

Metadata

– What, When, How, Why?

Stig Berild
(Myndigheten för skolutveckling)

“Metadata” is seen by many as a rather abstract and strange concept. Consequently also a host of different interpretations exist. The purpose of this report is to present my personal view in introducing the concept of “metadata” in chapters 2 and 3, and presenting personally flavoured reflections on the topic in chapters 4 and 5.

The examples discussed are taken from a National Agency for Education project on the use of so-called soft infrastructure for information management in the area of education. Note, however, that the report tries to express a general perspective on metadata rather than a perspective specific to learning and learning objects. Chapter 1 explains why.

1 Learning object --> Resource

By “learning object” we mean something that:

- is relevant in connection with knowledge acquisition
- is represented in some form of multimedia, e.g. a text document, file, image, audio or video media.

Also, just to simplify matters, our focus is only on objects accessible via the Internet.

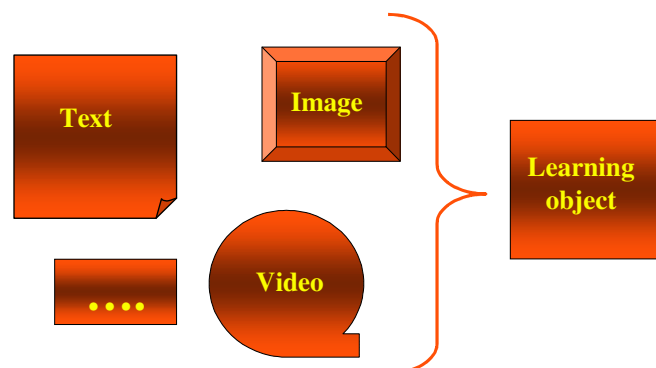


Figure 1

The web comprises an almost unlimited source of objects with a virtually endless variation of content. In metadata contexts, the accepted term for such an object is **resource**.

The resources are there for different purposes and often represent different subject areas. Learning objects are in this context to be viewed as a certain category of resources. From a general perspective, learning objects only represent a small part of the

total amount of resources, but obviously an important part for those working in different branches of education and learning.

In the discussion that follows, no distinction has been made between different types of resources other than their digital representation and accessibility on the Internet. Those who wish may certainly replace 'resource' with 'learning object' or any other type of object for that matter – anything that will help set the discussion within a familiar context for the reader.

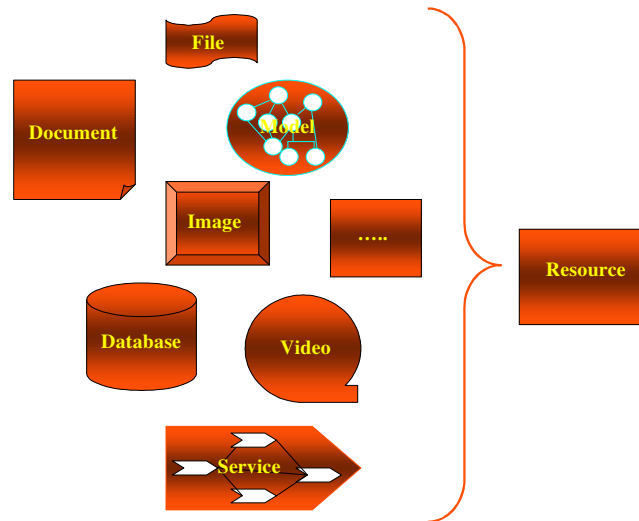


Figure 2

2 Metadata and metadata descriptions

Regardless of the type of resource, it may, for various reasons, be necessary to describe a number of relevant features of that resource. Every reason or purpose gives rise to its own description – its own set of features. Resource descriptions are usually referred to as **metadata descriptions** in which each descriptive element – information or data is called **metadata**. Just to blur things: sometimes metadata and metadata descriptions are used as synonyms. To avoid confusion each feature (metadata element) is in this report called an **attribute**.

In its simplest form, a description comprises a list of attributes.

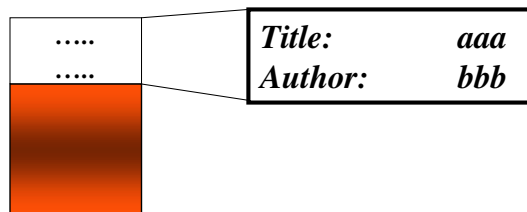


Figure 3

In principle, nothing prevents every unique resource from being described by its own set of metadata.

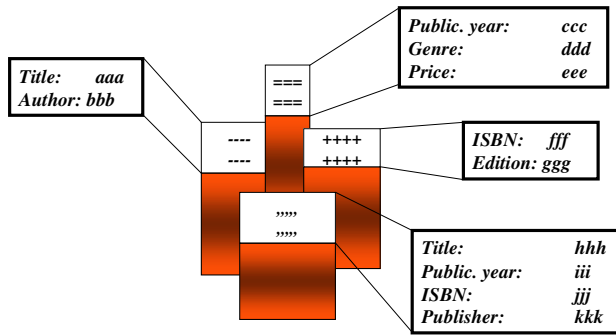


Figure 4

However, it is more common that some type of agreement is met regarding the types of metadata (**template**) that are to be used for a certain purpose, given a certain context.

