Identifying the identifiers

Douglas Campbell
National Library of New Zealand
douglas.campbell@natlib.govt.nz

Abstract

Identifying things and assigning identifiers to them is a fundamental part of working in the digital realm. We need to identify resources, concepts, agents, relationships, mappings, properties, namespaces, schemas, profiles, etc. But often we find assigning identifiers to these things can get confusing due to the range and subtleties of meaning, yet we manage to successfully identify and label things in our, just as complex, everyday lives. It may help to deconstruct these identification processes we perform intuitively so we can reconstruct a sensible approach to designing our identifiers.

This paper looks at how we identify things by comparing the sameness of their characteristics, how we associate symbols with things to simplify identifying them, and concludes there are six aspects that make up an identifier: a thing, a symbol, an association, a context, an agent, and a remembrance. It then considers some of the qualities of identifiers in more detail: scope, uniqueness, granularity, intelligence, actionability, persistence, extensibility, and context. It finally proposes a simple checklist for designing identifiers.

Keywords: identifier; symbol; description; framework; model; persistent; unique; actionable; intelligent; extensible; context.

1. Introduction

As a national library and archival institution, the National Library of New Zealand has many different types of objects and concepts to identify and manage. We have designed an identifier to manage our digital collection objects internally (Kebbel & Campbell, 2004), but we have struggled with determining what the next steps should be.

However, grappling with identifiers is a common issue (NISO, 2006, March & July); this may be due to us overlooking aspects of identifiers we do not realise we know intuitively. It may help to deconstruct what we do when identifying things in our lives to give us a model to frame our designs for identifiers and identifier systems.

We all use identifiers intuitively when communicating and interacting in our everyday life, e.g. "please pass the salt" or "my ticket is for seat D3". But computer systems are not as intuitive as humans, so we need to be more deliberate and precise in how we assign and use identifiers.

2. Identifying to Communicate Sameness

At some point when communicating we will want to refer to things (both concrete and abstract). We then need to find a way to codify the thing we are referring to into our message so that when the receiver decodes it they will be referring to the same thing that we had intended. (Note that in some cases the transmitter and receiver are the same, e.g. labeling or identifying things within our own thoughts.)

So, the purpose of identifying is to preserve sameness. Indeed 'identify' is derived from the Latin words 'idem' (the same) and 'facere' (to make).

To identify a thing we need to differentiate it from other things. To differentiate things we need to compare the sameness (or not) of their characteristics. To compare characteristics we first need to define the characteristics, i.e. build a description or 'metadata' (even if just in our minds).

To build a description we might:

- Record observable characteristics (known as 'descriptive cataloguing' in the library community), e.g. size or location
- Interpret other existing characteristics (known as 'subjective cataloguing' in the library community), e.g. the kind of smell or the subject concepts expressed
- Assign new characteristics, e.g. name, title, logo, or a unique sequence of characters (i.e. a 'string').

For convenience, we can use parts, or all, of these descriptions of characteristics as surrogates (substitutes) when referring to and discussing things.

Then, instead of identifying by comparing the sameness of multiple characteristics, we can pre-assign a convenience identifier consisting of one, or more, characteristics taken from our description (e.g. "The tall, yellow one on the end"). We can then differentiate by comparing the sameness of the identifiers (without having to go near the things themselves).

However, sameness is not absolute as it depends on the context, so a thing will have different identifiers for different contexts. This means an identifier is valid only in certain contexts (Paskin, 2003). For example, in the context of 'type of vessel' a wine glass and mug may be considered the same (i.e. they are a 'cup'), whereas in the context of 'type of cup' they are different. We would assign different identifiers appropriately within each context, drawing on different characteristics defined in our descriptions.

Many contexts are part of wider contexts (e.g. street, town, country), so often unambiguous identification requires joining multiple identifiers together.

