ENTERPRISE KNOWLEDGE

Taxonomy Use Cases: How to Estimate Effort and Complexity

Riko Fluchel

When asked to define taxonomy, I like to define it as a method rather than a thing. I typically say taxonomy is a way of categorizing things hierarchically, from general to more specific. Sounds simple enough, right? After all, who *hasn't* been grouping together things that have something in common, and slapping a name on that group since they first learned to speak? Every store, every house, every website has a way of categorizing and labeling stuff so that everything belongs in a place. Everyone does it, so it should be easy… right?

As any seasoned taxonomist, librarian, or knowledge manager will tell you: *it depends.* Specifically, it depends on the purpose of the taxonomy and its intended users. Even subtle differences in purpose or audience in similar environments can lead to vastly different results. Have you ever completely failed to find something in someone else's kitchen? This is because it was not organized for you, just like you organized your kitchen with your own purposes and needs in mind. The use case, then, is intertwined with an audience or persona and a goal.

This white paper explores taxonomy use cases as an indicator of complexity, and how they can be used to determine the amount of effort that may be required for an organization to design a taxonomy. Effort refers to the amount of dedicated work and brain power that will be needed in order to design for a taxonomy's complexity, particularly the effort to maintain the taxonomy in the long run and ensure its future success.

Use Cases

Use cases will establish scope and purpose of a taxonomy. Defining complete and detailed use cases will make a difference in planning out an effort for taxonomy design. Use cases identify who will be using a taxonomy, how they will be using it, and why. These can be similar to **user stories** in the Agile methodology. Once defined, use cases will delineate relevant scope by defining Minimum Viable Product features, and help decide the direction of a taxonomy (MVP is like a prototype: what are the bare minimum efforts and features we need to put in to this product in order to learn the most about the impact of the product and iteratively expand it?). There may be numerous use cases for a single taxonomy, so it will be necessary to prioritize and create a backlog of use cases that will drive future iterations of a taxonomy. Since we typically focus on First Implementable Versions (a taxonomy MVP), we want to first focus on use cases that are easily compatible with each other and are attainable, recognizing that taxonomies can grow to incorporate future use cases once we have the foundation built.

A use case can be broken down into three parts: the persona, action, and goal. The persona represents an archetype of user that will be interacting with the taxonomy; this could also be a specific role, such as a Sales Representative. The action includes the steps a persona is taking while using the taxonomy; this should also include a specific system in which the taxonomy will be implemented. The goal is the persona's purpose for using the taxonomy.

Persona	Action	Goal
Who is the user?	What is the user doing?	What is the goal?
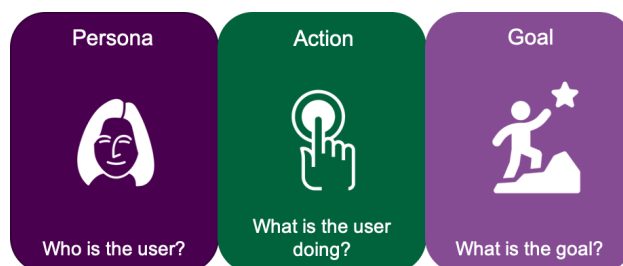
An example use case can be:

Clark the Customer (**persona**) needs to be able to use brand, color, and size facets on the customer shop of Shirts.com (**action**) so that they can find the perfect shirt for their upcoming interview (**goal**).

These specific details provide clear indicators of a successful taxonomy: we know that our taxonomy must describe clothing through facets (brand, color, size), including styles that are appropriate for interviews (this is a little bit of extra detail, but it can bring a use case to life). We know that the taxonomy must be implemented in faceted search and navigation on a specific system, so knowing whether this system has this capability is identified; there may also be implicit systems (such as databases) in the back-end we need to account for. Lastly, we will need to have a better understanding of how users currently go about using this system to achieve their goals, and what a taxonomy can do to improve the situation.

Classic Taxonomy Use Cases

Classic use cases for taxonomy include: tagging and **faceted search** for content, basic reporting or analytics, or creating organizational or navigational structures. These use cases are typically applied in content management repositories such as intranets and learning portals, or any other front-facing interfaces such as retail websites.

Classic use cases are people focused: a customer needs a navigational structure to be clear so that they can find what they're looking for when they need it. An employee needs to be able to search effectively to find the relevant training on the company learning portal in order to improve at their job. A revenue team needs to be able to classify products and services in one category in order to run reports on their profitability. A data governance team similarly needs to definitively classify data entities and attributes in a single category that corresponds to a business unit, in order to identify data stewards and owners for compliance purposes (such as GDPR or CCPA).

> Classic Use Cases are people focused and have a history of repeated implementation to rely on.

Classic use cases may appear to be less complex and therefore seem easier, but this is deceptive and not always true. Classic use cases can easily multiply into several use cases if it turns out there are multiple personas involved. For instance, you may have customers, sales representatives, and third-party vendors involved in a retail search and navigational use case in which each group has different needs from the taxonomy. Perhaps third party vendors need a way of managing product metadata, and sales representatives need to be able to track sales, while customers need clear facets to find products.

That being said, Classic Use Cases are "classic" because they've been implemented time and time again in systems that most organizations already have (unless they are adding an enterprise taxonomy tool to the mix, which will make a long term effort smoother); taxonomists and developers have reliable previous efforts to lean back on and may have a specific methodology for each use case that can be reused. Classic use cases tend to have a more predictable level of effort estimation that should also include other factors such as the complexity of the domain, the level of specificity or the breadth of concepts possible, and the type of content the taxonomy will be primarily organizing.