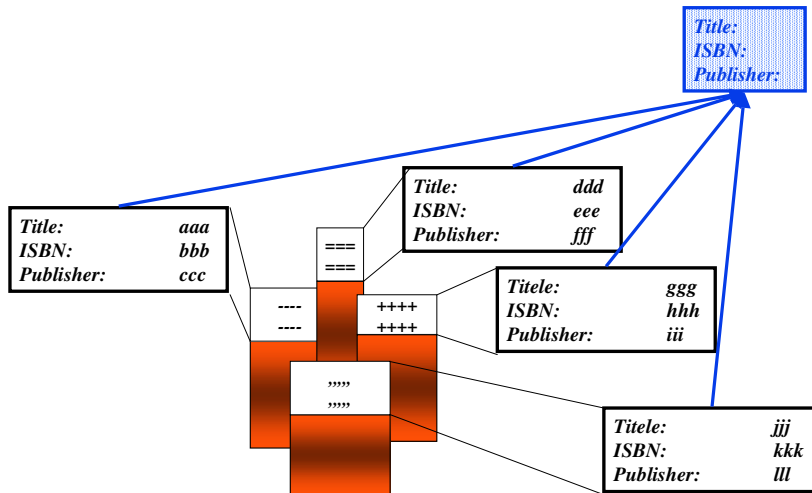


Figure 5

One of the better known examples of such an agreement is that of *Dublin Core* (www.dublincore.org), in which the current version defines fifteen different types of characteristics called elements to be used for description of resources. Dublin Core has its roots in the library environment.

The same resource can very well be an object described from more than one perspective, for more than one purpose. To attach numerous different sets of metadata descriptions (lists of attributes) directly to a single resource quickly becomes unmanageable, especially since new attributes and lists of attributes may be introduced and others be removed at different times during the life of the resource.

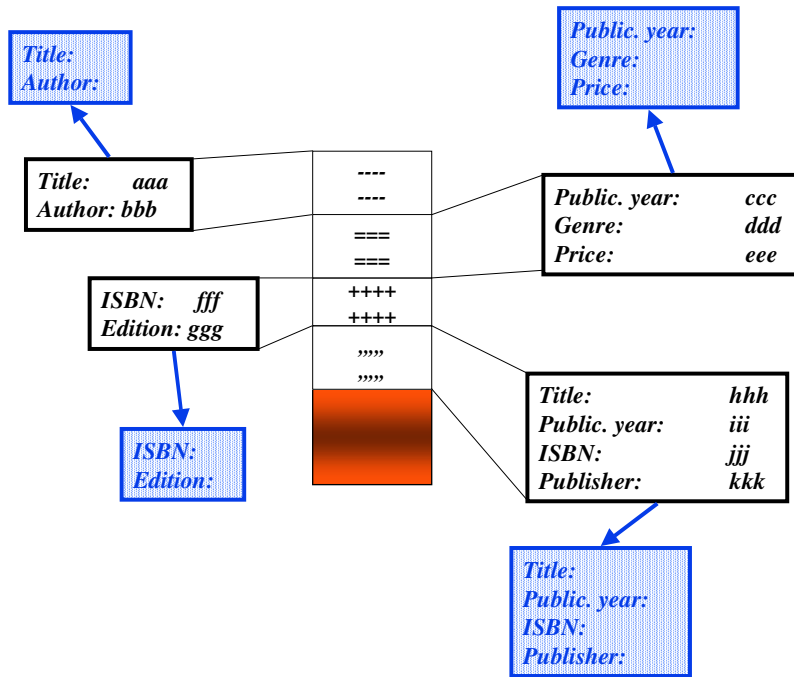


Figure 6

Collecting all attributes into a single metadata description can be one solution, but may be the cause of other negative consequences, as will be shown.

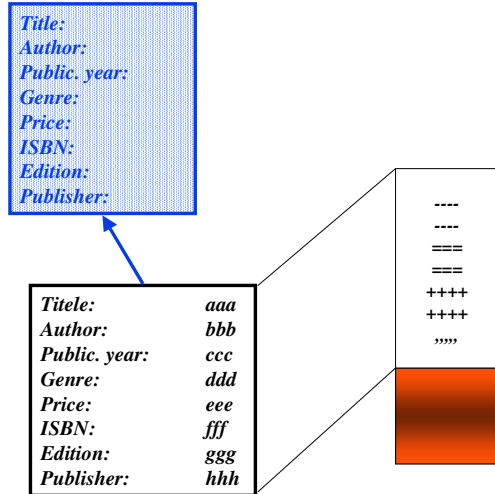


Figure 7

This solution requires the parties to co-ordinate their efforts and to make compromises regarding the appropriate term for each attribute – not to mention the semantics it is meant to represent. The need for a well-defined template suddenly becomes very tangible.

The same co-ordination must then be repeated every time one of the parties wants to make changes. Rules must also be established regarding who, what, when and how a

resource's descriptive information should be introduced and managed. A party is at heart only interested in his/her "own" characteristics. Is this party nevertheless to be responsible for other information if he/she is in charge of the resource in question? And if not, should the resource and its metadata be sent to the other instances to obtain complementary information and or approval? This sounds untenably difficult. Should then, as an alternative, an incomplete set of metadata be accepted? The questions are many.

This leads to a conclusion that every context and purpose will have to take care of its own metadata. The only feasible outcome is to separate resource and metadata .

