Knowledgeone corporation

Do you really need a Taxonomy / Classification Scheme with a Records Management System?

Why are we still doing it the old way?

Contents

Background

Classification schemes are a way to group or order data; the objective being to group 'like' objects together. Classification schemes have been in use for tens of thousands of years, probably beginning when man first realized that there were different types of animals and plants.

We use classifications schemes both to make things easier to find and to add value to a group of objects. By adding value I mean that a classification (describing a group) may provide more information about the members of that group that is obvious from an analysis of a member; this could be referred to as semantics.

Classification schemes are used in all walks of life, for example; in business, in science, in academia and in politics. Are you a liberal or a conservative? Is it a mammal? If it is, is it a marsupial or a monotreme or a placental mammal? This last example illustrates the usual hierarchical arrangement of classification schemes.

In business, we have long used classification schemes to order business documents, that is, records of business transactions. We are all familiar with file folders and filing cabinets; these things are tools of a classification scheme. They make implementing a classification scheme easier as do numbering systems, colors, barcodes and Lektrievers.

I am sure that if someone who is responsible for managing physical files today was to be transported back 200 years to an office of that time there would be little difference in how paper was classified and managed. The office hours would of course be much longer and the pay much less but apart from that the job wouldn't be much different and a primary requirement in both offices would be a command of the alphabet and numbering systems.

In the first half of the twentieth century manual filing systems became very sophisticated with multi-part and 'intelligent' numbering systems and a number of paper tools like a 'keyword-in-context' (KWIC), 'keyword-out-of-context' (KWOC), index cards and a keyword thesaurus.

With the first commercial availability of mainframe computers in the early 1960s came our first attempts to computerize filing systems. It was also in the 1960s that we saw the first text indexing systems and the first sophisticated search algorithms.

The advent of text indexing and search algorithms allowed us to do a much better job of classifying data but more importantly, they allowed us to do a much better job of finding data.

Let's not get in a debate about terminology and acronyms

Our industry (information management to use an all-encompassing term) is often its own worst enemy. It creates terms and acronyms at will with both confusing and overlapping definitions. Then it wonders why normal end–users (as opposed to the pseudo-academics who are desperately trying to make records management into a science) exhibit first bewilderment and then disinterest. Let's look at a few examples:

RIMS – Records and Information Management System

RMS – Records Management System

DMS – Document Management System

EDRMS – Electronic Document and Records Management System

IAMS – Information Asset Management System

CMS – Content Management System

ECM – Enterprise Content Management

KMS – Knowledge Management System

> Do you realize that the process of records management is part of each of the above terms?

The pseudo-academics and marketing gurus can debate the differences and nuances of the above terms (and countless more, we keep inventing) until the cows come home or the rest of the population has died of boredom.

For my part I will stick with my old friend the world records management standard, ISO 15489. It tells us that records are evidence of a business transaction and that records are in any form including paper, electronic documents and emails (I know emails are electronic documents but the world generally differentiates them because emails are 'different').

So as far as I am concerned the term Records Management System or RMS includes everything we do and is easily recognized and understood so this is the term and acronym I will use in this paper.

And, records management is as far away from being a science as is economics. This isn't a putdown (especially as records and economics are two of my favorite subjects) it is simply a realistic appraisal of the nature of the profession.

Browsing versus searching

Classification systems are very good at making it easier for us to find information by browsing but not very helpful when we are searching. Let me explain what I mean by this.

Let's say I need to find a document that is part of a collection of documents to do with a new project to build a new bridge across the Nile River. It is called 'New Nile River Bridge Project 2008' (somewhat unimaginatively). I know the document is a cost estimate. I have classified all the documents for this project into two major sub-classifications, Requests for Proposals and Submissions. Logically, a cost estimate should be in the submissions file so without knowing much else I pull the 'New Nile River Bridge Project 2008 – Submissions' file folder and begin examining the contents beginning with the first page. This is a classical example of browsing. I couldn't go directly to the document I wanted but at least I knew exactly where to look.

