**Please note that information provided in this article and the subsequent discussion is the opinion of the author(s) only and do not necessarily represent the views of IMS.**

# Paradigm Shifts and Challenges for Instructional Designers

An Introduction to Meta Tags and Knowledge Bits

Steven Schatz
Steven Schatz Information Design and Indiana University

schatz@powerstart.com

## Part One
## Introduction to The What and Why of Meta Tags

Meta tagged knowledge bits can fundamentally change the way designers build and consider training. We are on the verge of being able to provide learning customized for each specific learner at a specific time, taking into account their learning styles, experience, knowledge and learning goals. We are on the verge of a time where trainings are updated on the fly, using the very latest information. We are on the verge of being able to reuse pieces of trainings instead of starting from scratch every time. There is an excellent potential for trainings that will cost less, take less time to build and will be both more interesting and more effective.

This power is becoming available. A combination of the web, feedback mechanisms and, most significantly, meta tagged knowledge bits form the basis of this new approach. However, to take advantage of new powers requires new models for planning, design and development of training - a new paradigm for considering the what, where, when, why and how of training.

This article is a step along that road. It first attempts to offer a basic explanation (in terms mere mortals might understand) of what meta tagging is, how it may be applied and why that is useful. After covering those basics, most of the article focuses on a model to guide designers who want to create a unique meta tag system for their organization. This article attempts to offer specific prescriptive procedures to help guide the pioneer in this most exciting adventure.

### The Beginning: Why should you care?

*Scenario One*

Joe is the new admin assistant in product marketing for MegaGlop, Inc. Instead of a week of orientation, Joe is given a single day of orientation and 32 hours of training **to be done as he needs to learn**. A half hour a day is set aside for training. He is introduced to his **Performance Support Portal (PSP)** on his desktop computer. Here he can view performance support tools, read frequently asked questions, chat with others in his position and in the company, look up important information, and ask questions of cyber and human experts. His PSP also tracks his training progress. When his daily training time comes around, his PSP suggests skills he needs to learn. Based on his preferences, the short training bits from which he picks are in his preferred media - movies, animations or text. The trainings he sees are customized for his specific job, based on his knowledge and past training results. Information is always current. When, during his day, Joe runs into a problem (for example, how to fix the copy machine) he need only ask for a training at his PSP and he can get a walk through the process, a step by step guide, diagrams, and support phone numbers.

*Scenario Two*

Joyce is an instructional designer. She is charged with developing basic training for all support staff for an new department at Universal Juggling, Inc., a multinational corporation. She has an impossibly short time frame. However, she performs a general task analysis for each position.

She lays out what knowledge is required for each position. Then she searches the company database of knowledge bits for trainings that cover those needs. Her search yields over a hundred bits in different media each of which are specifically tagged by skill and skill level. She reviews the bits over a couple of days and finds there are existing bits for 60% of the necessary competencies. She then contacts three suppliers of bits which have high end animations and movies available on a per use royalty fee. Through these vendors, she finds bits to cover another 20% of the training needs. She already has developed a training delivery template which plugs into the company's Performance Support Portal. The template allows her to assign bits of text, graphics, animation and other media to specific pages. She spends two days adding company specific meta tags and setting up search and retrieve functions. A week after the task analysis is finished, Joyce has 80% of the training for an entire department ready to go. She has plenty of time to develop the 20% of new training that needs to be developed. She decides to take long lunches and three day weekends for the next month.

When the names of the positions change and some of the job duties shift 2 weeks before deadline, she is able to make the changes in an afternoon.

*Scenario Three*

You have a team of young, smart, sales execs that are on the go 18 hours a day. The last thing they want is to be forced into taking training sessions. They can't afford the time. They do, however, have learning needs. You set up a training system which lets them ask for help on a particular task and the system will offer them all the different bits available. They can then pick the bit that matches their wants, selecting from options including media, skill level and length of time.

**What are Knowledge Bits?**

As simply as possible, a knowledge bit is a piece of instruction. In the context of this article it is a piece of instruction that can be viewed on the web. It can be as small as a single web page with a bit of text, a single picture, diagram, animation, movie or other object. It can be as large as an entire training, but usually will not be. Knowledge bits can be viewed on their own - a text block, for example. They can also be combined on a single page.

For instructional designers, the idea of knowledge bits requires a small but immense change in thinking. Instead of looking at trainings as linear processions with a beginning, middle and end, we must now look at trainings as clusters of independent, stand alone bits of knowledge. They are certainly related to each other and they may be viewed together, **but they may also be viewed singly**. Just as you can enter a web site at any page and leave at any point, so too can training consumers. Lose the notion of a class of eager learners trapped before you for a day. These new consumers of training can come in at nearly any point in the training, stay as long or as short as they wish and leave when either when they are bored or when they have learned what they want. Bits of the training may be used in dozens of different trainings for different people. Designers will now develop instructional goals, piece together knowledge bits based on those goals and develop clear navigation. A much greater emphasis must be placed on developing clear instructional goals, for it will be these goals which guide what should be offered. In addition, navigation becomes crucial. Trainings must be developed to allow, indeed to help the learner get to exactly the point they wish, and then helping them learn and understand that exact piece of information, knowing that once they get what they want, they will leave - no evaluations…no thanks…no flowers.

This is the promise of knowledge bits. They are the individual grapes in a bunch. Consumers may eat one or all - it's up to them, not us.

As Downes (http://www.atl.ualberta.ca/downes/naweb/Learning_Objects.htm) points out, the idea of knowledge bits is not new. Teachers currently use instructional support tools created by others to support teaching - texts, films, videos, lesson plans. All these teaching aids are currently used. They offer pieces of knowledge which an instructor or designer can include within instruction. Knowledge bits allow the trading or acquisition of bits of any size - as small as a single picture or a text block, as large as an entire training.