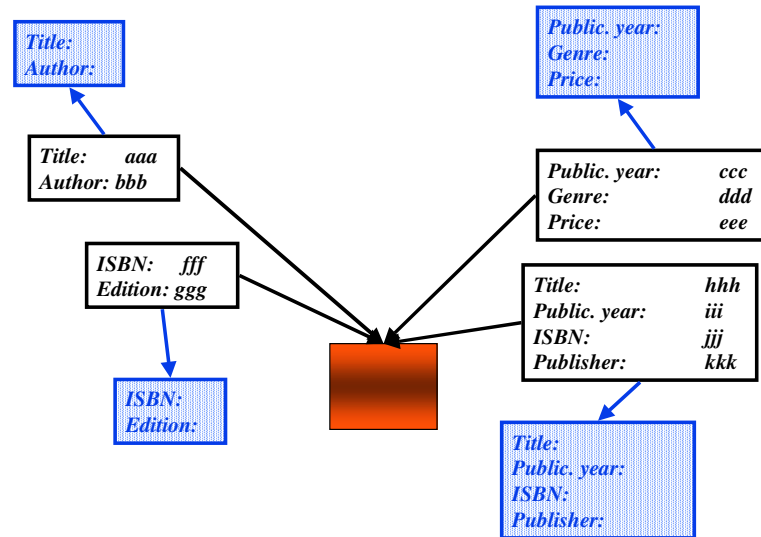


Figure 8

At the same time, we must remember that separation demands a certain discipline in the management of the resource. The metadata was defined on the basis of the characteristics the resource had at a certain point in time. Should the design or content of this resource change such that the changes were to affect the metadata description, then "someone" must also remember to update it so that it correctly reflects reality. For example, the content in a document has been reorganised so that the number of pages has increased from 23 to 27. The attribute *No. of pages* in the metadata description should in following be set to the new value. Perhaps one chooses to change the format that the document was saved in from Word to PDF. *Saved as* must be updated. And so on.

It may sound obvious, but how do we get the person who changed the resource to remember to change the metadata as well? Especially if the metadata is registered somewhere else? It becomes even more problematic if and when the person in charge of metadata is not the same person as the one who made the change – in reality, probably a very common situation. Routines need to be established. Perhaps even rules for managing versions of resources. Where several people are involved, principles for interplay between the parties are also needed.

Returning now to the metadata description, and turning our attention to a certain set of metadata, i.e. a certain unique resource with a given relevant purpose. In some contexts,

a template in the form of a simple list of attributes is sufficient, for example, one of the four alternatives shown in Figure 8. In other contexts, the demands are higher. Perhaps more than one value is valid for a certain characteristic. Perhaps the template content forms a hierarchy. Perhaps ... Perhaps many things. Figure 9 shows an example.

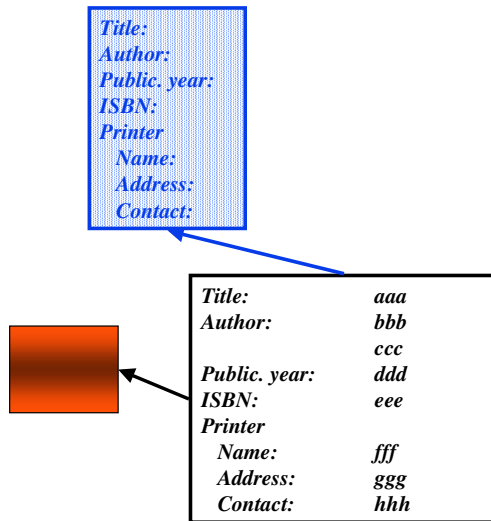


Figure 9

Of course, the hierarchy could be avoided here by removing *Printer* as a “heading” and replacing it by the more explicit *Printer name*, *Printer address* and *Printer contact*. In most cases, however, this would be tricky. Suppose that the resource is a book that has been published in a number of editions and that different printing houses were involved in printing the different editions. Also, suppose that some information about each edition must be manageable through the metadata description. And suppose, in addition, that further editions may be published in the future.

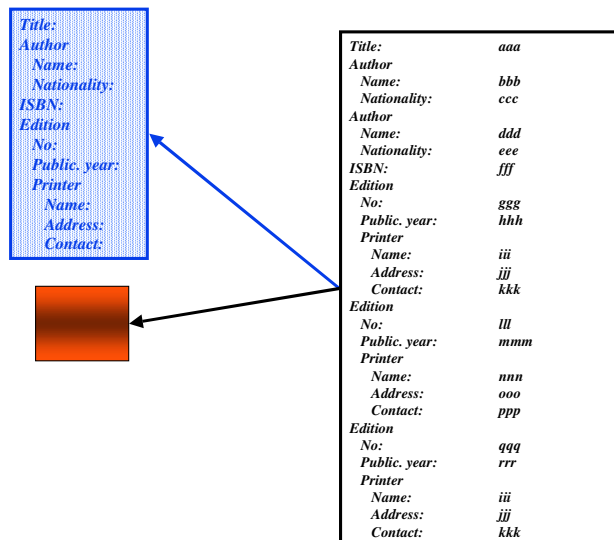


Figure 10

Careful readers will have noticed that *Author* is now also described with the hierarchy *Name* and *Nationality*. They will also have noticed that the same printer (iii) was involved printing both edition one (ggg) and edition three (qqq). The same printer information has to be repeated twice. A rather clumsy approach. There is also a risk for content differences in the two sets. Not to mention the risk that something may be missed at one place when an update is required.

Instead, the two editions should be able to refer to the same printer information. Not an easy task to accomplish in a regular list. The printer information should consequently be specified in its own separate “square” to which the edition information can refer. Why not apply the same approach when it comes to all “grouped” information, i.e. for *Author*, *Edition* and *Printer*? Figure 11 shows what such a breakdown would look like.

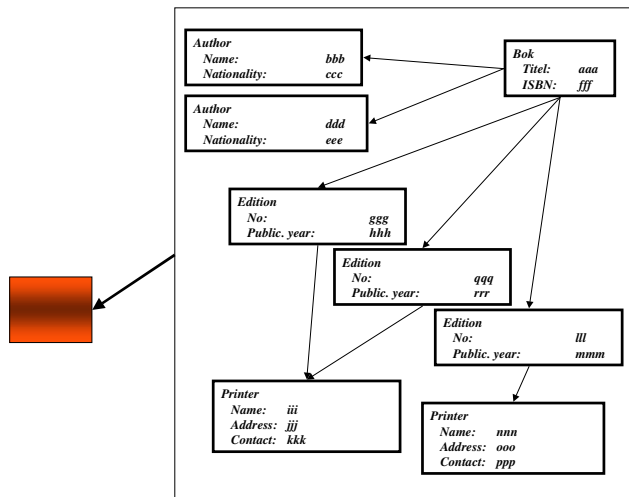


Figure 11

It would be even better if we also labelled the semantics of the arrows – the role of the description the arrow is pointing at – all in an effort to avoid any misunderstanding.