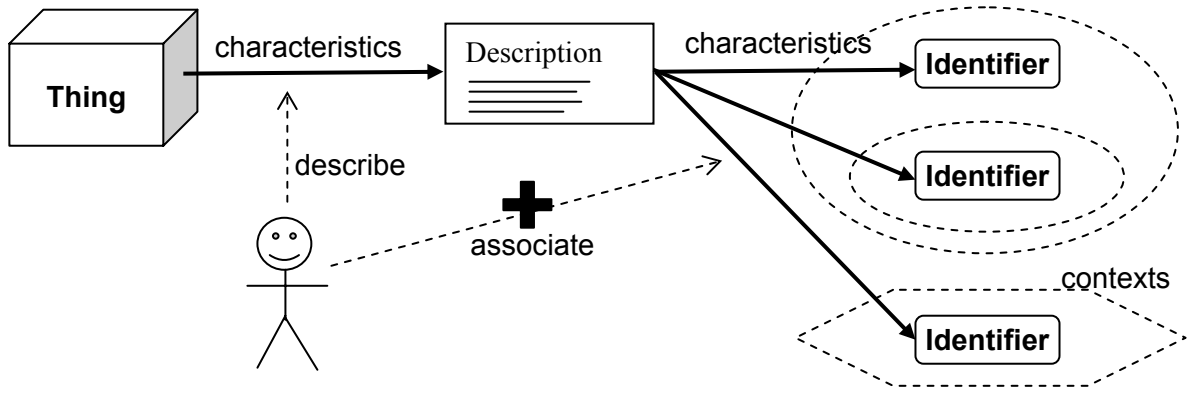

FIG 1. Describing characteristics and using them in identifiers.

3. Defining 'Identifier'

Kunze (2003) provides a good starting point with his uncluttered definition for identifier that focuses on the action rather than the function: "an association between a string and a thing." However this omits the motivation for the association, which might be added and generalised as:

> *Identifier: A stated association between a symbol and a thing; that the symbol may be used to unambiguously refer to the thing within a given context.*

Here, a thing is any entity, idea, action, resource, object, etc. and symbol is any mark, token, sensory stimulus, character string, etc. (The unambiguous nature and context of the association is explored in the next section.)

But is this all there is to an identifier? Since identifying is part of communicating, there may be communication theory models that help reveal other useful aspects of identifiers. Semiotics may be a relevant area of study – it is concerned with how we communicate using signs and symbols. It indicates we use symbols in our communications that actually have no intrinsic meaning, yet they do manage to convey meaning and represent things because we provide that meaning around

them. This is commonly referred to as the 'semiotic triangle' (Pierce, 1931-1938; Ogden & Richards, 1923; and Saussure, 1974). Identifiers hold a similar connecting role to symbols so might overlay reasonably over the semiotic triangle (see also FIG. 2):

- Symbol (or Representamen or Sign vehicle or Signifier) – the identifier symbol
- Concept (or Interpretant or Sense or Signified) – the association and context conceived by the stator
- Object (or Referent) – the thing identified.

We can see it is the thought originating in someone's mind that creates the (implied) relationship between the Symbol and the Object. If that thought is lost, so is the relationship.

What we can learn from this exercise is that an identifier will only exist as long as anyone remembers the declaration of association. Persistence of identifiers is not so much about remembering the identifier itself, but what it is associated with.

We can also conclude that identifiers are a manifestation of the act of identifying. They are separate from descriptions; while identifiers are indeed often descriptive, their primary purpose is for differentiation, not description. The identification action elevates selected, existing descriptive characteristics to a higher role. Pierce (1931-1958) declares: "nothing is a sign unless it is interpreted as a sign", so while any characteristic might be used to identify a thing, it is not until someone actually conceives or states it (the association) that the characteristic becomes an identifier.

Thus we can deconstruct identifiers into six aspects:

- A Thing
- A Symbol (built from characteristics defined in a description of the thing)
- An Association – between the symbol and the thing
- A Context – that the association occurs within
- An Agent – that states the association and context
- A Remembrance (memory or record) – of the association and context, and ideally also who the agent was (often in a record kept by the agent or a third party, though it may occur through a mechanism such as embedding the identifier back into the thing).