The difference with knowledge bits on the web is that these pieces can be more easily accessed, they can be more easily found, they can be located by computer, they can be easily updated, they can seamlessly tied in with other bits to make a more complete training. For example, a designer can use a picture from a previous training, a video from an outside vendor and write a series of assessment items, combine it all and make one training module.

Downes predicts knowledge objects will become much the same as texts, films, charts and other educational support products. Libraries of bits will be available for sale or rent from large publishers. These often will feature animations, graphics, video and other "sexy" production values that would be impossible to include for an in house training. However, as these production houses will certainly plan to sell the bits several times, the economies of scale allow a higher end production.

Designers will use learning goals to piece together courses from their in house library of bits and from rented or purchased bits (which can be charged on a per use basis). The sections of the training for which no bits exist, the navigation and any cohesive bits necessary will be produced. As you can see, the total amount of production may well be significantly reduced.

### What Are Meta Tags?

Meta tags are the mechanism that will allow you and/or your learners to find specific knowledge bits quickly and effectively. Actually, they allow a computer (specifically a search engine) to find specific bits. Meta tags are information about knowledge bits. They are put into the heading of a web page. You don't see them when you look at the web page, but they are in the source code of the page. So, a search engine can read the tags and deliver what you need. Here is an example of a meta tag:

```
<dc:Language> en </dc:language>
```

This tag says that the language of this particular knowledge bit is English. (en is the abbreviation for English).

dc in the above example stands for Dublin Core. Dublin Core is a meta data **schema**. There are two important concepts in this seven word sentence. The first is meta data. What is the difference between meta tags and meta data? Meta tags are specifically the html tags which describe a web based resource (a web page with something on it). Meta data can describe anything - a web resource, a statue, a book, a painting, geospatial data - anything. Meta data that is located in the html header which describes a web resource is meta tagging. OK? It is a bit confusing. If it is too confusing, don't worry about it. You will be adding meta tags and if someone says meta data…just translate that to be meta tags for your purposes.

The second concept in that short sentence (Dublin Core is a meta data schema.) is schema. A schema is a series of tags which combine to describe the knowledge bit. When you look up a book in a library's card catalog (or online), the information is displayed based on a schema. Certain important information is returned in a set format. By looking at it, you can tell the author, title, where the book can be found, and other important information. **This is very important!** The only way you can easily call up useful knowledge bits is by having a useful tagging schema. To understand the importance of this, go to any search engine on the web and do a search for human resources. A recent search yielded slightly over 1,097,000 hits. This is not useful. The reason these searches pull up so many matches is that the search engines have to generalize, compromise…try to figure out what you want and mean when you type in a search topic. You want to be able to do a search that yields only those bits which really relate to your instructional needs. The way that you do that is by having your search engine look at your tags. What tags you have are up to you. All the tags you decide to have make up your schema. Maybe you will have knowledge bits that are in many languages - you will want a tag that tells you in what language the bit is authored. Maybe you have three main skill areas and three skill levels in your company. Wouldn't it be nice to be able to search for a specific skill and level? Your schema will set up what tags you will have (in this case, one for level and one for specific skill). The tags you have will set up what you can search for.

So, a schema is a bunch of tags. Schemas may have just a few tags (Dublin Core has 15). They may have hundreds. Even when they have hundreds, most tags are not required. When you see a knowledge bit that is tagged, it is a web page that has a long string of tags within the heading. These tags can go on for several pages (and often do). They do not display when the page displays, but the search engines can read the tags. To see an example of a meta tagged page using the Dublin Core schema, go to www2.sub.uni-goettingen.de/metaform and view at the source of the page. See all the tags that say META NAME? Those are the meta tags using the rules of set down in the Dublin Core schema.

The focus of this paper is to provide a model for you to develop your own unique schema of meta tags for your organization so that you can search and retrieve knowledge bits in a fast and useful manner. It does not cover the specifics of writing the schema. That is done in XML based on the specifications set out by the World Wide Web Consortium. If you want to see the technical basis, you can…**but it is not necessary at this time!!!** In fact, it may scare or confuse you. What is most important now is considering how to develop your own schema. This conceptual work requires you apply your skills and experience to a new paradigm of training. This is challenge enough! A future paper will include a guide to building the tags and tagging the bits. For now, leave the technical to later or to an experienced web programmer. (You can see the basic documents at the following URLs: 1. The Resource Description Framework (RDF) Model and Syntax Specification at http://www.w3.org/TR/REC-rdf-syntax/ and 2. Resource Description Framework (RDF) Schema Specification 1.0 at http://www.w3.org/TR/red-schema/.)

### How will designers use meta tagged knowledge bits?

We suggest three basic applications for this new approach.

1. Fast production of training using your own or vendor built bits.
2. Allowing free access for self directed learners.
3. A hybrid of the two with suggestions to assist free access.

> *A brief but important aside* It is important to consider the medium of web based training. Every medium has it's own strengths and weaknesses. The web is very good for presenting graphics, color and interactivity. It is not a great medium for delivering great quantities of text. Perhaps it's largest handicap, from an instructional standpoint, is the isolation in which learners learn. A learner has control over the training flow, but they often have nowhere to turn if they need help or guidance. They have no one with whom to share their thoughts, inspirations and ideas. They have no sense of community. They have no community to help them put information into perspective, to support and encourage creating meaning. This is a huge challenge for web based trainings. In the excitement over the power of meta tags, we must not overlook the importance of feedback mechanisms - email, chat, bulletin boards and other

methods to support interaction and support community. The term **Performance Support Portal** conveys the idea of a web space which includes:

1. Training,
2. Feedback mechanisms
3. What's New? sections to keep everyone up to date and
4. "Look up information" (tax tables, shipping rates, maps, etc.) which need not be "learned", just must be accessed in the course of one's job.