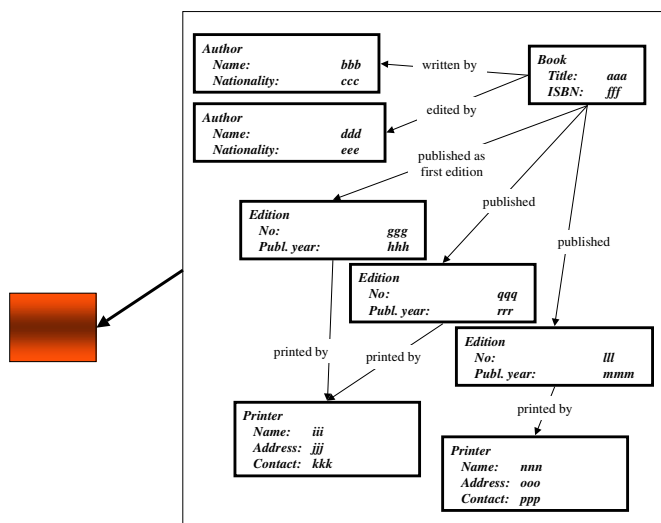


Figure 12

In a more structurally complex example, the need for arrow labels becomes even more tangible.

The template must be adapted to the new structure. In addition, new types of information have to be included to offer a complete specification on conditions to be followed when managing the metadata.

The fact that *Edition* may be repeated any number of times is lacking. It has also been agreed that at most four *Authors* may, and at least one must, be specified. Some books are edited by a special editor. This person is always included also in the *Author* group. *Printer* may be left out where this is not known. Each type of attribute should also have a descriptive element indicating how the values may be expressed, i.e. the data type to be used. All of this information must somehow be included in the template. Figure 13 suggests one way of doing this. Note that this is only one of many ways of expressing the same thing.

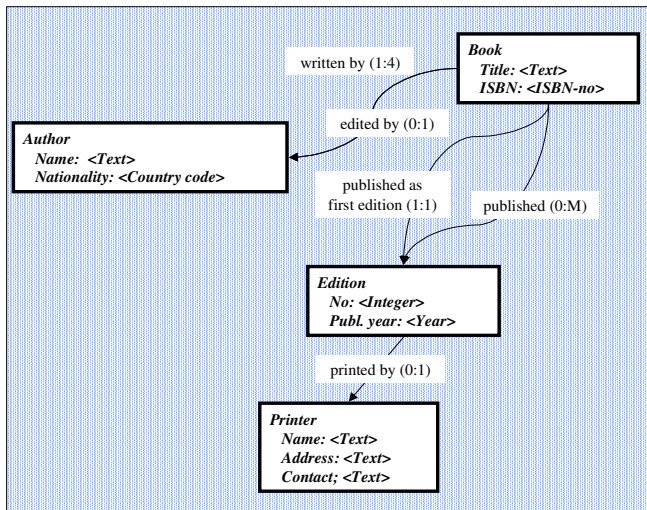


Figure 13

The data type to be used is given inside the brackets. Sometimes a simple <Text> or <Integer> is sufficient. In other contexts, more specific data types may be necessary, e.g. <ISBN no> or <Country code>.

The parentheses following an arrow label states the condition regarding how many copies there can be of the description information the arrow points to. When it comes to “written by (1:4)”, the book must point to at least one, and at most four, sets of author information. In other words – given the actual purpose, whatever that may be – no books can have only an editor (and no author) - in this specific context. *M* stands for the requirement “one or more”.

3 Metadata repositories and conceptual models

From now on, for the sake of clarity, we will use an oval shape with lines and bubbles as a general symbol for metadata.

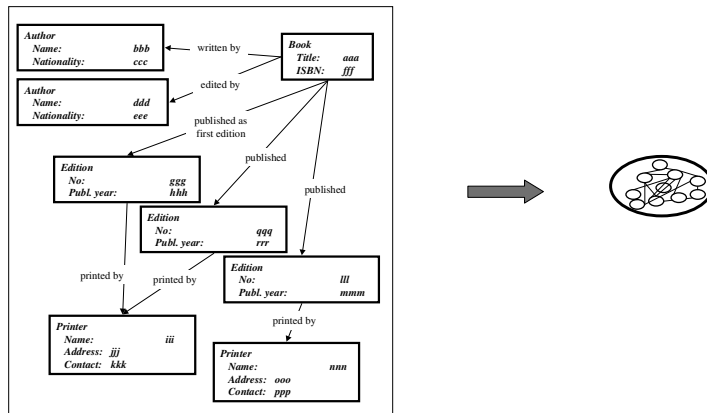


Figure 14

So far, a metadata description has dealt with a single resource. Let us now expand the discussion somewhat. One purpose can very well be to compile metadata descriptions, e.g. for a number of books, in a single database. The reason for this is not so much to “stow the description” in the same place, but rather to enable the creation (when necessary) of cross-references between the resource descriptions. Another reason can be to disconnect a certain part of a description from unnatural direct association with a certain resource (book).

Take *Printer “iii”*, for example. In addition to having printed two of the editions of our original book, it is highly likely that this printing house has also printed a number of other books in various editions. Naturally it would be perfect if all of these editions could refer to the same printer description – a rational and logical solution. Efficient as well, since an address change for the printer only needs to be registered in one place – the printer description – instead of in as many places as there are instances where that printing house has been used for printing.