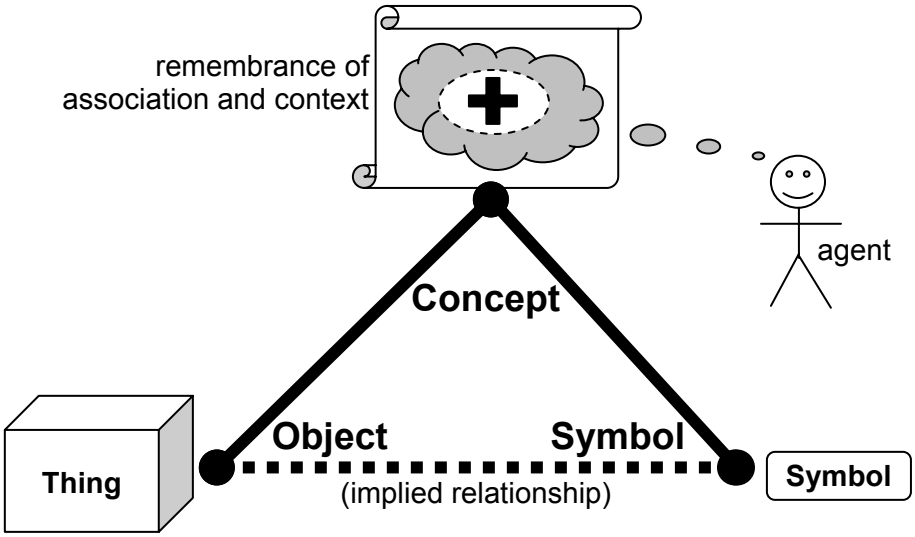

FIG 2. Identifier aspects in the semiotic triangle.

The actions and mechanisms we use to support identifiers, though not a part of the identifiers themselves, also have an impact on their design, so are useful to consider in the following discussions:

Identifier system: Policies, processes, and/or mechanisms for assigning, managing, and using identifiers.

An identifier system might be as simple as 'pen and paper' or a convention you use for naming things.

4. Identifier Qualities

This section takes a closer look at the qualities of identifiers themselves to help guide the design of identifiers. Each is limited to a summary and some points of interest discussed in terms of the models described above. While the examples are taken mostly from the information resource management space, the discussions are still intended to apply at a generic level (of identifying things).

A number of questions are noted around scope, uniqueness, and granularity; in practice these are best considered simultaneously rather than in isolation.

4.1. Scope

It is important to be clear what is being described so any identifier drawing on characteristics from that description identifies the intended thing.

A common moment of confusion occurs when creating descriptions of things – "what exactly is being described?" What looks like a single thing can be described from many different points of view, often with very subtle differences between them. Recognising these differences can be tricky. Each point of view might better be considered as a separate thing. For example, when preparing the description of a newspaper article, the scope might be any of: the print article, the idea behind the article, the physical newspaper (as a whole entity), a digitised page, the PDF document containing the digitised article, the archived web page(s) for the online version, the database record in an indexing and abstracting database for the article, the article syndicated in another newspaper, or the blog entry by the journalist, etc. (see Appendix A for other examples).

The Dublin Core Metadata Initiative (DCMI)'s one-to-one principle (Hillman, 2005) is relevant here. It essentially states that for each thing you need to describe, make a separate description. Note that some descriptive formats (e.g. MARC), for convenience, collapse descriptions at multiple levels into a single description (e.g. subject concepts, physical format, and http URIs (i.e. URLs) of multiple online versions are combined in one record). Others keep them separate (e.g. EAD, RDF). Both types are useful in different situations. It is a question of awareness; we need to consider the source and level of scope for each component of the description. Implementations of Dublin Core based descriptions have not always followed the one-to-one principle, but the DCMI is moving towards clarity of implementation in its recently developed Abstract Model (Powell, 2005).

The library community has developed a framework for considering the differences in scope – Functional Requirements for Bibliographic Records (FRBR) (IFLA, 1997). It separates a resource (e.g. document) into four entities:

- Work – the intellectual creation (e.g. an unwritten story)
- Expression – the act of expressing the idea into a form (e.g. writing, filming)
- Manifestation – the result of an expression (e.g. a book)
- Item – a particular physical instance of the manifestation (i.e. the copy you have in your hand).