This paper's focus is on the creation of the learning elements within a Performance Support Portal.

1. Production of training by instructional designers
   When a needs analysis indicates a need for training, designers now usually create training new from scratch. Some pieces of previous trainings may be used, but not in any systematic manner. With knowledge bits, instructional goals become very important. They provide the basis for the search a designer will undertake. After looking at the needs analysis and developing instructional goals, the designer lays out an instructional strategy. In what order should the instruction be presented? What is prerequisite knowledge at different points of the training? What is the terminal result desired at different points? How will goals be assessed? Once the designer has an idea of where the learner should be throughout the training (remember, the learner may enter and leave at many different points), a search of knowledge bits - both in house and from vendors - is done. The resulting bits are reviewed and assigned places within a training. Depending on the instructional needs and the sophistication of the web team, bits may be used either in their entirety, (as a web page or series of web pages) or as pieces of pages (A page may have a bit which is a piece of text, another bit which is a movie and a third which has an assessment question - all of which come from different sources.). Instructional goals for which an existing knowledge bit cannot be found, as well as navigation and cohesive elements will be authored (and tagged for later possible repurposing).

   The hallmark of this application is that the designer picks the bits to be used by searching and reviewing bits. Of course, as an organization begins to tag their bits, there will be very few bits to review. However, in time, this number can become very large. Your tagging schema will allow you to do a search which comes up with just those bits which meet your needs. The more exact your schema, the more exact your search.

   Depending on the organization, in time, a significant percentage of bits for any training can already be built and ready to plug in. Trainings can also be easily kept up to date. When a training calls an existing bit, it is actually like referencing a web page. A bit may be used in many different trainings. By updating the bit, all trainings using that bit will be updated. This holds forth the promise of greatly reduced cost and faster turn around time. Finally, as more vendors begin to offer knowledge bits, it will be possible to have very high end graphics, animations and other learning support bits for a very affordable price. The only problem with these bits is that they will not use your proprietary tagging schema, but will instead use a more universal schema. They will work with your trainings, it will just require a bit more review of bits to match them to the training needs.

2. Allowing free access for self directed learners.
   In some organizations, with some populations, the instructional designer's job is not to select and guide learning. It is to make learning available as efficiently as possible. Fast paced, ever changing organizations often have at least a subset of the work force who have neither the time nor the patience to attend a training class. Furthermore, the tasks are so fluid that traditional methods of measurement (task and needs analyses) cannot

capture the nature of the job. In circumstances like this, the best approach is to provide well defined descriptions of bits available and effective search tools, so that learners can find what they need when they need it. They will do the selecting, not the designer. The designer will become more the "learning coordinator". They will act as librarian for knowledge bits - providing ever better descriptions, tags and search techniques. They will take on a greater learning support role (with the feedback mechanisms in the Performance Support Portal) by answering questions, guiding learning and helping locate learning opportunities. Finally, they will keep everyone up to date with postings to the "What's New" sections of the PSP.

3. Hybrid combination
   In some organizations, the designer's role will become a combination of the above two. Particularly in support positions, the designer will outline instructional goals which the employee should attain. At scheduled training times, the learner will check in with their training plan and will be presented with a list of skills they need to learn. Upon picking a skill, they will be given the option of selecting from several knowledge bits all of which teach the skill, but do so in different ways - different mediums, different pedagogy, different depth of treatment, different lengths of time. The learner may choose which bit to use. They must, at the end, successfully complete an assessment, but they control how they learn.

**Limitations**

It is important to be aware of the limitations of this powerful tool. In particular, this approach has important limitations in rapidly changing environments because of requisite delays. There are three time delays between a need for training and the delivery of training. 1) Meta tagging is a time consuming process, even when as many steps as possible are automated. 2) Developing a unique schema for your organization is even more challenging (although it is only necessary once with occasional updates). 3) Training is inherently a conservative function. By the time a process is established and widespread enough to require designers to understand it and develop a training to "initiate" others in such a process, it is nearly always old news. After that, the process of needs analysis, task analysis, design, production and deployment of training is necessarily time consuming. Anything that is fast moving and fast changing is rarely codified in training. It is part of the tacit knowledge of individuals within a company.(Nonaka, 1994; Fahey & Prusak, 1998; Leonard & Sensiper, 1998) It may be established enough to become tacit knowledge of a group of individuals. (Encultured knowledge) (Blackler, 1995; Leonard & Sensiper, 1998) However, that kind of fast moving, fast changing knowledge is not a good match for meta tagged knowledge bits.

In that kind of environment, the designer cannot hope to capture and tag knowledge. Instead, a strategy combining knowledge management with a tagging schema which can be easily applied must be designed to support capture and sharing of knowledge **by those creating and explicating knowledge.** While the author finds this a fascinating study, it is beyond the purview of this paper. For this paper, we shall examine knowledge bits that have some "shelf life".

## There are already schemas out there

There have been many attempts to build schemas. There are currently dozens of international schemas which cover everything from architecture to art to geospatial data to bibliographic references to everything. There are a few that are strictly focused on instruction. The Dublin Core (www.purl.org) is one of the first. It attempts to be a basic reference to all schemas - compromising detail for breadth of application. It has only 15 key elements. By contrast, many schemas have several hundred elements. They aim for a much smaller, but more specific audience. To see some of the existing schema, go to www.blm.gov/gis/nsdi.html this is just one of many sites which point to different schemas.