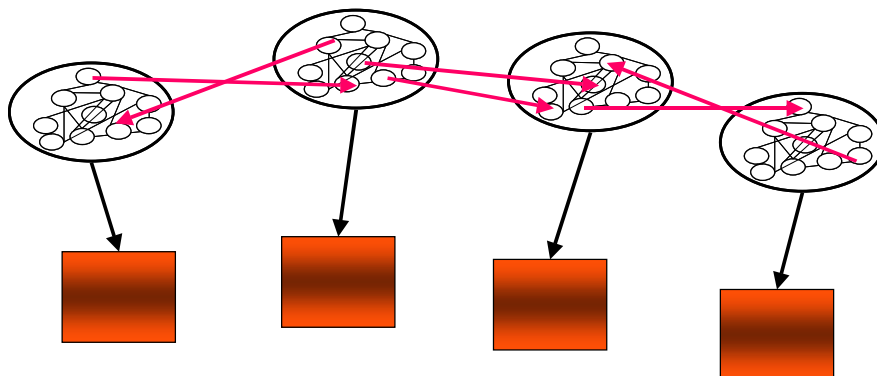


Figure 15

The information becomes increasingly interlinked at the same time as the various parts are placed where they logically belong. Furthermore, if we wish to be able to express relationships between the books (the resources), the need to gather the information in a common **metadata repository** for a certain purpose becomes even more tangible. Take for example a need to describe a suitable order in which to use resources from a learning perspective or to note which learning objects go well together with a certain theme. The template tends to become an increasingly complex compilation of concepts and their interrelations. This fact allures us at now substituting the term “template” with the – in other contexts – more accepted term **conceptual model**.

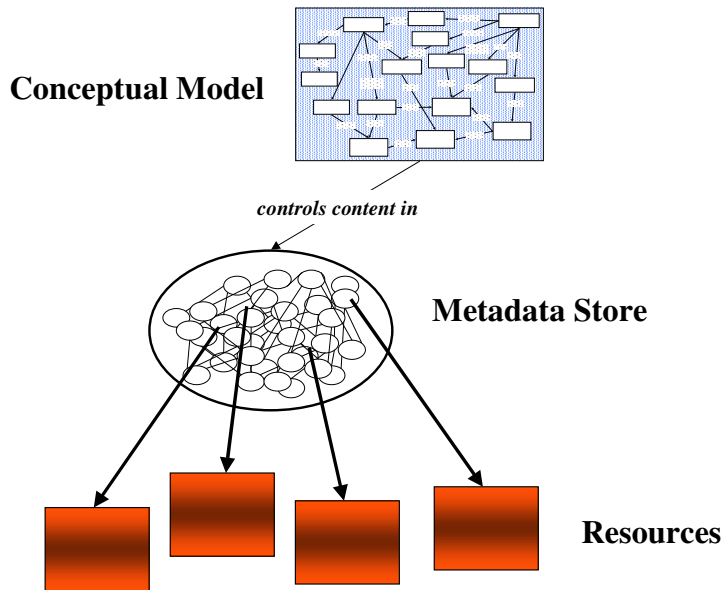


Figure 16

It is time also to explicitly add include into the conceptual model a reference to the resource in question in the form of an arrow from the metadata pointing at the resource. This reference may be a standard URL (Uniform Resource Locator).

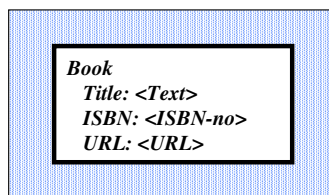


Figure 17

In many cases, the resource can be represented by a certain place in the metadata repository (where, among other references, the URL reference to the resource is probably found). In most cases, however, there may well be structures where a single resource is referred to from several locations in the same metadata repository. For example, there may be a need to describe the different roles the resource plays from different perspectives or in different contexts.

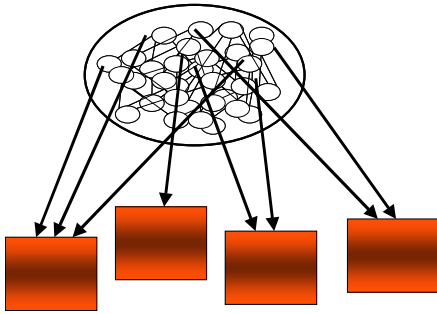


Figure 18

With an increasingly interwoven and complex conceptual model comes the need for more stringent expression of the rules and conditions that apply regarding the content of the metadata repository (indicated by “controls content in” in Figure 16).

We already know from a certain book whether it was written by one or up to four authors. But what happens with a certain author when there are a number of books in the metadata repository? Can the same person be the author of several books? Yes, in all probability. That is, if the sole purpose of the actual metadata repository is not just to present the most read book of every author who has received the Nobel Prize. In order to avoid misunderstandings, the valid restriction from the given context must be introduced into the conceptual model. A second parenthesis, namely “(1:M)” is added to the “written by” arrow. (See Figure 19.) This allows us to indicate, from the particular *Author*’s perspective, that this person can be linked to (1) or more (M) books. On the other hand, the “edited by” arrow shows (0:M). The “0” value is motivated by the fact that there may be authors who don’t appear in the role of editor.

Note that the notation used in the conceptual model is only one of many ways to express this. Every modelling language uses its own version. What is key here is not really how the condition is noted but that it is noted in such a way that everyone involved is able to interpret the actual condition.

From a *Printer*’s standpoint, it must be possible to refer from several editions to a printer via the “printed by” arrow. This is appropriate because this is the way things are in reality, that is, that a single printing house can be used to print whichever editions of whichever books. It can also be worth noting that a decision was made to only include printing houses that have printed at least one edition in the metadata repository. From the printer’s point of view, these two conditions are expressed by adding “(1:M)” to the “printed by” arrow. Other conditions added are also specified in Figure 19.