FRBR is a useful example we could use as a basis for frameworks in other areas.

Once we have the description scope clear we can feel more confident about assigning identifiers since the body of the identifier (e.g. a string) is drawn from characteristics in our descriptions. So, to ensure the identifier matches what we are identifying, the process ideally should be: identify the scope of the described thing, describe it (including assigning characteristics, e.g. identifier strings), and then choose an identifier(s) based on that description.

4.2. Uniqueness

Our aim in identifying things is so we can refer to them without ambiguity (i.e. it can be differentiated) in our communications, but this is not always possible.

A single identifier could potentially represent multiple things (e.g. the identifier string 'John' could be used to represent anyone in the world called John, Jonathon, etc). Each individual thing with an identifier shared by other things within a context cannot be differentiated uniquely using just that identifier (e.g. we cannot tell which person you mean when you say 'John' in a room full of Johns, or which book you mean when you say 'the book about fish' in a library).

Sometimes the set of things represented by the same identifier within a context has a size of *one,* so we may mistakenly believe there to be a one-to-one relationship between the identifier and that thing (e.g. there is only one 'John' in the room, so when I say 'John' I mean only that particular person, and can get away with that being considered valid). But if the set size increases (e.g. another John enters the room) it feels like it has changed to something different (e.g. 'John' now represents a group of people, not just one person), except this is actually its natural state.

We can make associations unambiguous by limiting these sets to a size of one, resulting in the one-to-one relationship being (correctly) valid. We do this by adding a constraint that each thing in a context must have a 'unique identifier':

- A thing has only one identifier, and
- An identifier only relates to one thing (Coyle, 2006).

Then, once an identifier is assigned, it will always be associated with the same thing as no other thing will be allowed to have the same identifier.

To make these multi-thing group identifiers unique within a certain context, separate identifiers are required. This might be achieved either by extending the group identifier to become unique within the existing context (e.g. adding birth date to name), or by creating a new identifier in a new context (e.g. assigning a unique number to each person).

The latter option (assigning a new identifier) makes discovery take longer. When a group has been discovered using an identifier system, further knowledge/interaction is required to identify the individual members of the group – to determine what new identifier system to use, and then query it. For example, a call centre may know of multiple people at the same phone number; it identifies incoming callers first by the caller id (phone number) then has to switch to asking the name of the person calling. However the additional steps may not be such an issue if they were anticipated and so built into the identifier system. For example, when requesting a web page via an http URI (i.e. URL) it is common for your web browser to perform a 'content negotiation' step with the server to determine which representation will actually be returned (e.g. HTML, PDF, English, French).

In the physical world, location is often used as an identifier to guarantee uniqueness – there can only be one thing in any given position (or each being usually only occurs once) globally. This handy idea is replicated to some extent in computer systems (e.g. file system folder/filename or database record number), though this uniqueness may be limited to just that particular computer system.

As our interactions become more global, there is often a desire to ensure our identifiers are unique globally. This means we can share our identifier with anyone and be confident that they will still refer to the same thing we are. Ensuring uniqueness of identifiers within our own local context is feasible but is more difficult in a global context. However, we can use this as an

opportunity. If the manager of each local context is assigned a globally unique identifier, we can wrap it around the locally unique identifier to get a globally unique identifier, as seen in table 1.

TABLE 1. Wrapping local identifiers to make them globally unique.

Thing	Naming Authority Identifier	Authority's Local Identifier
Phone number	Country prefix (+64)	Area code and phone number
http URI	Server domain name (example.org)	Path on server
ISBN	Country (first 1-5 digits)	Publisher and item digits

4.3. Granularity

We have seen that an identifier may represent a group of things—the big question in scope and uniqueness is how deep recursively should we go in breaking groups into separately identified things?

The answer is more or less self-defining: if you have a need to identify it, then you need to (be able to) identify it! Though unfortunately this still leaves us with the same question!

