

Reuse and connect content

December 5, 2022 • Boris Pocatko and Tomas Nosek • 9 min read

Reuse what's reusable. You might notice that some content types contain similar or even the same elements. Don't duplicate, reuse instead.

Reuse repeated content and elements

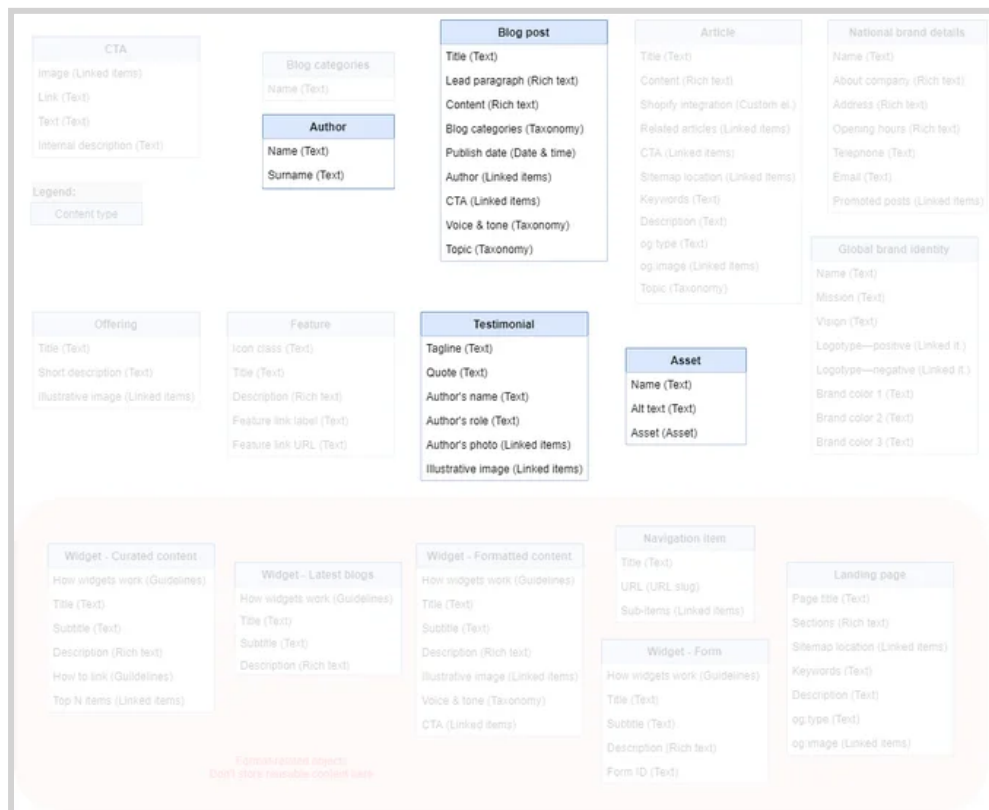
Repeated content

In the example content model, there are several places where repeated content is:

- *Blog post* contains an author, that will be repeated (one author usually writes many articles).
- Images are usually used in multiple places with the same title and alternative text.

The main criterion is **if the set of values for the element overlap**. In that case, **consider creating a separate content type representing the given set**. On the example website, there is also a listing of all authors and a filter set up so it makes even more sense to create a separate content type for them.

Images and other assets are usually distributed with additional information. It can be a title, an alternative text, or even tags. That's why it's suitable to [create a content type called Asset that will contain the asset](#) itself along with all the necessary information. The biggest advantage is that this implementation can be later easily extended.



You can also [view the diagram in diagrams.net](#) to zoom in and out or [browse through the website](#).