In the world of instruction, there have been two large projects which focus on building schemas which may be universally used to allow different knowledge bits from vendors or individual producers to be used in other trainings. The goal is interoperability - the ability to plug and play knowledge bits from any source into any source. The Instructional Management System project (www.imsglobal.org) of Educause (www.educause.edu) is an organization made up of software companies, training producers and educational institutions (mostly universities). They developed the first specification for meta tags focused on instruction. In January, 2000 ADLNet (Advanced Distributed Learning Network at www.adlnet.org) released their SCORM 1.0 (Sharable Courseware Object Reference Model). SCORM focuses mostly on training for the defense department. There will undoubtedly be more schemas released in the next few years.

**Why build your own schema?**

If there already are schemas to use, with more on the way, why should you worry about creating your own? The answer is simple - customization. The more customized the tags, the more exact the search. A really good tagging schema and really clear instructional goals should produce search results **where every single item returned is useful**. Doesn't that sound nice?

The purpose of most schemas with the express goal of interoperability is to allow everyone throughout the world to be able to successfully find and use knowledge bits. This is very different than your goal. Your goal is that **within your organization** you can find exactly what you want.

This is like computers within organizations. Most of us customize our computers to help us work the way we like to work. We set the desktop a certain color, install screen savers, put shortcuts on the desktop and task bar, even install special hardware and software. For computers in training rooms or other public places, a set desktop is decided upon (usually by MIS) and most people can use those computers…understand where the important functions are located. However, it is not as fast or effective or pleasant to work on a "one size fits all" computer as it is to work on your computer that you have customized to your needs.

Schemas are much the same. In order for a universal schema to be accepted and used, the tags must be developed on the basis of negotiation and compromise. It is important to develop a set of tags that is not too big, as vendors don't want to have to tag everything they make with enormous numbers of tags. The tags must lend themselves to a clear understanding by many different groups from many different countries, industries and organizations. There must be tags that are as useful to a large university as to a steel mill. The schemas must walk a fine line between being too specific and too general. Schemas like the Dublin Core are so broad as to be nearly useless for this purpose. While schemas specifically designed for knowledge bits go a long way to more useful implementation, they still cannot be as detailed as you want.

To reiterate, there is no problem with existing schemas. However, they do not go far enough. A designer building a unique tagging schema for their own organization has a very different motive underlying the creation of the schema. While the creator of a universal meta tag schema wants to give a "seal of approval" to authoring software and vendors of learning objects, the designer wants to be able to make specific searches which return very accurate findings. Universal tagging schemas simply cannot offer that kind of detail, because the detail will be different for every organization. A designer wants to be able to search for the skill set for specific jobs within a department. To do this effectively, the schema must have a controlled vocabulary, only allowing certain words to be entered(see the box on controlled vocabulary below). While this makes the searches very fast and accurate, it is not possible to do this without creating a unique schema.

This paper does not recommend ignoring existing schemas. Certainly, use them. There is no need to reinvent the wheel. In addition, using these schemas allows you to "trade" knowledge bits with other organizations. However, as you create your unique schemas you will weave existing schemas into your own schema, extending them, making them work for you.

**Transition**

This ends the first part of this paper. Hopefully, those reading have a clearer idea of meta tags and how they may be of use to you. It is an exciting time, with new schemas, new applications, new techniques coming out daily. As with any field in its infancy, there are no safe, tried and true ways to proceed. There are few (if any) models for building schemas. In the second part of this paper, a model is proposed. It is a model that supports both the creation of a unique schema and the new way of viewing training - as clusters of knowledge bits. It is learning through building. The ends are a new way of thinking and a unique schema. The means is a specific task. Remember that the task is not the end!

Throughout the development, return to the idea that this is a new and very different way of viewing training. Post signs - "Each piece stands alone!" This is a different way of thinking. Schemas are a different task. There are many beginnings in this model. Take it slow, reflect individually and as a group, think, and converse. Use the process of building to reveal the process of developing a design philosophy of using mind size bits as well as to reveal the process of developing unique schemas.

## Part Two

**Model for developing Unique schema**

This second part is completely involved with your building your first unique schema. This is a process which must be understood by taking action. Reading about creating schemas will be as effective as reading about exercise. Nothing happens until you get down to it and start to sweat. Also like exercise, you will probably get very little positive reward at first - most likely, you will hurt in areas you never knew you had. However, you **will** have a much better understanding once you have gone through the process once. The first time through the process should be a "test drive" if at all possible. There is too much to learn without adding the pressure of a "real" implementation. The second time through is soon enough for that.

- The goal in developing a unique schema for your organization is to codify knowledge bits **precisely for the needs of your organization**.
- The goal is to have a schema which uses no more and no less information than is absolutely necessary for a computer to find and deliver the correct bits of training in a specific situation.
- The goal is to develop a schema which can be **immediately useful** and can also be **useful over time and over the growth of use within the organization.**

This is not a static process. Your schema can grow over time. It can change, if need be. However, as the schema is applied to a larger number of knowledge bits, the added specificity gained from growth of the detail in the schema will not be reflected in previously tagged bits. Changes are even more problematic, as they usually require retagging knowledge bits. Very quickly, the number of tagged bits in your organization make this a task undertaken only when absolutely necessary. So, it is **more important that the elements in your schema are <u>right</u>** than it is for you to build a complete schema.

It is time to begin. First, we list the process. An explanation of each step follows. This is followed by specific directions for you and your team. For some of the steps, examples are included. Read through the process. Then roll up your sleeves, gather your team and give it a try.