It may help to choose a methodology for how to define our needs. This could start with identifying who will use the identifiers and how, and may include considering potential future uses too, for example, in China rare characters are being used for children's names which become a problem when those children eventually apply for a driver's licence as those characters do not exist in the database systems (Xinhuanet, 2004). Another approach is to use, or adapt, existing ontological frameworks such as FRBR or <indecs> (Rust, 2000).

Many identifiers are not intended to be standalone (e.g. street number in an address) so they are best interpreted combined with identifiers from other contexts. As discussed above, we can uniquely identify the thing by either joining the identifiers together (e.g. number-street-town-country, version-songtitle-date-artist, or an XPath to an XML element), or by assigning new identifiers directly to each permutation of the possible combinations (e.g. a unique number for every address in a country).

In practice, it is the capabilities of the system that the identifier will be used within that often determine at what level of granularity identifiers will be assigned. Some systems will not allow compound identifiers so a separate set of identifiers must be created and mapped to the combinations.

4.4. Intelligence

Due to the body of identifiers (e.g. the identifier string) being drawn from the characteristics described, there is an obvious attraction towards making identifiers descriptive themselves, e.g. "nytimes_22may2004". These 'intelligent' (or 'semantic' or 'transparent') identifiers can then play an additional role of 'description'.

The advantage of having intelligent identifiers is that remembrance is encoded directly into the identifier (if it is sufficiently descriptive). Arbitrary 'dumb' (or 'opaque') identifiers rely on external descriptions to remember the association. Intelligent identifiers are also easier for humans to deal with than dumb identifiers (which may appear to just be random characters).

The disadvantage of intelligent identifiers is that it creates an expectation for how the identifiers will behave (e.g. that the association is predictable somehow). This exposes their weakness; they are based on your worldview at the time of assigning and you cannot anticipate how this worldview might change in the future which may affect how they behave. It is worth looking at some examples:

- A thing's title is often used as an identifier but the title may be meaningless (e.g. a pun) and is prone to change over its lifetime (e.g. a person's name may become abbreviated or

changed through marriage, movie and TV show names are often changed when syndicated to different countries).

- The date (of production) may seem sufficiently stable to use safely, however the BBC found this caused a problem for radio episodes that are rebroadcast – there was an expectation the episode would be located on their website under the recent broadcast date instead of the original production date, eventually they decided to move to dumb strings to identify each episode (Coates, 2004).
- Kunze (2003) reminds us that the meaning of words can change over time (e.g. 'gay').
- Identifiers may be re-purposed over time, e.g. an email address originally identified an electronic mail box, but email addresses are often used as website logins, meaning they are now also used to identify a person.

Using location as an identifier is quite common, often because no other, more considered, identifiers have been assigned, but it comes with risks. Persistence can become a problem if the things are reorganized (e.g. books reshelved, country political borders shift, Web page '404 not found' errors), or if the location is defined using system-specific methods that subsequently become obsolete (e.g. proprietary shelving terminology or a web page file named 'default.asp').

Location is a characteristic that may be more appropriate to use at a lower granularity level such as when accessing a particular instance of a thing.

Another risk of using location as an identifier is dilution. When copies of a thing are available from multiple locations, the thing is effectively assigned multiple identifiers (e.g. building locations or http URIs) instead of one higher-level identifier, so each instance may be mistakenly identified as a completely different thing when they are actually all the same (at that higher level) (Weibel, 2007).

It is worth clarifying that URIs (Universal Resource Identifiers) beginning with 'http:' are not necessarily location-based URLs (Universal Resource Locators) (W3C/IETF URI Planning Interest Group, 2001); in fact, the 'L' stands for locator (not location) which might be considered synonymous with actioning (see the next section). Older-style URLs were purely location-based whereas many are now assigned with more care; effectively they are identifiers that just happen to start with 'http:' (Fielding, 2002). So the risk of their location changing may not apply. However, the intelligence risks discussed above would still apply. Indeed, the W3C warns you should not rely on metadata embedded in URIs (W3C Technical Architecture Group, 2007).

4.5. Actionability

Identifiers provide a way to refer to and discuss things at an abstract level. At some point we will likely want to access/retrieve/experience the thing the identifier refers to. This involves invoking the remembrance of the association between the identifier and the object.

