context vocabularies can also be used to:

- Guide language generation results by identifying entities in the output, link entities to knowledge graphs, and connect models to knowledge bases;
- Index source content to improve document recall and precision; and
- Annotate output to improve explainability and trustworthiness.

Retrieval augmented generation (RAG) involves searching for relevant facts and inserting them into the context window to improve the model's answers. The approach can be expanded to include various data sources beyond search engines, such as an indexed domain specific collection, databases, and calendars, making it a complex but powerful tool for enhancing LLM performance. The response may include annotations, such as source citations or bibliographic references from the domain specific collection content. A

good way to think about RAG is as a method for pre-processing LLM prompts to make them more responsive to domain specific questions.

A knowledge graph is a representation of the relationships among named entities. A knowledge graph may be queried in the RAG method to improve the information retrieval with equivalent terms but especially to provide associative, typed related concepts. It can also be integrated as a post-process to an LLM response to identify possible refinements through reasoning, decomposition, verification, or other methods.
RAG applications may use information retrieval in different ways. In particular, they may not necessarily index content using vocabularies. They may use vector search to identify similar content statistically without any reference to vocabularies. A better way (to achieve more reliable and measurable better recall and precision) is to use vectors to map content to vocabularies, and then to use vocabularies to identify similar content.

[Slide 17] To summarize:

- Application profiles provide a framework for structuring information related to particular applications.
- There are important specifications for profile vocabularies developed by the Dublin Core and W3C communities.
- There are two kinds of vocabularies—metadata schemas and controlled vocabularies.
- The practice of documenting requirements may be more important than interoperability.

- Profiles may be a useful framework for integrating both types of vocabularies into AI applications to improve their reliability and transparency.

[Slide 18] Resources

These are links to the technical standards mentioned in this presentation.

- *Data Catalog Vocabulary* (DCAT) - Version 3 (https://www.w3.org/TR/vocab-dcat-3/), an RDF vocabulary designed to facilitate interoperability between data catalogs published on the Web. W3C Recommendation (22 August 2024)
- *DCAT-AP 3.0* (https://semiceu.github.io/DCAT-AP/releases/3.0.0/), a DCAT profile for sharing information about catalogs containing datasets and data services descriptions in Europe. SEMIC Support Centre (14 June 2024)
- *DCTAP* (https://www.dublincore.org/specifications/dctap/) Dublin Core Application Profiles Working Group (n.d.)
- *Dublin Core Application Profile Guidelines* (https://www.dublincore.org/specifications/dublin-core/application-profile-guidelines/) (2005-09-03)
- *Guidance on Publishing Application Profiles of Vocabularies* (https://www.w3.org/2017/dxwg/wiki/Guidance_on_publishing_application_profiles_of_vocabularies). W3C Dataset Exchange Working Group (last edited on 10 October 2025)
- *Networked Knowledge Organization Systems Dublin Core Application Profile* (NKOS AP) (https://nkos.dublincore.org/nkos-ap.html) DCMI NKOS Task group (2019-11-03)

- *The Profiles Vocabulary* (https://www.w3.org/TR/dx-prof/), an RDF vocabulary created to allow the machine-readable description of profiles of specifications for information resources. W3C Working Group Note (18 December 2019)

[Slide 19] More information

This is how to reach me and a link to a microsite related to this presentation -- “The Struggle to Make Language Models a Human Powered Instrument: How Did We Get Here and What Can We Do About This?”