**The Process**

1. **Choose a universe**
2. **Choose a training task**
3. **Brainstorm Tag Categories - Major and Minor**
4. **Layout the training task - add tags, set vocabulary**
5. **Build a first prototype - add tags, set vocabulary**
6. **Other Views to spur modification**
    a. **Envision other training tasks**
    b. **Break up an existing training**
    c. **Use your schema to tags some bits**
7. **Review tags**
    a. **What else could be added**
    b. **What could be taken away**
    c. **Result is total schema**
    d. **Look at existing schema - IMS, SCORM, Dublin Core, etc. - to see which tags exist already. ID unique tags.**
    e. **Write a best practices and tagging guide to explain to others your thoughts on what each tag means.**

**Step One Choose a Universe**

Your first task is to decide to how far you want to cast your net. What jobs or skills do you plan to impact - not necessarily right away, but in the future? How much of your organization's training will be provided using knowledge bits? Will, in time, you use this for training internationally? If so, you will need to consider language tags. Will you be applying these trainings to a wide variety of educational backgrounds? If so, some measure of vocabulary or skill level should be included. Will there be time sensitive information? As you can see, the variables get big quickly.

By setting the universe, you are deciding the scope of your project - the scope of it's influence as well as the scope of the work you must accomplish. One's first impulse is to cast as wide a net as possible - meet every eventuality. However, in order to have success, it is essential that you limit your universe. As the number of areas impacted increases, you must either increase the number and detail of tags or reduce the precision of the tags so that they may apply to a greater number of instances. The Dublin Core schema opted to generalize a limited number of tags applied to a broad universe. In this exercise, you do not want to follow that route. Limit your universe and aim for high precision.

You can include tags that you will not use now, but will be used as the universe expands. You can add tags in the future. **However, you must realize that any tags you add in the future will not exist on knowledge bits already brought on line.** Going back over already catalogued bits should be avoided. The greatest expense and delay in using this powerful new method for developing instruction is assessing, cataloging and tagging knowledge bits. Forcing a return because of a change in schema is certain to adversely affect adoption. Think about possible expansion. In some cases, adding a tag in the future to either further refine or expand meta data will be painless. If all current materials are in English and you expand internationally or begin to offer multilingual trainings, adding a language tag would be relatively easy. You can assume that anything not tagged is English. However, if you have a skill level measure from one to five and you started with a universe of manufacturing staff and then expand to include diagnostic staff, the previous skill rating becomes untenable. Remember that this paradigm is not particularly useful in areas that are constantly evolving as the time required to build and tag bits may render them obsolete before they come on line.

During your initial work through, set your universe on the small side. The task of understanding this new paradigm and applying it successfully is a large one. This model anticipates that your thinking will evolve dramatically during the process. This is why you are guided to develop a small system for a limited universe at first. You probably will have to return to the beginning after you build your first training and redefine your universe and rebuild your schema. Doing this after a small training will strengthen your understanding and enhance the likely success and extensibility of your schema.

Anytime a project feels like it is drowning in detail, when the tagging schema is getting impossibly large… you may want to revisit this point. Look at your universe and see if you have defined it correctly. Tagging schemas can get large and detailed. However, briefly returning to this step is a good idea upon occasion during any project.

**How to do it**

- Over several brainstorm sessions, list the jobs (or parts of jobs), locations, skills and/or people which will make up your universe - who will directly use knowledge bits. Make a list for "Certainly Include", one for "Possibly Include" and one for "Will Not Use".
- Make a list of unique characteristics which distinguish the parts of the universe from each other.
- Keep the lists available. As you go through the process, revisit them regularly and edit them.
- Make a decision of the "edges" of the universe. Try and be as clear as possible.

**Example**

Our example will focus on the staff at a nursing facility. For this example, we will define the universe as support staff in all departments - nursing, maintenance, foodservice, security, entry level administrative assistants (secretaries, etc.). We will exclude administrative functions, as they are usually few in number and self trained, so the cost of developing on line training would not be a wise expenditure. However, support staff, medical and non medical have a high turn over, oversight of competency by governmental agencies, and an essential pool of knowledge which does not change very quickly. There are multiple skill levels required (RNs, LVN's, CNAs) for the same tasks. There are different entry knowledge levels. There are several languages. This is an excellent target for training through knowledge bits.

**Step Two Choose a training task**

Remember, the goal of this process is to develop a unique schema for your organization. The training task you pick should impact a large part of the universe you defined in step one. The more types of people, the more different skill levels, the more variables involved, the better the task will be for this exercise. Keep in mind throughout your development that your purpose is to develop as complete a schema as possible. This process **should** stimulate you to think of tags that are not applicable to this specific training task, but will be necessary in other tasks. Always keep a list of **"Other Tags"** available as this task suggests others.

**How to do it**

Pick a real training task. Pick one that affects a broad range of people in different positions if possible. Pick a task can be presented in many different ways - different media, different training approaches, different levels of detail. Lay out the instructional goals for this task. It is best to pick a task which you and your team can create instructional goals in short order without requiring a task analysis, needs analysis or working with a SME. So, pick a task that you and the team knows. Once you start building the training, if the task does not seem to be useful for generating tags, change the task.

**Do not take an already existing training and try to add tags.** The purpose of the exercise is to recreate the situation of developing training materials. You need to recreate the thought processes involved as you consider how to convey information. You will not get this by merely looking at a previously built training. You can use pieces from existing trainings. You should know **what** you will teach, but not have decided **how** to teach.

This is not a cataloging exercise. You are trying to envision tags that will be useful in this and other training situations. Seeing how to apply the schema you develop to an existing piece of training will be an appropriate exercise in the future to test the schema and to train staff who will be cataloging bits. Now, you are trying to invent the schema.

**Example**

We have chosen infection control. This is an important set of knowledge for any staff member who comes in contact with patients, including all levels of nursing, as well as food service, housekeeping, maintenance and entry level administrative. The approaches to infection control are pretty standard. There are numerous training support materials in a variety of media available. The information can be presented as lecture, printed lists, problem based scenarios, or a combination of these and more. This is a great task for developing tags. It is a broad topic with broad application, not a great design challenge, with potential for integrating many different knowledge bits.