We may consider an identifier to be 'live' if there is a remembrance of what it is associated with, i.e. it is possible to access the thing or its description somehow, e.g. it may require manually searching for paper records, but it is possible.

An identifier is 'actionable' (or 'resolvable' or 'de-referenceable') if it can be used in an automated mechanism to access the identified object, or a representation of it, e.g. a car key or an http URI (i.e. URL).

Actionability also has its own context as there may be multiple mechanisms availablealthough the mechanisms may not be consistent (e.g. the URI for a controlled vocabulary term may return a description of the term whereas the URI for a Flickr tag returns the content it is associated with) and mechanisms come and go over time (e.g. ISBNs are not currently actionable in Web browsers by default but it is conceivable they may become so in the future).

Some actionable identifiers use the location as the identifier (e.g. shelf location); this is what makes them easily actionable (the location is already known). But this also makes them intelligent identifiers and so come with the risks discussed in the previous section.

4.6. Persistence

An identifier is useful only if someone actually has a need for it. It is useable only if someone remembers what thing is associated with it.

So the two questions here are: how long does an identifier need to live and how do we keep it alive that long? A network packet (and its identifier) only needs to live for a few seconds (long enough to make it across the network to its destination), a web page may only be considered topical for a few days or months whereas collecting institutions such as libraries and archives have no end date for how long they need to maintain and identify things (Coyle, 2006).

Technology can help implement persistence but ultimately it comes down to the commitment of people and organizations (Shafer et al., 1996). The duration of persistence required is less of an issue than actually getting someone to take the time to consider how persistent the identifiers they create need to be plus how to make that persistence happen.

Ensuring persistence is primarily about establishing policies for how to handle changes in the environment, for example:

- When an identifier is retired (including ensuring it is not re-used to identify another thing)
- When the thing itself changes, e.g. a newspaper changes its name
- When the identifier system being used becomes obsolete, e.g. the HTTP Internet protocol is superseded
- When the custodian of the identifier changes.

These policies may include the degree of mutability, i.e. the acceptability of changing the association to different things over time (in the interests of continuity). For example, when a newspaper changes its name we may prefer the existing identifier to be associated with the new name, or alternatively we may prefer to create a new identifier leaving the previous identifier only associated with the previous name (though ideally with a note indicating the new identifier).

We have seen that it is the association aspect of the identifier that needs to be remembered. This takes effort and resources so we should look for ways to minimise the effort needed so it is more likely to happen. For example, follow standards (safety in numbers), embed the identifier back into the thing (cannot be lost), and actually using the identifier (it is harder to justify supporting something that is not used).

Previously we (Kebbell & Campbell, 2004) suggested two levels of granularity for persistent identifiers for our digital collection objects: Persistent Identifiers (PIDs) and Persistent Locators (PLs). The aim was to differentiate between the collection objects we are identifying and locators for current representations of them (which will change over the years as file formats come and go). We will make those locator identifiers persistent (PLs) for the natural lifetime of the representation but no longer (they might better be re-named Semi-Persistent Locators).

Interestingly, the term 'permalink' in the blogging community might be a realisation of this PL concept – their purpose is more to facilitate linking than to identify, and while they are more permanent than many typical http URIs, they may be location-based so can still 'break', e.g. if the blogger moves their blog entries to a new service provider.

4.7. Extensibility

Identifier persistence looks at individual identifiers, but we also need to consider the persistence of identifier systems (i.e. policies for designing identifiers). Some identifier systems will have unexpected demands placed on them, for example, from becoming popular (e.g. 4 byte IP addresses), from the identifiers being used in ways they were not originally designed for (e.g. email as login username), or because the environment changes.

We can attempt to future-proof identifier systems by building in extensibility, i.e. the capability to be adapted. This might include such things as keeping the identifier form as generic

as possible, providing 'hooks' where community-defined components can be added, considering scalability, following international standards, and being application independent.