Most classification systems require you to first 'browse' before finding the exact information you want; you usually have to examine multiple objects before you find the one you want. But this is what classifications systems are very good at; because they organize data in a logical (to a human being) way, we usually know where to begin looking. This is why a classification scheme works so well with a manual filing system (multiple cabinets or multiple shelves of file folders)

Classification schemes are great for physical data and, I would say, absolutely necessary for physical data; how else would you organize fifty-thousand file folders (containing seven and a half million pages) in a huge filing room with hundreds of shelves?

However, with computers I don't need to browse through multiple objects to find the one I want. By using techniques more appropriate to the computer than the filing room, I can search for and find exactly what I want almost instantly. I do not need to leaf through the file folder, I can go directly to the page or directly to the word. I can use the power of the computer.

Now the following statement will be probably seen as heresy by most practicing records managers but we actually don't need a classification system (Taxonomy) when computerizing records. We just need a way to index and then search for information.

Let's be honest about this; most end users don't understand classifications systems and find them cumbersome and impossible to use. They add form and clarity to the manual environment but they add complexity to the computerized environment. For most normal end users they actually become a barrier to storing and finding information.

However, try to tell a records manager that we really don't need their wonderfully thought out and proven (in the physical world) classification system when we computerize and then see the business go to a competitor.

The fact of the matter is we do not need the weight and complexity of a classification system when we computerize records; instead, we need a way to index it and find information instantly. We do not need to propagate systems designed for the manual world into the computer world (but we do). We need to organize our data so an ordinary end-user can easily find what they need without having to be a trained, professional records manager.

Classification systems are also much better with relatively static data than they are with dynamic data. I can easily design a classification system for a room full of file folders organized in an administrative or functional way because the data we are managing is relatively static. However, designing a classification system to handle electronic documents and emails is a nightmare because the data is not static and we would be constantly updating and modifying the classification system to accommodate the new types of data we are receiving. This is why for example; the naming convention for folders and sub folders in a shared drive environment quickly becomes an unworkable mess.

Indexing versus classifying

Now I know my interpretation of these two terms will not thrill everyone but the differentiation is an important part of my hypothesis.

Let's start by looking at two kinds of books, a reference book and a work of fiction. Both have tables of content (a classification system usually called a TOC) but only one (the reference book) has an index (usually).

The TOC for the reference book is both useful and often used. The TOC for the work of fiction is both not useful and rarely used (readers rarely need more than a bookmark).

The TOC for the reference book is way to organize information into a logical form grouping 'like' information together in chapters and sections. A TOC for the work of fiction is just a list of chapters; it serves little or no purpose for the typical 'end-user', the reader.

All the reader of a fiction book really needs is two things; a bookmark and a 'memory' of the author, title, cover combination so he/she doesn't accidentally buy it again at the airport bookshop before that dreaded long and boring flight.

The reader of the reference book actually needs both the TOC and the index for browsing (the TOC) and searching (the index).

A work of fiction doesn't usually have nor need an index because the end-user doesn't require it. A reference book usually has an index (unless the author is lazy) and it is often used to go direct to a page (or pages) and locate something very specific.

Drawing parallels with our broader topic, some information needs both a classification system and an index, some information needs just an index and some doesn't require either (e.g., works of fiction).

Generally speaking, scientific collections (if I may use that term) require a classification system (a scientific taxonomy); for example, the study of plant species and the study of animal species (e.g., using a phylogenetic classification system). Scientists simply could not communicate with each other without having a detailed and exact classification system in place. But, most end-users are not scientists; they are just people trying to find the best place to store something and want to find it again with the least amount of effort and pain. Nor do most end-users have much of a grasp of Latin.

Most end-users don't want a classification system (it is too hard to learn, unwieldy and too difficult to navigate) they just want a fast and easy way to find stuff; they want an index, a simple entry point to the information they need.