**Step Three Brainstorm Tag Categories**

Now that you have selected your universe and your task, you need to begin the process of developing tags. In this step, you will work on brainstorming possible tags and identifying the uppermost level - Super Categories. There are three types of tags you will be developing. All three types are tags. They are nested in a nested hierarchy.

1. Super Categories
2. Sub Categories
3. Detail Tags

In addition, a fourth element **controlled vocabulary** can be applied to tags at any of the three levels. A more detailed explanation of all four elements is contained in step four. For now, your focus is first to brainstorm possible tags without consideration of what level they will be and then to begin to identify those tags which seem to be at the top of the hierarchy, not descriptors of any of the others. (For example, language is usually a Super Category. It does not describe anything else about the bit. On the other hand, under a Super Category of Associations (how a knowledge bit fits in with other bits and machines) you might have Sub Categories of Software (with a controlled vocabulary), Hardware (with Detail Tags for Monitor, RAM, OS and Processor) and a Sub Categories of Other Bits with Detail Tags of Required Bits (if this bit is the caption of a picture, the picture would be a required bit), Recommended Bits and Related Bits. Remember, this is **only an example**. You and your team will develop your own tags!)

At this stage, do not get lost in detail! The most important task is to begin to brainstorm ideas of tags. Don't try to classify them at first. Once you have gone through the brainstorming process and have a large list of possible tags, you will try to identify tags which seem to be Super Categories.

Once you have some Super Categories, you will want to consider what other Super Categories might be useful in distinguishing unique knowledge bits. Remember, there is no right way to do this for libraries or, even more so, for meta tag schemas. You are not trying to find the "correct" way to tag your knowledge bits. You are trying to develop **a useful way**.

**How to do it**

Get several large pieces of paper (or the electronic equivalent). Review the training task and the instructional goals. Your team wants to answer the following question: "If we were sitting down to find knowledge bits about this training in our library of bits, for what would we like to search? How would we distinguish between bits for **this** training and other bits?"

As your team comes up with ideas, write them down. Standard brainstorming rules apply - no editing, no criticizing, everything gets listed, don't worry about duplication. In addition, d not worry about categorizing ideas as Super, Sub, Detail or Vocabulary at this point. On another piece of paper entitled "Knowledge Bits" list any bits that the team thinks might be contained in the training. (This is not the focus of this exercise. It is a side task which may help during step four.

After the brainstorming session, review the possible tags. Now you will try to identify Super Categories. The test of a Super Category is a primary descriptor of a knowledge bit. A Super Category may be a single tag (like Language) or it may have numerous Sub Categories. The important qualifier is that it directly describes the knowledge bit, not a piece of another category. Circle the entries you think are Super Categories. Put each Super Category at the top of a piece of paper and post it.

Now review the Super Categories and think about potential knowledge bits. Do your Super Categories provide enough description? Are more needed? Revisit this task more than once with time in between to give your team a chance to consider different perspectives.

The number of Super Categories will vary greatly. Dublin Core has 15 (Creator, Subject, Description, Publisher, Contributor, Date, Format, Identifier, Source, Language, Relations, Coverage, and Rights). Other schemas have hundreds. Particularly during this first experience, your goal is to think about meta tags and how they apply to your organization, so there is no need to go wild. Keep your categories limited and expand as needed.

Start with theses six Super Categories (five if you are not worried about royalities).

1. Who - Who will need this. This is set more to function title than job title. Job titles change.
2. What -What is this? Descriptors can include media, level, length and purpose. (document, picture, movie, etc)
3. Skill - Why look at this? What skill does it teach?
4. How - type of instruction. Is it a test, a problem based scenario, a list, a document?
5. When - when was it built - so is it current?
6. Cost - who owns it? Does it cost to use? (this may not be a consideration in your organization)

Other Super Categories may include administrative (explicators will include version of evaluation, when the bit was evaluated, who evaluated), language (could also be a subset of #2, What), location (Is it in an online library, on a company server?), associations (Is this part of another bit? Are there bits that must accompany to make sense?) and security.

Every organization will have a different set. You are not making a schema for the world. You are making a schema to tag bits so that **YOU** can find them in a useful way for **only your**

**organization**. As you work on this task, Sub Categories and Detail Tags certainly will be suggested. Include them as they arise. If they do not seem to fit under any existing Super Categories, develop new ones. Work on this at least two sessions. Not only does this task develop Super Categories, it starts the process of considering what tags your organization needs. It starts your group thinking about tags.

> **Example**
>
> After our first brainstorm session, we have the following tags suggested. Language, Skill Level(1 - 5), Media, Length, State Guideline (matches?), Associations, Assessment, Renewal, When Built, Who tagged, Engaging (1-5), Pedagogy, Setting, Hardware requirements, Software requirements.
>
> We assigned the following Super Categories. Language, Job skill, What is it (Length, Guideline, media), Requirements (hardware, software, prerequisites, associations), How (pedagogy, engaging), Tagging (who, schema version, date), Description (text field).

**Step Four Layout the Training Task Continue To Evolve Tags**

You have your Super Categories. You have some Sub Categories. You may even have some detail tags. Now begin to look at the training. Think about it in a different fashion. You are not trying to build the training from beginning to end. List the learning objectives. **Order them only when necessary.** Think of students learning for short periods, so design short bits. Think about the medium of the web - weak on pages of text, good for visual layout, good for graphics, colors and pictures, excellent powers of interactivity - so getting more information, related information and answering questions is easy, easily updated and easily customized. You want to end up with short bits of knowledge.

As this is a "vaporware" project, don't let the usual constraints of time and money hold you back. Think about the best way to convey bits of information. Would a Problem Based Learning module well? Would a movie divided into bits with questions guiding which clip is to be seen next be best? What languages, what skill levels, what mediums? It is not so wild to consider these options. Not only because you don't have to pay for them now, but because there may well be production houses in the future which will make and "rent" modules with far more power than you can produce.