4.8. Context

Identifiers lose some, or all, of their value if their intended context is not known. The remembrance must include both the association *and* the context, so ideally identifiers will travel with details of their context attached. This might be alongside (e.g. "the journal has ISSN 1234-5678") or combined within the identifier (e.g. "urn:issn:1234-5678"). Even so, there are likely to be even wider contexts that are not declared, e.g. to understand "urn:issn:1234-5678" you need to comprehend that this is a URI. To process a context requires some pre-knowledge.

We should be aware that the context is often omitted when communicating identifiers with the (often unreasonable) expectation that the receiver will be able to infer it.

We have seen that a thing will most likely have many identifiers in multiple contexts, for example, you as a person are probably identified differently by each organization you interact with (e.g. identify by phone number, credit card number, etc.).

Having multiple identifiers for a thing is extremely common and should not be considered undesirable as sameness is different for different communities. However, it might be considered undesirable that similar contexts exist separately. This results in separate identifiers that have virtually the same meaning so translations/mappings are needed between them otherwise people think they are discussing different things when in fact they are the same. In these cases, communication would be eased if the different communities agreed to combine their identifiers and contexts.

5. Framework for Designing Identifiers

Unfortunately the discussion above probably identifies more issues than solutions. Here is an attempt at a simple checklist that pulls together the various qualities of identifiers to consider:

1. Audience – consider how the identifiers are intended to be used and potential downstream uses
2. Scope – determine the thing(s) being identified/described (scope, granularity)
3. Context – determine the context(s) things are being identified within (granularity). For example, is it a concept/item/component/instance/etc., or what communities will it serve?
4. Overlap – consider the relationship of the identifiers to other similar identifiers and/or contexts, consider merging
5. Persistence – determine the expected identifier lifespan and strategies to preserve the relationship to the associated thing for that long (e.g. commitment level, resources, and policies)
6. Design the identifier system:
    - Identifier structure design – uniqueness, intelligence, actionability, persistence, extensibility, and communication of context
    - Addressability – is it acceptable to combine identifiers to identify a particular thing or are single standalone identifiers required?
    - Support – policies, processes, and mechanisms
7. Assign locally – implementation (within your scope of control)
8. Global uniqueness – wrap local identifiers with global authority identifiers for wider use
9. Use them (i.e. avoid using equivalent identifiers that may cause duplication or confusion)!

Note that upgrading existing identifiers may mean re-evaluating policies that have become taken for granted.

6. Conclusion

Correctly and appropriately identifying things requires critical analysis. Unfortunately, there is no magic formula for the selection and protection of identifiers but this is not all negative as we need a very flexible approach to cope with the wide range of situations.

However, the critical analysis needed to design the identifiers and identifier systems for our resources, schemas, etc. should be eased with a better understanding of what it is we intuitively do when we identify things and the kinds of aspects we need to consider.

Future investigations into the classifying and typing of identifiers, associations, and contexts may ease our identifier aches even further.

References

Coates, Tom. (2004, June). *Developing a URL structure for broadcast radio sites*. Plasticbag.org. Retrieved April 2, 2007, from http://www.plasticbag.org/archives/2004/06/developing_a_url_structure_for_broadcast_radio_sites/.

Coyle, Karen. (2006, July). Identifiers: Unique, persistent, global. *The Journal of Academic Librarianship, 32*(4), 428-431.

Fielding, Roy. (2002, October). *Re: now://example.org/car* [email]. Retrieved April 2, 2007, from http://lists.w3.org/Archives/Public/www-tag/2002Oct/0167.

Hillman, Diane. (2005, November). *Using Dublin Core*. Retrieved, 2 April 2007, from http://dublincore.org/documents/usageguide/.

IFLA, Section on Cataloguing. (1997, September). *Functional requirements for bibliographic records*. Retrieved April 2, 2007, from http://www.ifla.org/VII/s13/frbr/frbr.htm.

Kebbell, Adrienne, and Douglas Campbell. (2004, October). Managing digital objects and their metadata: challenges and responses. *Proceedings of the International Conference on Dublin Core and Metadata Applications, 2004*. Retrieved April 2, 2007, from http://purl.org/metadataresearch/dcconf2004/papers/Paper_09.pdf.