My contention is that we can solve all 'content management' and records management needs with a solution based on the application of a sensible, simple and self-evident (read that as easy to use or human-oriented) indexing system plus the required searching capabilities (i.e., covering both Metadata and full text). Or, records management systems do not need classification systems; classification systems add only complexity to a records management system, not value. There is a better way.

What indexing system?

Whenever I consult with customers who are contemplating the capture and organization of data (hopefully into information) I always give the same advice. That is, "When you are thinking about how to index data first think about how you will find it later." Ask this key question of your end-users, "When you are about to search for information what do you usually know about it?" For example:

- Do you know the last name?
- Do you know the first name?
- Do you know the date of birth?
- Do you know the employee number?
- Do you know the date or the date range?
- Do you know something about the subject?
- Do you know the file number or case number?
- Do you know the contract number?
- Do you know any terms?
- Do you know the street?
- Do you know the town name?
- Do you know the postcode/zipcode?

A good indexing scheme reflects real life usage of the system; it reflects how ordinary humans work and 'see' information. Put simply, it indexes the information people will later need to search on. It indexes the information people understand and are comfortable with because it is self-evident. It indexes the 'natural' or 'common' Metadata.

Let's take emails as an example (and a very important example as by all accounts, emails now account for around eighty-percent of all business transactions).

Indexing Emails

An email is usually described as an unstructured document (the same way a Word or Excel document is described as being 'unstructured') but in fact it does have structure. Even better, everyone is familiar with an email's structure so we have very little to teach end-users; that is, we have a simple and self-evident 'natural' set of Metadata items to index.

1. Date of email
2. Sender
3. Recipient
4. CC
5. BCC
6. Subject

7. Text of the body of the email
8. Text of any attachments

For any normal end-user trying to find an email this is how they would envision an appropriate search. They wouldn't care that the email has been classified down to 6 hierarchies using the world's most sophisticated Business Classification Scheme (BCS).

I further guarantee that if you asked the end-users what they 'knew' before doing the search their knowledge would pertain to one or more of the 8 elements above.

If you told end-users they could search for emails using one or more of the above 8 elements they would be happy and they would understand how to do the search and most of their searches would be successful.

Understanding what end-users typically 'know' before they do a search determines what elements you have to index. This is the key to implementing a successful indexing system.

The above 8 elements of an email are self-evident insomuch as, "Of course I need to be able to search on the sender or recipient or subject…."

Indexing Electronic Documents

Now let's look at ordinary electronic documents (i.e., not emails) because they are much less structured. We all know there are ways to add a common structure using features of MS Office like the information dialog box (asking for keywords etc) and templates and smart tags but these things are rarely and inconsistently used. Different electronic document types also have different properties (e.g., a TIFF file does not have an author as a system property) so it is difficult to have an across the board standard.

With shared drives we usually find some form of 'evolved' classification system because managing electronic documents in shared drives is akin to managing millions of pieces of paper in tens of thousands of file folders in hundreds of filing cabinets. Unfortunately, the good intentions and purity of design of the original architects of the shared drives folder/sub folder naming conventions (a classification system) are soon corrupted as users make uncoordinated changes and the structure soon becomes unwieldy and incomprehensible to all but a few geniuses (and even they struggle).

This is the major real life difference between a 'store' of paper and a 'store' of electronic documents. With paper the classification system is rigid and enforced because it is centrally controlled. With electronic documents (i.e., in shared drives) the classification system is not centrally managed or controlled and it soon reverts to a state of anarchy.

In my opinion shared drives are OK for the creation of documents (i.e., a work area) but not OK for the management of documents. In fact I would say shared drives are absolutely hopeless for the management of documents as history and practice will attest.

Not everything that is added to a shared drive is a corporate record (far from it) and shared drives are not secure. I have examined a lot of shared drives and I always find an enormous amount of redundant, duplicate and irrelevant data. One of the oldest acronyms in the IT business is GIGO or Garbage In equals Garbage Out. The last thing we want to do given the huge amount of data we are required to capture, is capture garbage.

