Managing Social Data: Is SharePoint the

Andrew Chapman

Records management (RM) professionals have been challenged to manage electronic data for some time. Their efforts have tended to focus on unstructured data, such as documents, scanned images, and spreadsheets. Recently, structured data held in databases has also come under close scrutiny, especially during discovery.

However, there is a third category of data that tends to be overlooked. This type of data lies between unstructured and structured data and is a result of the explosion in Web 2.0 technologies (e.g., blogs, wikis, web pages, activity feeds). The temptation is to call this "semi-structured data," but the database world already owns that term; for the sake of clarity we will call it "social data."

What Is Social Data?

Technically, *social data* is structured data that has an associated presentation layer. Confused? Consider a practical example like a blog. A blog entry looks like an unstructured document with paragraphs of text and associated headings. However, if you looked inside the SharePoint server, you'd discover the blog page is actually stored as structured data in the SQL database. Would you consider a blog unstructured as it appears or structured as it is stored? Neither, it is a classic example of social data.

In the world of enterprise content management and records management, you can think of social data as being data that looks like unstructured data to the end user, but the underlying content follows a pre-determined structure. In most cases, the underlying data will actually be stored in a relational database; in other cases (e.g., XML and HTML), the content is encapsulated within the actual object itself.

From a records management perspective, this means you either need to retain both the data and the presentation layer or a standalone rendered copy of the data that encapsulates both the data and the view. Take for example a web page; should you store the HTML file and associated style sheets, or should you render the completed web page to PDF and store the PDF?

What Makes Social Data a Different Use-Case?

The list of social data types you might find in SharePoint includes blogs, blog comments, wiki pages, activity feeds, surveys, tasks, calendar items, web pages, and XML. With the exception of XML, all of these items are actually stored by SharePoint as structured data in SQL server.

As discussed previously, consideration needs to be given with regards to what represents a social data's copy of record, but that's not the only thing that makes social data an important consideration.

Consider that the list of types of social data is long and growing rapidly. More and more companies are using social data tools for critical business applications; the records teams may not be aware of it yet, but it is happening. Blogs are being used to disseminate internal company plans, and wikis are used as a way of collaborating on external-facing documentation. Teams within organizations are collaborating on legal contracts using instant messaging tools.

Typically, access to conventional structured data is well controlled. For example, the enterprise resource planning (ERP) system that creates it will control who can access, update, and delete the content that it generates and can audit the process. Social data systems might generate the same kind of structured data behind the scenes, but the access model around those systems will typically be less regulated. Within a SharePoint deployment, many people might have access not only to update a blog, but to decide who else can update that same blog.

Answer?

Social data tends to be more dynamic than either structured or unstructured content. One person's view of the data may not be the same as another person's view, depending on their individual access rights. Consider a web page that uses Microsoft's enterprise search tool FAST to aggregate content from systems across your organization personalized to you; the web page you see could be significantly different from the page that someone else sees. Which view is the view of record?

To further confuse things, consider that a blog entry in SharePoint can be versioned independently of the comments that relate to it. If you decide that a specific version of a blog entry is to be retained, do you also need to capture the comments from earlier versions of that entry as part of the record?

As you can see, social data will often have more "context" than unstructured data; unstructured content like Word documents can usually stand alone, but frequently, semi-structured content will contain embedded content and links. Just like any other content in SharePoint, the social data will also have associated objects (e.g., ratings, tagging, user-generated comments). Again, when deciding what constitutes a record for your organization, you need to consider all these component parts.

What Operations Apply to Social Data?

Now that it's clear what you are trying to manage and why the management of social data needs special consideration, ponder the levels of control you might need to apply to this content. As usual, you are likely to need to retain immutable copies of some data either for simple retention, for legal holds, or for the formal declaration of a record. You should also seriously consider a formal disposition schedule for social data. For example, treating social data like e-mail for disposition purposes is a good practice.