One more thing, the figure has been made slightly more complex by the addition of *Theme*. Every *Theme* is described by *Name* and *Target group*, and refers to the books that are presumed to be used during the study of that *Theme* (“retrieves knowledge from”).

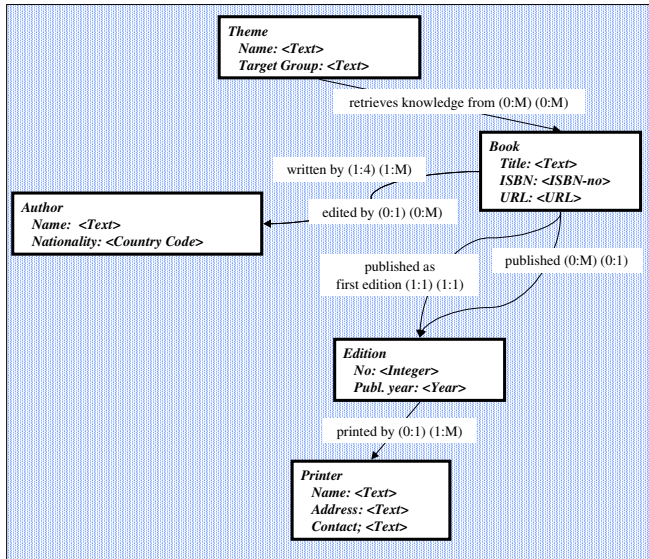


Figure 19

We now find that thematic knowledge is not retrieved from books alone. Resources in the form of *Images* and *Videos* are also of interest as *Educational material* used for the Theme. Regardless of the type of *Educational material*, it must have a *Title* and be accessible via an *URL*. In addition, every specific type of *Educational material* is described by its own set of metadata. For *Book* – this is the ISBN number, for *Image* – the format it is saved in, and for *Video* – it's playing time. Certainly it would be positive if there were a simple notation in the diagram to indicate that *Book*, *Image* and *Video* each is a type of *Educational material*. With that in place each specific attribute may be put in its logical place. Figure 20 suggests one notation that could be used for this.

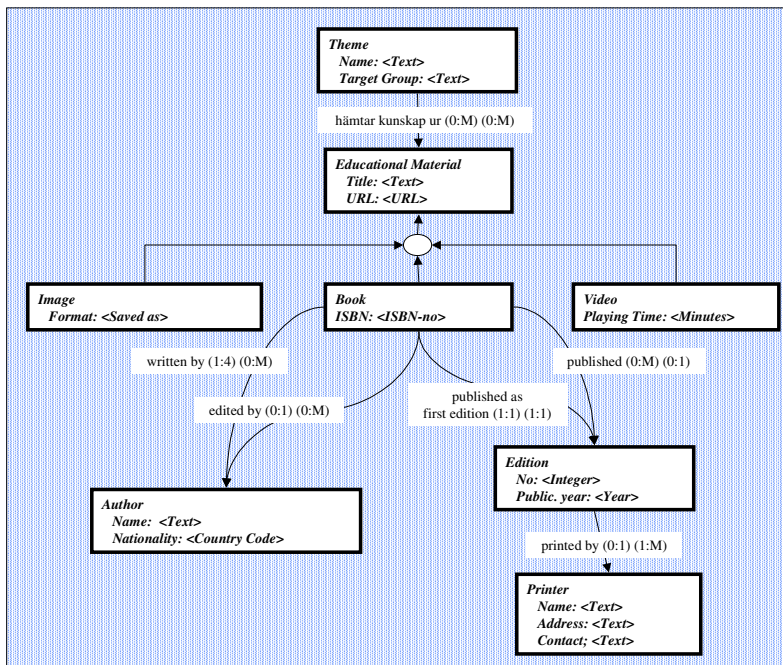


Figure 20

The small circle represents a so called “specialisation”, i.e. that an *Educational material* can more specifically be either a *Book*, *Image* or *Video*. From a *Book* perspective, every *Book* is also a piece of *Educational material*. From which follows that, in addition to the *Book*’s own metadata, *ISBN*, it is also described (as an *Educational material*) with the help of *Title* and *URL*.

The purpose has not been an attempt to turn the reader into a full-fledged modeller – or to confuse matters. Rather the examples hopefully shows a number of possible ways of expressing things to increase the preciseness of the conditions that apply for the metadata that will be permitted to be stored in the metadata repository.

The metadata repository and attached conceptual model in fact very much resemble how a regular database is constructed. This should come as no surprise, since metadata is at its core regular data. It is worth noting that the languages and notation used in database contexts to formulate conceptual models frequently contain significantly more possibilities of expressing the desired conditions and restrictions than those shown above.

Hopefully the discussion has shown that even metadata needs to be specified using some form of general **modelling language** that is able not only to define templates in their simple form but also to define more complex conceptual models according to the above. W3C has specified a simple language for storing metadata and exchanging metadata, namely, *RDF (Resource Description Framework)*.

Note that a number of the conditions expressed in figures 19 and 20 above cannot be specified in RDF, but certainly with other modelling languages. Because these types of conditions will in all probability be of interest in more advanced applications, work is underway in various places to come up with more “expressive” modelling languages. One such project is *DAML+OIL* (*DAML*=*DARPA Agent Mark-up Language*, *OIL*=*Ontology Inference Layer*), a co-operation between two organisations that previously operated separately.

Why the actors in the metadata field have not chosen, at least to begin with, to use one of the existing and proven modelling languages that have been used in database contexts for decades, is a mystery.

4 What a lot of meta!

Some closing examples can hopefully top off all the talk of metadata in this report. A bit bantering perhaps, but with an underlying tone of seriousness, aimed at presenting a conceptual groundwork for the views discussed in Chapter 5.