Think about assessment. How will you test completion? Write some test questions. Think about the other parts of a Performance Support Portal - feedback and look up items. Is it important to have training bits that show how and where to find and understand online resources? Are there maps, charts or checklists that should be posted and integrated into the PSP?

**How To Do It**

As you begin to identify instructional goals and strategies for meeting those goals, stop on a regular basis and look at your Super Categories. Ask yourself, "If I had built such a module and knew it was somewhere, on some server…what tags would be necessary to find it, and only it?" If you had dozens of similar knowledge bits, what tags would you want to be able to pull a useful mix of bits from which you could choose? Imagine that you are going to build a training. What do you what to be able to search on? Every option will be a tag. Write it down under the appropriate super category. As Sub Categories or new Super Categories or Detail Tags suggest themselves, add them. If a tag can be limited to a certain few specific answers, write those answers under the tag - this is the beginning of developing a controlled vocabulary (see below).

Each organization will do this task in their own way. A good way to start is with large pieces of paper on the walls. After developing most of the Super Categories and adding many Sub Categories, you may want to transfer the work online. These are difficult ideas to conceptualize, so lists on large pieces of paper are often helpful. After going through the project a few times, you may want to move onto an electronic medium. The danger in the electronic medium is a tendency to focus on one page with one Super Category at a time. It is important that your team maintains a holistic view of the tags and how they represent knowledge bits. Having them all over the walls is useful to maintain this view.

**Pointers** As you go through the process, keep in mind that you are developing three levels of labels.

1. **Super Categories** - over arching general subjects. Sometimes the super category is specific enough. Usually it needs additional descriptors.
2. **Sub Categories** - descriptors of Super Categories. For example, you may have a Super Category for Requirements. You will want Sub Categories to add more refinement to your searches. In this example, you may want Sub Categories for Hardware, Software, Prerequisites, Available Time.
3. **Detail Tags** - descriptors of Sub Categories. Depending on how specific your Sub Categories are, you may have many Detail Tags. In the example above, the Sub Category Hardware might need a Detail Tag for Monitor, Sound Card, Speakers, Internet connection and Processor and the Sub Category Software might include QuickTime (yes/no), OS, and a browser.

**Super Categories, Sub Categories and Detail Tags are All Tags.**

**Controlled Vocabulary**

Many schemas have a problem with interpreting tags. Each person will use different words to describe the same bit. When you try to search, this reduces the accuracy. The best way to overcome this limitation is to employ a controlled vocabulary. By specifying options wherever possible, a rose will always be a rose. Of course, there will be tags which allow a text description without **controlled vocabular**y. However, these tags will usually not be used to search. They will be used once a search has been completed to enable either the designer or the student to make a decision whether or not to use the bit without actually looking through the bit. Whenever possible, specify controlled vocabulary. As you list Super Categories and Sub Categories, begin to list specific words you would like to use. Keep notes on what these words mean. Think about different situations and how different bits should be tagged. You will need to write a tagging guide so that the tags are used as uniformly as possible.

**Step Five - Build a first prototype - Modify Tags**

Now move from conceptualizing the training to actually building. It is very important that you and your team actually build the prototype. If you merely think the process through, you are certain to miss problems with existing tags, new tags that are necessary, and controlled vocabulary that will make searches more accurate. The time spent will also support the new concepts of instructional knowledge bits. This may be the first time your team thinks in terms of independent knowledge bits. That is too important a jump to make by thinking and not doing. Take the time! Go into as much detail as you can afford time. As you build, continue to look at your tags -Super Categories, Sub Categories and Detail Tags. As you add a category or descriptor, decide whether you can have a controlled vocabulary. If you can, make the vocabulary lists.

Take your time. Examine the process of building and see what tags are needed. You and your designers need to have this task in the back of their minds at all times as they develop trainings. A schema is not a static document. It continues to grow and be refined as you add more knowledge bits and cover more applications. You want to develop tags here that will continue to be useful over time as the schema grows.

During this process, you will probably realize that some tags are inappropriate - drop them or change them. In addition, you need to discuss how small a knowledge bit will be useful within your organization. Involve your technical staff in these discussions. It is possible to tag a single picture or a text block. If you will be using animations or pictures with explanations in different languages, you will need to go to this kind of detail. This takes more time to cut down as well as to tag, technical expertise and budgets. Consider how the pieces you build may be used in different trainings. Does this suggest new tags you may want to include?

It is essential that you and your team go through this process. Only by discovering and solving challenges such as the one above will you be able to develop a schema which will meet your specific needs. In addition, the exercise will acclimate your team to the numerous decisions required for implementing a tagging schema. Remember that after you develop the schema, it must be used. People will view bits and need to understand what tags are appropriate. Often the people tagging will not have gone through this development process. As you and your team develop tags, make notes to clarify what each tag means, giving examples. The goal is that the schema will result in uniform tagging even with different taggers.

By the time you have completed the prototype, you should have a pretty solid view of what your schema should include. Review your Super Categories, Sub Categories, Detail Tags and controlled vocabulary items. Consider additions and changes. Document examples and explanations for tags (particularly tags without controlled vocabulary). This is a reiterative process. Continue to revisit your tags, adding, refining, and modifying.

### Step Six Other Views to Spur Modification

Once you have gone through the design process with a fair degree of detail and are satisfied with the tags you have developed, you need a different take on your schema. We suggest three exercises.

1. Pick at least one more training task, significantly different from the first. While your team need not go through the entire process including building a prototype, look closely at the design process. Consider what new tags would be necessary to make this project work. **Continue to consider different projects until you are able to tag a project without adding a tag.**
2. Turn the process around. Pick an existing training available in a digital form (no matter what the final delivery media, such as the file for a print based workbook). Go through the training and see what bits within the training might be useful - text sections, drawings, pictures, videos, sounds, animations. Look at individual bits - as big or as small as you have decided to use.