Kunze, John. (2003, August). Towards electronic persistence using ARK identifiers. *Proceedings of the ECDL Web Archiving Workshop 2003*. Retrieved April 2, 2007, from http://www.cdlib.org/inside/diglib/ark/arkcdl.pdf.

NISO. (2006, July). *Report of the NISO identifiers roundtable*. Retrieved April 2, 2007, from http://www.niso.org/news/events_workshops/ID-workshop-Report2006725.pdf.

NISO. (2006, March). *Definitions of terms related to identifiers*. Retrieved April 2, 2007, from http://www.niso.org/news/events_workshops/ID-docs/definitions.html.

Ogden, Charles, and Ivor Richards. (1923). *The meaning of meaning*. London: Routledge & Kegan Paul.

Paskin, Norman. (2003, January). On making and identifying a "copy". *D-Lib Magazine, 9*(1). Retrieved March 28, 2007, from http://www.dlib.org/dlib/january03/paskin/01paskin.html.

Peirce, Charles (1931-58). *Collected writings*. Cambridge, MA: Harvard University Press.

Powell, Andy, Mikael Nilsson, Ambjörn Naeve, and Pete Johnston. (2005, March). *DCMI Abstract Model*. Retrieved April 2, 2007, from http://dublincore.org/documents/abstract-model/.

Rust, Godfrey, and Mark Bide. (2000, June). *The <indecs> metadata framework: Principles, model and data dictionary*. Retrieved April 2, 2007, from http://web.archive.org/web/20060509143342/www.indecs.org/pdf/framework.pdf.

Saussure, Ferdinand de. (1974). *Course in general linguistics* (trans. Wade Baskin). London: Fontana/Collins.

Shafer, Keith, Stuart Weibel, Erik Jul, and Jon Fausey. (1996). *Introduction to persistent uniform resource locators*. Retrieved April 2, 2007, from http://purl.oclc.org/docs/inet96.html.

Vitiello, Giuseppe. (2004, January). Identifiers and identification systems. *D-Lib Magazine 10*(1). Retrieved March 28 2007, from http://www.dlib.org//dlib/january04/vitiello/01vitiello.html.

W3C Technical Architecture Group. (2007, January). *The use of metadata in URIs*. Retrieved April 2, 2007, from http://www.w3.org/2001/tag/doc/metaDataInURI-31.html.

W3C/IETF URI Planning Interest Group. (2001, September). *URIs, URLs, and URNs: Clarifications and recommendations 1.0*. Retrieved April 2, 2007, from http://www.w3.org/TR/uri-clarification/.

Weibel, Stuart. (2007, February). Failure points and manifestations. *Weibel Lines, 20 February 2007*. Retrieved March 29, 2007, from http://weibel-lines.typepad.com/weibelines/2007/02/failure_points_.html.

Xinhuanet. (2004, June). *Chinese names: A unique and beautiful name*. Retrieved April 2, 2007, from http://www.lechinois.com/chinesename/info/chinesenameunique.html.

Appendix A. Description Scope

Examples of what the scope could potentially be in a description of a newspaper article:

- The article in the physical newspaper
- The article in the second edition of the physical newspaper
- The idea behind the article
- The page in the physical newspaper the article appears on
- The physical newspaper (as a whole entity)
- The digitised page from the newspaper containing the article
- The thumbnail view of the digitised page
- The cropped image of just the article from the digitised page
- The web page delivering the digitised page image
- The web page delivering the digitised article image
- The PDF document containing the digitised article image
- The online version of the newspaper (as a whole entity)
- The web page for the individual article in the online version
- The archived web page(s)
- The database record in an indexing and abstracting database for the article
- The database record in a library catalogue for the physical newspaper
- The database record in a library catalogue for the digitised version of the newspaper
- The database record in a library catalogue for the online version of the newspaper
- The database record transformed into an alternative machine-readable format
- The concept of the article (the FRBR "Work")
- The translation of the article
- The article syndicated in another newspaper
- The online forum discussing the article
- The blog entry by the journalist that led to, or discusses, the article