In my ideal world we would use 'smart' Agents to monitor all shares drives and analyze all new documents as they are created. The Agents then determine which documents to capture (corporate records) and which to ignore (garbage like birthday or lunch invitations). All corporate documents are immediately stored in the RMS, versioned and indexed. I would also delete the document from the shared drive once it had been successfully saved in the RMS. Obviously these Agents have to be rules-driven and the rules have to apply to all the available properties of an electronic document.

All subsequent check-outs and check-ins should then be done from the RMS.

Once again we need an appropriate indexing system and once again we need to ask, “What do people know at the time of the search?” For example:

1. Original filename
2. Original path/filename
3. Type/suffix – e.g., .DOC, .XLS, .PDF, etc
4. Author
5. *Subject
6. Date created
7. *Contract number
8. *Case number
9. *Client number
10. *Reference number
11. *Customer name
12. *Type of document, e.g., customer complaint

This is where full text indexing is even more important than when searching for emails because we have much less structure to utilize. All the elements listed above with an asterisk (*) require that the electronic document be full text indexed. We need to be fully aware of its content because we have little structure to work with.

We also need to think carefully about how we are going to build our search. With emails it was easy (just utilize the common elements) but with electronic documents it has generally been thought that it is impossible to design a structured search with a fixed set of search parameters (Metadata elements) because the data in the documents varies too much. I obviously do not agree.

Metadata and the Dublin Core

Let me quote from the Dublin Core website:

http://dublincore.org/

“The Dublin Core Metadata Element Set is a vocabulary of fifteen properties for use in resource description. The name "Dublin" is due to its origin at a 1995 invitational workshop in Dublin, Ohio; "core" because its elements are broad and generic, usable for describing a wide range of resources.”

To quote Wikipedia:

http://en.wikipedia.org/wiki/Dublin_Core

“It provides a simple and standardized set of conventions for describing things online in ways that make them easier to find. Dublin Core is widely used to describe digital materials such as video, sound, image, text, and composite media like web pages.”

The Simple Dublin Core Metadata Element Set (DCMES) consists of 15 elements.

1. Title
2. Creator
3. Subject
4. Description
5. Publisher
6. Contributor
7. Date

8. Type
9. Format
10. Identifier
11. Source
12. Language
13. Relation
14. Coverage
15. Rights

I will leave it to the reader to go to the URLs I have listed above for a more detailed explanation of the properties of each of the above 15 elements.

To my mind the Dublin Core is an excellent set of elements for describing almost any ‘record’ because it is both simple and appropriate to both computers and ‘normal’ end-users. As a professional, I like the elegance of the Dublin Core.

I also like the basic principle because it fits in with my hypothesis. That is, there is a better way to store, index and find records than a complex and unwieldy Taxonomy. There is a simpler, easy to implement, easier to manage and easier to use way to manage records/content. The fifteen Simple Dublin Core elements are a prime example of this principle.

The Full Solution?

- We need an application that stores documents of all types, i.e., all types of content.
- We need an application that indexes both Metadata and full text.
- We need an application with a customer configurable Metadata model.
- We need an application that allows you to search on both Metadata and full text in a single search.
- We need a search that combines BOOLEAN and numeric operators, e.g., AND, OR, NOT, =, <, >, etc.
- We need a ‘standard’ Metadata definition (a Class if you will) that includes a simple (not more than 20 in my estimation) set of data elements that includes all of the elements necessary to index all of the types of documents (including file folders and paper) that you manage. This is a Master Metadata Class; it is a superset of the set of data elements required for any particular type of document/record.
- We need an application that includes all types of data capture, e.g., from the file system, from the native application, from a scanner, etc.
- We need an application with a comprehensive security system.
- We need an application with all reporting options, e.g., both standard reports and ad hoc reports.
- We need an application with a configurable audit trail.
- We need an application with comprehensive import and export capabilities.

The standard Metadata definition (Master Metadata Class)

I don't yet have a fancy name or acronym for this so for now let's just call it Frank's Metadata standard.