   > Note: This may cause you to reconsider how small a knowledge bit you wish to use. The smaller the bits you use, the more agility you have, but (as you can see during this exercise) the more work you have. As with everything else, there is no right answer. You try to decide what is most useful for you. This is an area where you can extend tagging in the future…start with bigger knowledge bits and over time, add tags for smaller bits.

   As you look at the process of carving up a training and tagging the bits, see what new tags are needed. You may wish to try a few trainings in different mediums in order to make sure you are ready for as many circumstances as possible.

3. This is a very time consuming, but vital task. Assign a few bits to individuals or very small groups. Have them use the schema you have developed to tag the bits. Very quickly, holes in the tagging schema will appear. As people try to tag, they will see the tags which made sense during group work which are no longer clear. It is essential that each member of the team document tags which seem unclear. The group can decide on a definition or clarification for those tags, but those clarifications must be documented. If not, each time someone uses those tags, they will probably come to a different conclusion. The result will be tags that are not uniform, which defeats the purpose of having unique tags.

During this exercise, your team will see for the first time all the different pieces of the schema. It will often be several pages of text to describe each bit. This is a time to consider all tags. Are they important? Are they necessary? Do not delete tags simply because the list is long. Once the schema is in place, tagging a knowledge bit will not be so arduous. Your programmers will be able to build a simple front end which looks like an online form, with pull down menus and check boxes for controlled vocabulary, radio buttons for yes/no, and text boxes for the few (hopefully) tags which allow open text entry (the description tag). The results of this front end will be able to be appended to the knowledge bit without actually writing all the tags. Make decisions based on utility - is this tag necessary to find this knowledge bit - to distinguish it from all the other bits? If so, keep it. If not, don't.

### Step Seven Review Tags

In the last step, you used the schema to tag knowledge bits. You had to bring all the pieces together. For most groups, this means they are on several pieces of papers with several versions and different explanations. Bring all the different pieces together. List the Super Categories, Sub Categories and Detail Tags. List the controlled vocabulary for all applicable tags. Write an explanation and list examples for each tag. Get the schema together - clear and formalized. Then review the schema with the group. Bring in people who were not involved in the development process and explain the schema to them. Show your schema to others involved in meta tagging and ask for their reviews. Throughout all these reviews, the focus is:

1. What should be added?
2. What can be taken away?
3. What needs to be clarified?

Once the reviews are complete, formalize your schema with tags, explanations and examples. Your goal should be that anyone who uses your schema will uniformly apply tags to bits. The time spent on developing a guide of explanations and examples (often called a best practices document) will reap rewards in accurate tagging in the future.

Congratulations! You have a unique schema for your organization. You should complete one more step before getting the programmers to turn your list into a machine readable document. (Note: The process of building a schema using accepted formats is too large a task for this paper. It is not a terribly difficult task, but one which requires a workshop specifically focused on that skill.)

**Step 8 Look at existing schema to see which tags exist already**

As mentioned before, there are numerous schemas already in existence. It was recommended that you not look closely at those schemas before you developed your own unique schema. It is important to look at your organization and see what tags you need, rather than to see how you can alter your needs to fit existing schemas. However, now that you are clear on your needs, it would be useful to take some time now to review other schemas to see if any of your tags are already defined in other schemas. If a tag in another schema matches exactly or closely one of your tags, it is easy to plug that tag into your schema. There are two advantages to using a tag from a universal schema. The first is that using these tags makes your schema more interoperable and more understandable by others. This is not a high priority, as you are building a schema unique to your organization, but it is a good habit. The second, more important advantage is that, as you bring in knowledge bits from the outside, they will be tagged using one or more of the universal schemas. The more tags you can use without retagging, the less work you have to do.

Of course, as you do more work with tagging, you will become more familiar with other schemas. Much of the work done to create a schema from scratch will not be necessary. You will use existing tags either directly, with modification or simply as guides. However, do not fall into the trap of using a tag that is nearly the same but does not give you an accurate search result.

Remember the difference between universal and unique schemas. Universal schemas have a goal of interoperability. Like buying clothes off the rack, you should be able to walk into any store anywhere and, given your measurements, get a pair of pants that fit. Unique schemas only are aimed at returning accurate searches within a specific organization. They are like going to a tailor and having a suit made to fit. It takes more time and resources, it won't work for anyone else, but it works really well for you.

Look at the Dublin Core (http://purl.oclc.org/docs/core/documents/rec-dces-19990702.htm) which is the most universal schema, but probably too general for use in your circumstance. Many people use some Dublin Core elements (their name for Super Categories) and add tags for details. The IMS model is also worth review. (http://www.imsglobal.org/metadata/mdinfov1p1.html). The SCORM model may prove interesting (www.adlnet.org).

Look at these models. See if any of their tags are as good or better for describing your bits. If you want to use one, adjust your schema. Make a note indicating from which schema you took the tag. The programmers know how to reference those schemas. Then you will use that tag as part of your schema, just as if it were one of the tags you developed on your own.

**Conclusion**

There is much work to be done in the process of making and using schemas. There is even more in the design and deployment of knowledge bits to support performance. However, the need for training which can match the scale and speed of change of the companies of today and tomorrow make these technologies inevitable. As designers, we need to begin to understand and utilize knowledge bits. We need to build trainings as mosaics of bits. We need to begin to utilize existing bits and use the expertise and economy of scale of vendors to utilize library bits.

This first model for creation of a schema is just a step. We need models for creation of trainings with bits and models for repurposing existing training into bits. We are sure to see these and other models in the future.