Advanced Use Cases

Advanced use cases tend to delve into **ontologies, knowledge graphs**, and **artificial intelligence**, but taxonomy is still a foundational aspect of these technologies. These use cases include text parsing and automated classification, predictive analytics, insight inferencing, chatbots, and **recommendation engines**. While people will still benefit from the end result of these use cases, the complexity of these taxonomies are amplified by the fact they are primarily meant to be utilized by machine learning processes that humans can't effectively reproduce, on a massive volume of data. A taxonomy meant purely for text parsing and auto-classification will not be directly intuitive or usable by people since these tend to be significantly larger, highly specific, or repetitive as a way to disambiguate concepts, and therefore highly complex. They may also have polyhierarchy or semantic relationships that go beyond hierarchy.

Advanced use cases will require a higher level of effort, more so than classic use cases. The barrier for entry is much higher than a classic use case, requiring specific knowledge regarding machine

learning and other semantic capabilities. Advanced use cases will use specific technology that many organizations don't have, unless they have some of these capabilities already, so new technology may need to be purchased and added to an organization's system architecture. This is also an actively developing field within artificial intelligence; while of course there are demonstrated successes, these use cases are open to experimentation as the field develops, and may face a higher degree of uncertainty (see my previous blog on **NLP and Taxonomy Design** to learn more about an example of an advanced use case).

> Advanced Use Cases rely on machine-learning processes, have a higher barrier of entry, and require more specialized efforts.

System Use Case Limitations

Systems that are in scope for taxonomy implementation should be noted as part of the action of a use case, in which a persona uses a system to interact with a taxonomy. The added element of a specific in-scope system opens the possibility of certain limitations that can dictate the design of a taxonomy, and will restrict other use cases. For example, some systems do not handle hierarchical values easily. If a taxonomy informs the values of a metadata field in this kind of system, that field will not be able to fully represent the hierarchy of the taxonomy.

This means the implementation of a taxonomy will have to get creative, but it also means the usable fields of the taxonomy may be limited to a certain level. In other words, only the lowest level in the taxonomy can be used as metadata values. The taxonomy must conform to this level across the board, and all areas of this taxonomy must go to a certain level of depth in order to be used. A good rule of thumb for taxonomies with the classic use cases is Three Levels.

A strict hierarchy and a strict number of levels that are both imposed by a system is great for classic use cases, but it will not be ideal for advanced use cases like text parsing. A limitation like this can make fulfilling an advanced use case exceedingly difficult, since certain levels of specificity will have to be sacrificed. This means that certain classic and advanced use cases are incompatible and may require different designs.

While system limitations don't necessarily change the level of effort of a taxonomy design, not knowing system limitations in advance has a risk for more effort if there needs to be significant rework (which can still be accounted for

> System Limitations can restrict different types of Use Cases that may make them incompatible.

ahead of time if we plan for constant iteration). However, as mentioned above, system limitations will have an effect on other use cases. As a result, the more systems that are selected to be a part of a taxonomy effort, the higher chance there are system limitations which can impact design decisions; this may increase the level of effort, and restrict the taxonomy's ability to fulfill other types of use cases, especially if each system roughly corresponds to a classic or advanced use case.

Mixed Use Cases as Indicators of Complexity

Multiple use cases for a taxonomy can be a sign of complex business needs. Multiple use cases can be due to the fact multiple groups of users or even departments are relying on a single taxonomy to achieve their specific goals. Likewise, multiple in-scope systems can indicate multiple groups of users or departments that use their own designated system, each with its own capabilities and limitations that may need to be accounted for.

Depending on the nature of these departments, even if they have the same use case, they may require different concepts or structures to be in the taxonomy. For example, if a global enterprise needs a taxonomy for their products and services, it is usually the case that regional offices offer unique services and products, or engage in markets/industries respective to their regions, but not others.

The implication being, this taxonomy will have parts that are not relevant to specific regions. This increases the potential for misalignment and **lower adoption** if not identified early on by establishing thorough use cases for each region, which may need to have the ability to designate

sections of a master enterprise taxonomy that are relevant to them.

While some use cases are very compatible with each other, every distinct use case for a taxonomy runs the risk of changing the nature or content of a taxonomy, thus potentially increasing the effort required. A taxonomy intended for search and navigation may be a different shape than a taxonomy for reporting, because these entail different users with different goals, even if the taxonomy is modeling the same information domain. As more use cases are introduced to a single taxonomy effort and the effort is not planned accordingly, the higher the risk of not being able to meet expectations, thus lowering adoption.

Prioritize Use Cases that are compatible with each other in the initial MVP effort.

Conclusion

It's important to emphasize again that we use terms like "First Implementable Version" and "Initial Design" for a reason: to set expectations that a taxonomy is necessarily iterative, and you don't need to tackle all possible use cases at once on Day 1. Similarly, expecting to achieve all of your possible use cases within a few months' initial design project is unrealistic. A sustained effort can be grown as value is realized with an MVP, and then more use cases, as well as the

advanced use cases, can eventually be explored. Start small, prioritize the first use cases to the ones that are compatible and attainable, realize and demonstrate the value of your MVP, and grow as necessary.

Taxonomy is incredibly flexible, and it can be designed in many different ways to suit your users' needs. Taxonomy is an elegant solution to complex, wide ranging yet common problems in the information world. Identifying and analyzing use cases, and considering the potential complexity represented by them, can be used as a way to estimate the effort required for an enterprise taxonomy. From here, a viable long-term roadmap can be created with realistic expectations and priorities.

Know you need a taxonomy, but unsure where to start? **Contact Enterprise Knowledge**'s team of expert taxonomists and KM consultants to learn more.