I have come up with a limited set of elements that I believe can be used to index and find any type of record, paper or electronic. I have borrowed heavily from the Dublin Core because it makes good sense to do so; there is no need to reinvent the wheel.

#	Element	Explanation
1	Title	A name given to the record. Typically, a Title will be a name by which the record is formally known. Text, e.g., "Business Plan for 2010"
2	Author(s)	The sender or author, E.g., Mark Twain or f.mckenna@k1corp.com
3	Dated	The original date of the document or published date
4	Date Received	Date received by the recipient or recipient's organization, whichever is the earlier
5	Original Name	e.g., filename or file\pathname for electronic documents - C:\franks stuff\sample.xls
6	Primary Identifier	An unambiguous reference to the record within a given context. E.g., The file number
7	Secondary Identifier	An unambiguous reference to the record within a given secondary context. E.g., The case number or contract number or employee number
8	Barcode	Barcode number or RFID tag
9	Subject	The topic of the record. Typically, the subject will be represented using keywords or key phrases. Recommended best practice is to use a controlled vocabulary.
10	Description	An account of the record. Description may include but is not limited to: an abstract, a table of contents, a graphical representation, or a free-text account of the record.
11	Content	Words or phrases from the text content of the main document and attached documents
12	Contents	Description of contents if the document is a container, e.g., an archive box
13	Recipient(s)	Addressed to, sent to etc. People or organizations.
14	CC recipient(s)	CC and BCC recipients
15	Publisher	An entity responsible for making the record available. Company or organization that either published the document or that employs the author
16	Type	The nature or genre of the record, usually from a controlled list, e.g., complaint, quotation, submission, application, etc.
17	Format	The file format, physical medium, or dimensions of the record. E.g., Word, Excel, PDF, etc
18	Language	e.g., English, French, Spanish
19	Retention	The retention code determining the record's lifecycle
20	Security	Access rights, security code, etc

My contention is that by using an 'index set' like the above 20 Metadata elements you can index, manage and retrieve any 'record' regardless of form and content.

You don't need to use Frank's Metadata standard, you can modify my list and come up with your own. However, please don't go overboard; try to keep to 20 elements. This is one task where the age old acronym of KISS is absolutely applicable.

What about all the standards 'out there'?

There is a plethora of local, state, federal, industry and international standards pertaining to the management of records. Examples are DoD 5015, MoReq2, Dublin Core, ISO 15489, VERS etc and literally thousands of standards for Metadata.

The problem with most of these standards is that they are extraordinarily difficult to read and understand. I would draw a parallel back to the times when the Bible was in Latin but Christians were supposed to order their lives by its teachings. The problem being that only about 0.025% of Christians spoke Latin. Ergo, how do you order your life by a book you can't read?

My assertion is that most records managers do not understand the standards they are charged with enforcing. I have experienced this hundreds of times in face to face discussions with records managers who glibly quote a standard as if it were one of the Ten Commandments but then can't answer any questions about the details of that standard. I can honestly say that I have never met a records manager who fully understood (or had even fully read) the standard(s) they were charged with enforcing.

The problem isn't with the records managers; it is with the people who write the standards. The standards are not written for records managers, they are written for academics and technical people (i.e., systems engineers who are experts in XML). Just like the Latin Bible, they are not written in the language of the intended user.

I have to read and understand standards because I am in the business of designing software that meets standards. I find the majority to be simply awful. They are badly written, badly constructed and are never a single 'easy-to-digest' and complete document. I suffer brain fade from having to follow multiple links and references to other documents. It is no wonder the majority of records managers don't understand some of these standards because they are literally unintelligible to a normal human being.

And even when you do think you have a grasp of the fundamentals there are always multiple points to be clarified (as to the exact meaning) with the standards authority.

Maybe we need a new standard for the writing of standards? But, who would we get to write it?

What about Retention schedules?