Immutability

There are two ways to ensure immutability for social data in SharePoint. You can make the object immutable in place, or you can make a copy of the object into a location where it can be protected. In SharePoint 2007, the only real option was the latter. But SharePoint 2010 adds some new functionality that will be considered in the next section.

The option to copy the object to a protected location may not sound too onerous, as mentioned previously. You need to consider what you are going to copy and how you are going to ensure that you do not lose the context and access to the dynamic aspects of the content. This is far from trivial.

Disposition

Disposition is always a good idea – it is often not given as much attention as retention, but it reduces risk, reduces operational costs, and makes discovery a lot less painful. With conventional records, disposition is often the final phase of a retention policy. But in the case of social data, you might see disposition as an operation that is independent of any immutability. For example, you might dispose of any blog comments 12 months after the related project has been completed; in the interim, anyone with the right permissions to the site could also delete them before the scheduled disposition. This is a model you might already use for e-mails.

Another interesting aspect of social data: it is actually structured and relational behind the scenes, so you can do disposition of just parts of the content. For example, you might purge all of a blog entry's comments, but not the actual blog.

How to Do It?

You can either make use of the built-in functionality of SharePoint, or you can develop your own solution. The functionality provided by SharePoint 2007 for social data is absolutely minimal, but Microsoft has added some interesting social data-related functionality in SharePoint 2010. For most medium- to large-scale deployments, it is likely you will have to supplement SharePoint's capabilities regardless of the version that your organization is rolling out.

Consider the factors outlined so far. You have content that:

- Is dynamic in nature. What you see might differ from what others see based on role. What you see one minute might differ from what you see the next based on the content in an external system.
- Contains a lot of context. Web pages contain links, blogs have associated comments, and content might have ratings and tags associated with it.
- Needs to be made immutable "as-is" or as a self-contained snapshot.
- Should be disposed of – both the original content and also any snapshots

Base SharePoint Functionality

Although SharePoint 2007 does not specifically include any social data-specific compliance controls, you can leverage some of SharePoint's standard features to implement limited capabilities. For example, you could develop a workflow and associate it to a blog object type. The workflow would dispose of all related comments after six months, all draft versions after 12 months, and then automatically dispose of the actual blog after 24 months.

You can either make use of the built-in functionality of SharePoint, or you can develop your own solution. The functionality provided by SharePoint 2007 for social data is absolutely minimal, but Microsoft has added some interesting social data-related functionality in SharePoint 2010.

In SharePoint 2007 and SharePoint 2010, implementation of a relatively simple extension allows you to add a "send to" menu item for that object type. This allows you to route a semi-structured object over to another site. Typically, you would not want to route it as-is, because if you routed a wiki over to a records center, you could lose the entire aforementioned context.

A better approach is to route the object over after first "flattening" it into a standalone object. The plug-in code would render the page to a standardized format (e.g., MHTML or PDF) using a given persona and would include any embedded content, comments, tags, ratings, etc. This resultant flat file would then be routed to the records center and managed using the standard records center capabilities – consider that this flat file is a normal, unstructured object now.

There are tons of examples of menu plug-ins online. Therefore, there is a good chance that Microsoft might re-

lease some code samples to illustrate how to perform this function. There is also at least one Microsoft partner with technology that renders wikis to PDF for this purpose.

In SharePoint 2010, Microsoft will be adding a few new features aimed specifically at social data. In SharePoint 2010, you will be able to declare any non-list item as a record; but you can only perform an "in-place" declaration, not a "route to records center" declaration. Why only in-place? As mentioned earlier, social data includes a presentation layer and associated content that constitute part of the record. If you leave the record in-place, you still have access to the associated content, roles, and presentation layer. If you move it to a separate location, you run the risk of losing a lot of this integrity.

In SharePoint 2010, you will be able to declare any non-list item as a record; but you can only perform an "in-place" declaration, not a "route to records center" declaration … As mentioned earlier, social data includes a presentation layer and associated content that constitute part of the record.