A common definition of metadata is “data about data” or “information about information”. This seems reasonable considering the examples given above. But let us introduce a new dimension. Suppose that we have a newly released book that is available on the web. The book is reviewed. In addition to information such as title,

author, etc. given in the review, there is also a reference to the book, i.e. its URL. The review is a type of metadata.

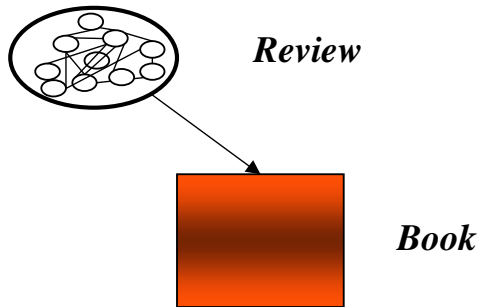


Figure 21

The reviewer places the review on its own web site along with other information about the book, neatly structured in accordance with an appropriate metadata template (conceptual model). The review thus gets its own *URL*, itself becoming a resource.

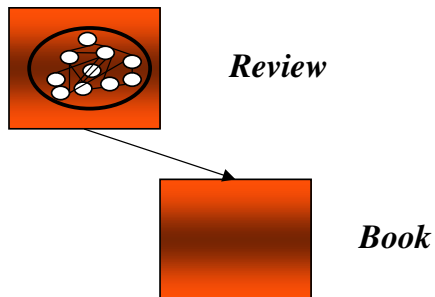


Figure 22

Enter a book lover who reads the review, thinks it is “all wrong” and decides to write his/her own response to the review and post it on the weekly e-magazine “Open Forum” according to the conceptual model used there. The response is in practice a review of the review, as well as a new web resource with its own *URL*.

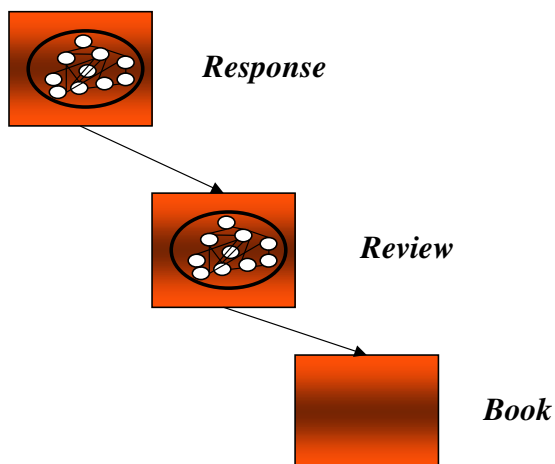


Figure 23

Does this make the response meta-metadata? Well, it is metadata about metadata. But let's wait a bit. A feature writer sees the response and backtracks to the review. Because this person has also read the book, she decides to write a humorous commentary on the book, the review and the response. This commentary is placed on the Culture Club's web site.

So now we have references to data (the book), metadata (the review) and meta-metadata (the response). The commentary itself falls in at the meta-meta-metadata level. Or does it?

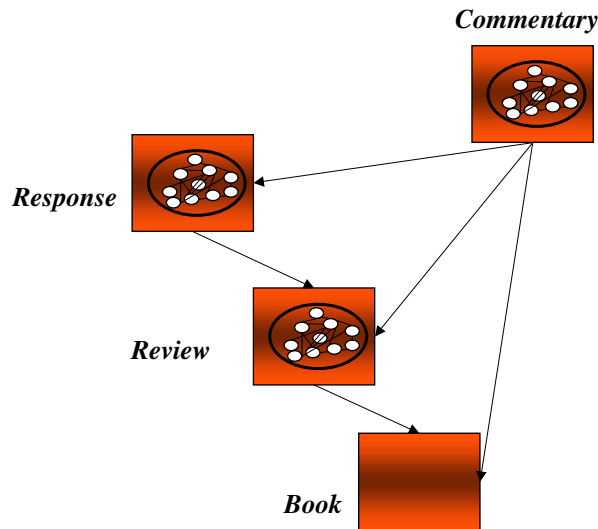


Figure 24

Adding to all this, it turns out that the book is an autobiography that in many respects describes the various exceptional qualities of the author, i.e. the capability or resource the said author possesses. The resource in question can hardly be placed on the Web since it firmly resides in the author's body and soul. There is no doubt, however, that the book is data about this physical resource. Perhaps the book should by itself be seen as somewhat metadata after all, at least if we include as resources even those not necessarily found on the web? Certainly, in an extended sort of way.

And it is an equal stretch to view all of the pictures of the author found in the book as metadata. More reasonable perhaps is to consider a picture as at least data or a resource in those cases where different components of the book are accessible separately. It becomes even more difficult when we find that some of the pictures are actually photographs of *collages* of photographs of the author. Still stretching – it is metadata in the form of a picture (data about data). Or is it? If so, we must, in the name of consistency, add another meta level to the metadata resources already mentioned that have bearing on the book.

Phew! You see how this could escalate into a string of “meta” prefixes. Something must be done here. One approach would be to do a clean up between meta levels. From the book lover's perspective, the review is an article, i.e. an entirely common information resource meaning that the response must be seen as metadata, not meta-metadata. The

book, the review and the response are, in the eye of the feature writer, all data or resources meaning the commentary can be seen as metadata. And so on.

Aha – by starting from a particular perspective in each individual case, we can limit it to a single metadata level. This quickly renders things a bit easier to handle.

It would be far simpler still if all reviews, responses, commentaries, etc. could be described and managed in a single metadata repository. Necessary references are achieved within the framework of the metadata repository, and all data in the store is seen, from a definition standpoint, as metadata, including our reviews, responses and commentaries. Fixed up and ready – easy as that.