This should probably be the subject of another paper because retention schedules have also become way too complex, unwieldy and difficult to understand and apply. It is almost as if some authority created a pretty good retention schedule 30 years ago and then each year since felt compelled to add stuff. Mark Twain (Samuel Clemens) would have described it as "Growing like Topsy".

The question will be, "How can I do away with my classification system when my retention codes are linked to it?"

I have looked at hundreds of retention schedules and every single one has been way too complicated for the organization trying to use it. Another problem is that very few of the authorities that compile retention schedules do so with computers in mind. This means that we end up with lots of very vague conditional statements that are almost impossible to computerize.

I can recall ‘discussions’ with authorities that insisted on publishing the retention schedule in Word format and then demanding we automate the import. I tried in vain to convince them to use Excel because I could import from Excel but not from Word. When it is Word format the only solution is to hire a clerk to do manual data entry into a form that is suitable for importing by a computer system. This is also when you turn the vague conditional statements into something a computer can handle.

Most retention schedules are written for archivists to read, not for computers to process. This is the heritage of retention schedules; they assumed an appraisal process by a trained and expert archivist.

The Continuum model or ‘Whole of Life’ model or File Plan model all assume we will allocate a retention code at the time the record is created, not during a later appraisal process. This made much more sense and allowed us to better manage the record throughout its life cycle. However, many such schemes also linked the retention code to a classification term or embedded the retention codes within the classification system. This of course made the classification system even more complex and difficult to understand and apply. Then we moved up to multiple retention codes per record and no one ever understood how that was supposed to work except maybe an archivist.

I have visited hundreds of records management sites over the last 25 years and discussed retention with records managers at most of them. I honestly can’t think of any records manager I met with that didn’t think the retention scheme was way too complex for their needs.

To my mind no organization needs more than ten retention codes (shortest period, longest period and eight in between) and three life cycles (e.g., active, inactive, destroyed). This is also probably heresy to a lot of the records management profession but, I would ask them to think about the proposition that something that was entirely appropriate to the manual world is not necessarily entirely appropriate to the computerized world. There is an easier and simpler way to manage retention and there is no need to embed retention codes into the classification system just as there is no need for a classification system in any modern, computerized records management system.

What about File Folders and Archive Boxes?

This is the classic stumbling block. This is when the records manager tells you that all the standards require you to use the same taxonomy for emails and electronic documents that he/she uses for traditional file folders and archive boxes (and has used for 200 years).

You need to explain that the classification from the manual paper handling world is inappropriate to the computerized world, that it is an anachronism. You need to explain that all it will add is complexity, massive cost, confusion and a seriously negative attitude to end-users. You should say it is time to discard techniques and tools from the eighteenth century and adopt techniques from the twenty-first century. You should say you have a much better way. Then you should probably duck and run. Failing all else, blame me and give them my email address.

Frank McKenna, CEO Knowledgeone Corporation

f.mckenna@knowledgeonecorp.com

Appendix A – The Dublin Core

The Elements

Term Name: contributor	
URI:	http://purl.org/dc/elements/1.1/contributor
Label:	Contributor
Definition:	An entity responsible for making contributions to the resource.
Comment:	Examples of a Contributor include a person, an organization, or a service. Typically, the name of a Contributor should be used to indicate the entity.
Term Name: coverage	
URI:	http://purl.org/dc/elements/1.1/coverage
Label:	Coverage
Definition:	The spatial or temporal topic of the resource, the spatial applicability of the resource, or the jurisdiction under which the resource is relevant.
Comment:	Spatial topic and spatial applicability may be a named place or a location specified by its geographic coordinates. Temporal topic may be a named period, date, or date range. A jurisdiction may be a named administrative entity or a geographic place to which the resource applies. Recommended best practice is to use a controlled vocabulary such as the Thesaurus of Geographic Names [TGN]. Where appropriate, named places or time periods can be used in preference to numeric identifiers such as sets of coordinates or date ranges.
References:	[TGN] http://www.getty.edu/research/tools/vocabulary/tgn/index.html
Term Name: creator	
URI:	http://purl.org/dc/elements/1.1/creator
Label:	Creator
Definition:	An entity primarily responsible for making the resource.
Comment:	Examples of a Creator include a person, an organization, or a service. Typically, the name of a Creator should be used to indicate the entity.
Term Name: date	
URI:	http://purl.org/dc/elements/1.1/date