Repeated elements

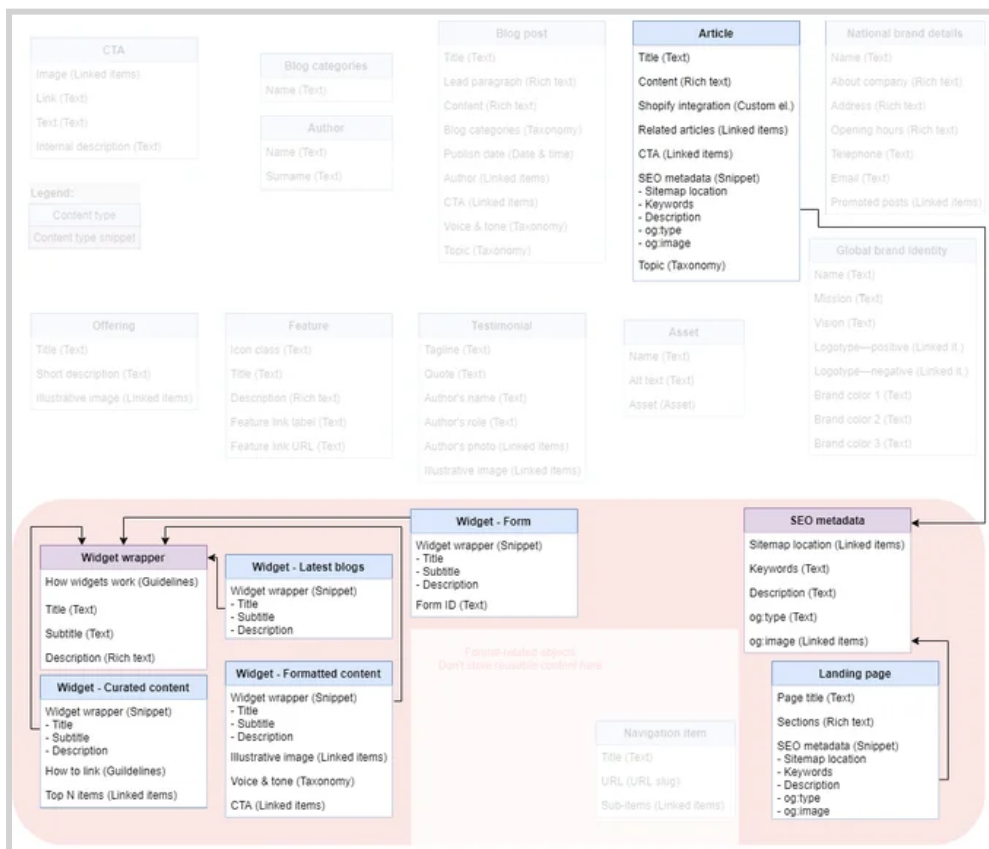
Again, in the example content model, there are several places with repeated elements:

- All widget types contain *How widgets work*, *Title*, and *Subtitle*.
- *Article* and *Blog post* contain *Title* or *Body copy*.

If the set of elements is only repeating without sharing the given set of content, consider using [content type snippets](#). They allow you to create a repeating set of elements for multiple content types.

The widgets in the example are suitable to be used as a content type snippet as the content in them won't be the same but they are going to be always used together as all widgets relate to each other.

On the other hand, you can see that for *Article* and *Blog post*, the *Title* and *Body copy* stayed separate. Even one element can be created as a content type snippet but in this case, we decided to keep them separate as the elements could later change independently of each other.



You can also [view the diagram in diagrams.net](#) to zoom in and out or [browse through the website](#).

A part of content repeats a lot?

If you have an element with a part of content that repeats a lot (like `https://` in URLs), **set a default value** for the element.

When you create a new content item, the element gets pre-filled with the default value and you can just add the parts that are different.

You can set default values for the asset, date & time, linked items, multiple choice, number, taxonomy, text, and [custom elements](#).

The screenshot shows the configuration panel for a 'Video link' element. It includes the following settings:

- Element required
- Element non-localizable
- Limit length to:
- Match a specific pattern
- Default value:
- Footer note: Add the ID of the video from Videoprovider.

Remember, if an element repeats with content as a whole, [convert it to a content item](#) and reuse it like that.

Connect the content types with relationships

One of the first sentences in this tutorial said that **content models are mainly about relationships among content types**. This step will be mainly about noting them in the diagram.

💡 If you already know database relationships, content model relationships are practically the same.

Types of relationships

It's important to specify types of relationships in your diagram as they are modeled differently later. Their type also depends on the purpose of the connection. There are three types of relationships:

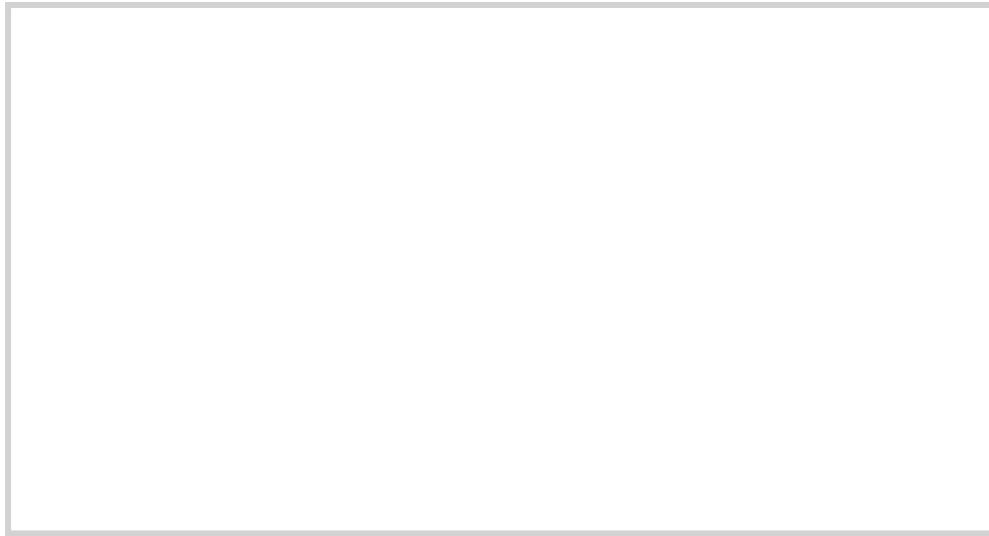
- **1:1 (one-to-one)** – means that one content item relates to exactly one other content item. It usually applies for one-time content that is not reusable or for categorization and other metadata. For example:

- Each department must have a manager, and each manager can manage one department only.
- Every person in most countries has one ID (e.g., Social Security number), and every ID has exactly one person assigned to it.
- In content modeling, you may want to have each article to have a call to action. That call to action is used in one article only.
- **1:N (one-to-many)** – means that one content item can relate to multiple content items, yet the other content items relate in that relationship to that one content item only. For example:
 - One account manager takes care of multiple accounts, yet every account always has one account manager.
 - A car brand (Ford) has multiple models (Mustang, Fiesta) but every model (Mustang) is produced under one brand (Ford).
 - As a content modeling example, when a navigation menu is expandable, a menu item has multiple sub-items, yet one sub-item always belongs only to one menu item.
 - Going to the most common use-case, related articles would also be an example of the 1:N relationship. An article is linked to multiple articles, yet in that relationship, the related articles are linked to that one article.
- **M:N (many-to-many)** – means that multiple content items can relate to multiple other items and vice versa. To use it in content modeling, these relationships require an additional content type used for purposes of the connection. For example:
 - Multiple authors can write multiple books. So, one or more authors can write a book, and one or more books can be written by one author or a group of authors.
 - Multiple doctors have multiple patients. So, one or more doctors have examined multiple patients, and multiple patients have been examined by multiple doctors.
 - In content modeling, if you want to categorize multiple articles and blog posts together because they all fit within a topic, you'll create the M:N relationship.

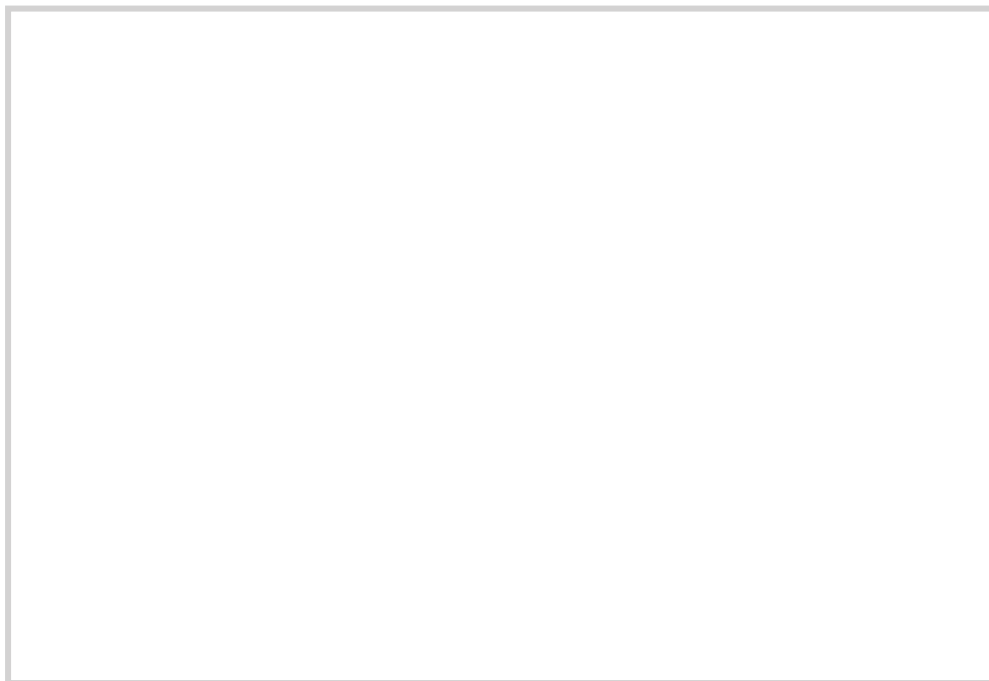
Is the relationship for categorization 1:N or M:N?