So what doesn't this capability give you? The most obvious feature missing in SharePoint 2010 is the lack of any hierarchical immutability. If you make a blog entry immutable, it can still have blog comments added, be tagged, rated, etc. This might be okay for your particular policies; however, consider that any embedded objects (inline images) are also not made immutable, which is probably not the desired result. The same issue exists for disposition. If you dispose of a blog entry, you will need to consider how to dispose of all of the associated content if that's appropriate.

Another consideration for more complex implementations: When performing in-place declaration, SharePoint 2010 allows only one version of an object to be declared as a record at any one time. So, if version 2.0 of your wiki is declared as a record, you can continue to version the wiki (version 2.0 remains immutable). But if you then want to declare version 3.0 as a record, you would have to undeclare version 2.0 first.

Will these features provide everything that your organization needs in order to have a comprehensive compliance strategy for social data? Probably not, but they do provide two things: First, they give you native functionality for some use cases, and second, they implement some features you could build out from.

Third-Party Tools and Custom Solutions

In many cases, custom code will need to be developed to implement your specific requirements. This custom code may live behind a "send to" menu item; but given the volume of social data, it might be an automated process that runs behind the scenes. Consider these two real-life cases of automated declaration of records from SharePoint 2007:

The client had web pages that displayed lists of information drawn from an ERP system. What appeared on the web page depended on the viewer's role in the ERP system and on the content in that system. The requirement was to store these web pages as a record.

The solution was a job that monitored each time a new piece of content was added to the ERP system and then work out which web pages referenced that content. The process then rendered that web page to a PDF file, which was then stored in the ECM system as the copy of record.

However, the client didn't render just one PDF version of the page; the page was rendered 14 times. Why? Within the ERP system, there were 14 different levels of access, and the client needed to store every possible rendered form of the page.

If that seems onerous, then consider one client who wanted to store all views of all of the organization's web pages as records. The client had a network listener running 24x7 that monitored any HTML page request. It then captured a copy of every generated HTML page and stored anything that was unique as a record. This solution is very comprehensive, albeit complex.

What is the lesson behind these cases? If you want to take an all-encompassing approach to storing social data, then you'll need lots of storage, servers, CPU capacity, and a good development team.

What's Makes SharePoint Special?

This article focuses specifically on SharePoint as a platform for two reasons: First, any article with SharePoint in the title typically gets more exposure, and second, out of the box, SharePoint includes support for the most commonly used social data that your organization will use today (e.g., blogs/comments, wikis, activity feeds, calendar items, tasks).

Given that you are managing your organization's un-

structured content, its social data content, and even some of its fully structured content in SharePoint, it seems like an ideal platform to address many of your organization's compliance issues from within a single system.

How You Can Contribute to the Solution

Social data types, usage models, and volume are growing faster than the technology solutions and internal policies that address them. Microsoft has an opportunity to develop some standardized technology solutions to address social data within its unified SharePoint platform. Users should encourage Microsoft and the other vendors to address social data within the appropriate platforms.

One critical component that was apparent from researching this article is the need for some guidance from the records management community regarding the types of operations that need to be applied to social data and what exactly constitutes the copy of a record. Vendors, including Microsoft, can keep adding discrete functionality for managing social data, but they are shooting in the dark if they are not following best practices from the records management community.

The paradigm used in social data to store data and associated presentation layers separately is bleeding over into both structured and unstructured data systems. For example, formats such as Office Open XML and ODF (e.g., Microsoft Word .docx file) are actually compositions of XML files and rendering information. Effectively, these documents are being handled in the same way as social data in which the presentation layer and the data are stored separately.

While this article might seem like another pessimistic commentary adding yet another thing for you to worry about, the RM community is in a much better place today regarding social data than it has been in the past with other compliance considerations. Consider that vendors already see the value in adding compliance around social data into their products, and companies already have many of the underlying frameworks in place to manage this type of data. No doubt that with the right people, processes, and technology, the RM community should be able to absorb social data in its compliance planning when necessary. END

Andrew Chapman can be contacted at chapman.andrew@emc.com. *See his bio on page 43.*