Label:	Date
Definition:	A point or period of time associated with an event in the lifecycle of the resource.
Comment:	Date may be used to express temporal information at any level of granularity. Recommended best practice is to use an encoding scheme, such as the W3CDTF profile of ISO 8601 [W3CDTF].
References:	[W3CDTF] http://www.w3.org/TR/NOTE-datetime
Term Name: description	
URI:	http://purl.org/dc/elements/1.1/description
Label:	Description
Definition:	An account of the resource.
Comment:	Description may include but is not limited to: an abstract, a table of contents, a graphical representation, or a free-text account of the resource.
Term Name: format	
URI:	http://purl.org/dc/elements/1.1/format
Label:	Format
Definition:	The file format, physical medium, or dimensions of the resource.
Comment:	Examples of dimensions include size and duration. Recommended best practice is to use a controlled vocabulary such as the list of Internet Media Types [MIME].
References:	[MIME] http://www.iana.org/assignments/media-types/
Term Name: identifier	
URI:	http://purl.org/dc/elements/1.1/identifier
Label:	Identifier
Definition:	An unambiguous reference to the resource within a given context.
Comment:	Recommended best practice is to identify the resource by means of a string conforming to a formal identification system.
Term Name: language	
URI:	http://purl.org/dc/elements/1.1/language

Label:	Language
Definition:	A language of the resource.
Comment:	Recommended best practice is to use a controlled vocabulary such as RFC 4646 [RFC4646].
References:	[RFC4646] http://www.ietf.org/rfc/rfc4646.txt
Term Name: publisher	
URI:	http://purl.org/dc/elements/1.1/publisher
Label:	Publisher
Definition:	An entity responsible for making the resource available.
Comment:	Examples of a Publisher include a person, an organization, or a service. Typically, the name of a Publisher should be used to indicate the entity.
Term Name: relation	
URI:	http://purl.org/dc/elements/1.1/relation
Label:	Relation
Definition:	A related resource.
Comment:	Recommended best practice is to identify the related resource by means of a string conforming to a formal identification system.
Term Name: rights	
URI:	http://purl.org/dc/elements/1.1/rights
Label:	Rights
Definition:	Information about rights held in and over the resource.
Comment:	Typically, rights information includes a statement about various property rights associated with the resource, including intellectual property rights.
Term Name: source	
URI:	http://purl.org/dc/elements/1.1/source
Label:	Source

Definition:	A related resource from which the described resource is derived.
Comment:	The described resource may be derived from the related resource in whole or in part. Recommended best practice is to identify the related resource by means of a string conforming to a formal identification system.
Term Name: subject	
URI:	http://purl.org/dc/elements/1.1/subject
Label:	Subject
Definition:	The topic of the resource.
Comment:	Typically, the subject will be represented using keywords, key phrases, or classification codes. Recommended best practice is to use a controlled vocabulary. To describe the spatial or temporal topic of the resource, use the Coverage element.
Term Name: title	
URI:	http://purl.org/dc/elements/1.1/title
Label:	Title
Definition:	A name given to the resource.
Comment:	Typically, a Title will be a name by which the resource is formally known.
Term Name: type	
URI:	http://purl.org/dc/elements/1.1/type
Label:	Type
Definition:	The nature or genre of the resource.
Comment:	Recommended best practice is to use a controlled vocabulary such as the DCMI Type Vocabulary [DCMITYPE]. To describe the file format, physical medium, or dimensions of the resource, use the Format element.
References:	[DCMITYPE] http://dublincore.org/documents/dcmi-type-vocabulary/