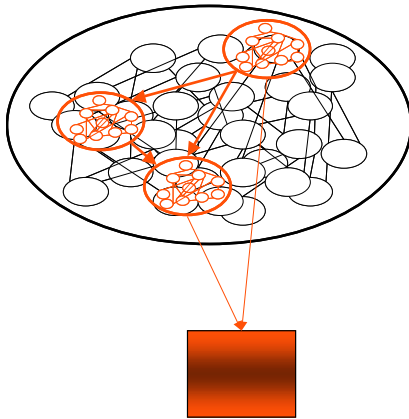


Figure 25

Unfortunately, you don't get anything for free. The disadvantage here is that they are no longer "free" resources in cyberspace, but only accessible and possible to relate via the metadata repository's interface. Probably acceptable for some purposes and unacceptable for others.

A compromise would indeed be to allow the respective review, response and commentary to be described in the metadata repository according to the above, but at the same time also allow them to exist as independent resources with certain vital information somehow enclosed in the body text. A complete hedging it may seem, but one that suddenly allows the same information in several places, with all what that entails.

It does not require great ingenious to come up with other variations on the same theme.

Could this perhaps be something the magic "Semantic Web" will have to put up with, or can the discussion be refined even more? In any case, speculations about what "meta" really stands for, and why it is needed, remain. If it is needed at all, that is. On to the next chapter.

5 What do we really mean by “meta”?

We have discussed (web) resources and the need to describe them. In other contexts, there may be a need to describe goods in a stockroom system, employees in a personnel system, invoices in a financial system, or whatever. The world is full of database applications for an infinite number of purposes. Is a web resource, from a descriptive standpoint, really so magically unlike goods or employees when it comes to the principles used to describe them?

Of course not. They are all objects that need to be described. The fact that one is electronically available in some multimedia form on the web, another on a shelf in a warehouse, and yet another behind a desk in an office, is hardly a critical difference in this context. (Possibly in some cases, such as the fact that an employee can move around and has behaviour. But because the personnel system represents static rather than dynamic information, even this slight difference can be ignored here.)

In all of the cases, there is an interest in describing static information - or more properly a camera view - about the phenomenon of interest. To this end, database management systems (DBMS) have been at our disposal for several decades. Every data management application is in most cases based on an implementation-independent data model (conceptual model) that is later adapted to the requirements under which the actual DBMS in question works (e.g. according to the relational model – SQL). The person’s physical location may change, while the description of this person, from a certain perspective, is found in a database.

Note that the person at the desk may well be an object for different interests as well, i.e. the need to describe the same subject for different purposes. Each purpose or role can give rise to its own database. This can be a personnel system (in which education, qualifications, etc. are described), payroll system (in which past and present salaries are documented), project management system (in which current work tasks, responsibility, etc. are kept up-to-date), and the list goes on.

The implementation-independent conceptual model is expressed in a modelling language that hopefully has been chosen in consideration of the strength of expression desired. This can be anything from a simple binary modelling language (similar to that shown in Figure 19, above) to the considerably more complex UML (Unified Modelling Language). RDF is, incidentally, for the most part, nothing else than a simple binary modelling language.

The important thing to understand is that every conceptual model is specified with regard to:

- a certain closed piece of “reality”, the information of which is decided to be of interest to manage.
- a certain purpose. (A structural drawing probably doesn’t describe a bridge in the same way as a tourist brochure.)
- the actual observers, i.e. the experiences, values and personalities that characterise the intended users of the information and how they relate to both “reality” and the purpose.

Following the points above, no distinction is needed between the formulation of conceptual models and data models for data management.

What is possibly new to the metadata world, is that we are permitting the use of new types of DBMS that are able to take the specified conceptual model and apply it directly as a schema for storing information. In conventional contexts, SQL (a relational database) is commonly used. Implementation thus requires translation of the implementation-independent conceptual model to a schema expressed in the relational model's language. Difficult and inflexible, in addition to the risk for incorrect translations. Here, the metadata perspective paves the way for new attitudes and implementation patterns. This could just as easily be achieved without bringing "meta" into it.

6 In closing

The report has introduced a number of angles on metadata and metadata management. It is perfectly clear that, from a metadata perspective, data management and conceptual modelling appears to be new, at the same time as it is fairly conventional from a data perspective. All current trends indicate that a description of the resources that can be found and managed over the Internet will play a key role in the growth of the next web generation, one that presently goes under the name of the "Semantic Web". To this end established and generally accepted conceptual modelling languages will be needed by which presumably an almost endless number of conceptual models may be created.

Modelling languages already exist. Conceptual models for web resources are also being generated at an increasingly rapid rate. Many are still of a relatively simple nature (attribute lists). Based on successively gained experience, demands become more precise. More complex models will gradually emerge. Development would take considerably larger strides and move forward much faster if the advocates of metadata would navigate towards the solid knowledge of modelling that exist in the area of conventional data and connected conceptual modelling.

What would still seem to be a relatively open field is how metadata can best be managed with respect to a well-formulated perspective of usage and of benefit and how different user roles may need to be supported.

An aggravating circumstance often pointed out in connection with web resources has to do with incentive. Those who are best able to formulate a resource description, often the person who created the resource, seldom benefits from the existence of the description information. Perhaps too much work has gone into how web resources can and should be described without looking at the description from a purpose and use perspective. Presumably there lurk significantly harder "nuts to crack" around the corner, not least when we shift our focus from metadata as such to how metadata can be managed in the environments currently taking shape in trends such as "Web Services" and the "Semantic Web". The digital rights perspective is another aspect that certainly will influence the future of resource management on the web. And so on.

It is probably not too risky to venture a claim that the field of metadata stands on the brink of exciting, perhaps dizzying, development trends in both the near future and a more long-term perspective.