As examples in the list above, there are related articles as 1:N and blog posts on a topic as M:N. These two relationships are very similar when people are talking, yet for computers, those are two different things.

Related articles usually don't share the same group. Article A is related to articles B and C, but article B is related to A and D, not C. In this case, the relationship is 1:N as each (1) article has multiple (N) related articles. So, four articles would have their own 1:N relationship each.



However, if there are **articles on a specific topic**, all are related to each other within one group, which is represented as an M:N relationship. Article A is on the same topic as B, C, and D while article B is on the same topic as A, C, and D. The relationship is the same for all articles on the same topic.

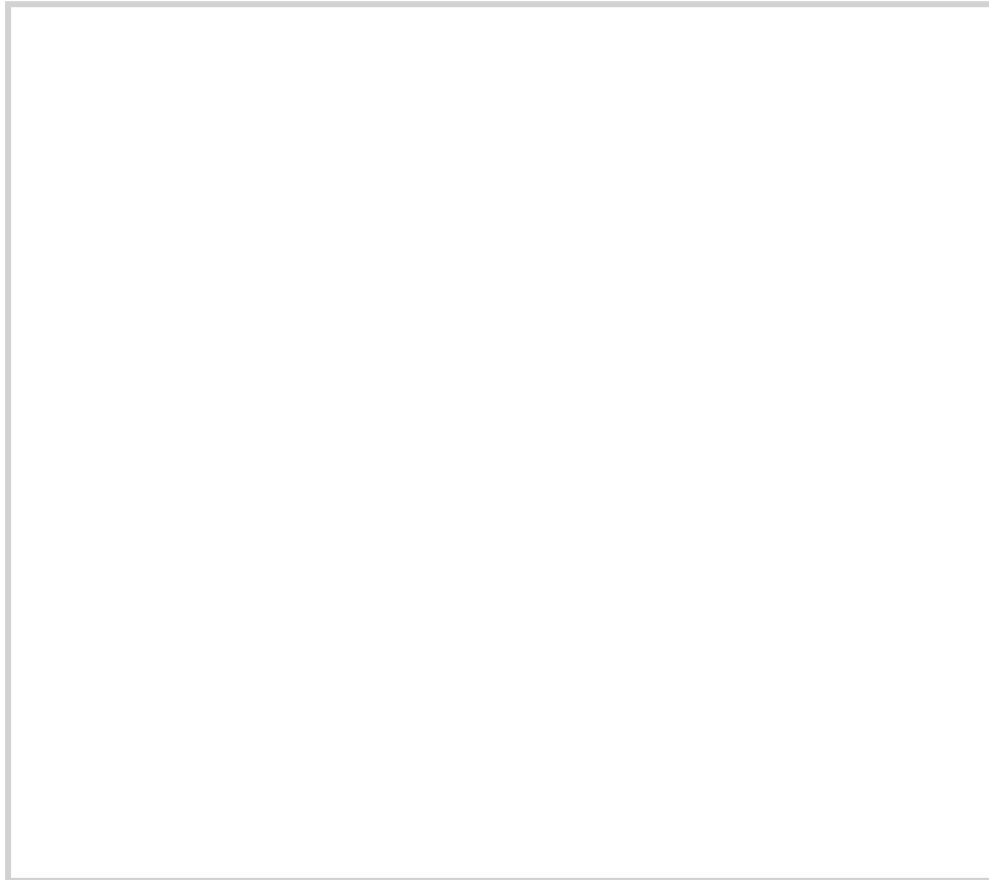


 Opposed to 1:N relationships, the great thing about M:N relationships is their flexibility. You can add them to any content model pretty much at any time, which means less content model re-designs in the future. The downside might be the required mind shift and the need for better content creator training.

Now, there's a question of whether the relationships are going to be implemented as content types or taxonomies in your content model. Don't think about it now, the next step will describe this part of the process.

Add relationships to the diagram

Draw relationships in your diagram as lines connecting the element with the related content type. Moreover, **every relationship can have a limitation**. For example, the example content model can have up to three related articles. Mark these limitations to the diagram as well.



You can also [view the diagram in diagrams.net](#) to zoom in and out or [browse through the website](#).

What's next?

Now that you understand the relationships in your content model and know how to approach reusing content, let's get to the next topic. The next step is to explore the hierarchy of your structure and leverage it for building your app navigation.

[Define relationships to create navigation](#)