What are ATOM/RSS Feeds?

Series:  Web 2.0 for Lotus, WebSphere Portal and You

Listen to Pete Janzen from IBM Lotus interview various experts from the Lotus and WebSphere Portal development teams on Web 2.0 technologies.  Hear how IBM has utilized these technologies in the various products and how you, the developer, can leverage them in your custom applications.

Abstract:  Greg Mehlan explores the topic of RSS and ATOM feeds which are an important technologies for publishing content in the Web 2.0 world.  Hear how Lotus Quickr has used this technology along with the ATOM Publishing Protocol to provide an interface to Lotus Quickr services.  Greg Mehlan is an architect for Lotus Quickr.

Discussion:

JANSEN:  Welcome to our developerWorks podcast series on Web 2.0 as it applies to various Lotus and WebSphere portal offerings.  My name is Pete Jansen and I am a product manager for Lotus Component Center and Enterprise Integration for the Notes and Domino platform.

I'm your host for this podcast series where I am interviewing a number of our technical thought leaders on topics related to Web 2.0.  Our objective is to enlighten you on the various concepts and technologies that comprise Web 2.0 and how they apply to the products we develop.

In this podcast we will explore the topic of RSS and Atom.  I am very pleased to
welcome Greg Mehlan, who is a senior technical staff member at IBM and a lead architect for Lotus Quickr.

Lotus Quickr 8.0 recently shipped and makes extensive use of both of these technologies. Welcome to our podcast series, Greg.

NELON: Thank you Peter.

JANSEN: So Greg, today we're, as I mentioned, we're talking about RSS and Atom. And I thought we would start off with some of the basics. Can you kind of give us a definition for both RSS and Atom?

MEHLAN: Sure. A great topic. RSS and Atom are just ways of expressing what generically are called feeds. And a feed in a Web sense is a mechanism for distributing frequently updated content.

So for example a feed is really just a document that's fetched using HTTP and it's used for, by example, Web site providers to broadcast news information. It's also used for other kinds of frequently updated content like blogs and podcasts.

There are two sort of main flavors of feeds. One's RSS and one's Atom. And by flavors technically I mean that they're just different ways of expressing a feed in XML.

JANSEN: Right. So there's two things you mentioned there that I'd like to explore a little bit more. One is you mentioned that they're a way of basically presenting information over HTTP. Is that typically the only service that one would need to enable to support this kind of feed generation?
NELON: Typically yes and the idea behind using HTTP it's really the same idea
behind Web 2.0 in general, is we're trying to take advantage of the plumbing that's
already in place which is based on HTTP and make it really easy for people to get
content using that existing plumbing. And feed is just a way of doing that.

JANSEN: Oh, okay, great. Another thing, RSS. I have seen, you know, the
definition for that acronym being Really Simple Syndication and Rich Site Summary. Is
one more appropriate than the other, and why are there kind of two names out there?

MEHLAN: That's a good question. You have to go back to the history of RSS to
understand a little bit more about that. It was introduced in the late nineties by
Netscape, it was popularized by Netscape anyway.

And at the time it was originally standing I think for RDF Site Summary. It's actually
gone through several different generations. And the latest generation is called version
2.0 of RSS, and now it's intended to mean, stand for Really Simple Syndication. So I
think the best term or the best way of describing RSS today is Really Simple
Syndication.

JANSEN: Okay, great. So next I was wondering if you could kind of compare
and contrast both RSS and Atom.

MEHLAN: Sure, and first of all, they both use XML and they both use HTTP as a
mechanism for broadcast. So they share a lot in common conceptually.

Now in terms of the details, if you look at RSS and its history, you find that it evolved
over time starting from a format for broadcasting news sites. And it has become sort of
a de facto standard but it isn't really a standard, meaning it hasn't been adopted by a standards body.

And if you look at Atom in contrast to RSS, Atom originated in 2003. It was really started because there were some issues about the openness and extensibility of RSS that needed to be addressed. And in order to sort of make it a real standard, the standards body was introduced, the INTERFACE, to guide that process.

So there's actually an RFC in the ITS standards groups describing now Atom. And so it really has become a real standard. And Atom was also designed with extension in mind so it was designed to allow you to type content and to be able to broadcast lots of different kinds of content across a feed.

JANSEN: I see. And I think that is one of the real differences between RSS and Atom in the fact that with RSS you can read data but you can't write data.

MEHLAN: Well, that's really an aspect of Atom that was introduced in addition to it being a standards based protocol. It's actually also now a protocol that allows you to turn the line around.

So while RSS was generally intended to be a way of broadcasting or syndicating information Atom also is used, as you say, to publish information into a site. So there's another standard now being developed again through the Internet Engineering Task Force, the INTERFACE, to describe Atom as a publishing protocol.

So you'll hear something called APP which is Atom Publishing Protocol being discussed. And it's a way of doing things like for example updating a blog or publishing a document
basically using Atom as a write as well as a read mechanism.

JANSEN: I see, great. So maybe you could go into a little bit about why these technologies are important to Web 2.0.

MEHLAN: Okay, sure. Another good question. Web 2.0 really the intention is for the Web to become a platform for users to interact with content and not just read it. So essentially in Web 2.0 world, everyone is potentially a content author.

Now feeds provide a useful and very simple abstraction of content. So they provide a way for reducing the friction in the system to be able to get content from many different sources and also to contribute content to those sources.

So with feeds now, you don't need to worry about the details of how the content is stored, whether it's in a relational database or on a file system because you really just view the content and write the content through feeds.

So one way of looking at feeds is it's sort of a way of unlocking the content that was previously held in content silos and making it available to applications using these really basic protocols.

MEHLAN: Okay, so one of the things you just mentioned is for Web 2.0 is that, you know, it really allows end users to interact with the content. And based on your descriptions for RSS and Atom, does that mean that Atom is the more likely feed technology to use in a Web 2.0 app because of the fact that there's also, it has this ability to publish?
MEHLAN: Not necessarily. If you look at RSS it's become very popular for as I said news providers. So there's a huge amount of content that's already out there and publishes RSS. There's hundreds of thousands of such feeds.

So even without the ability to write to those content sources, it's possible to mash in the content which is another part of, you know, Web 2.0's principle, that you should be able to sort of use content as you need to. So RSS is also still an important player in Web 2.0, feeds in general, but RSS in particular is still an important part of that.

JANSEN: Okay. So how has IBM applied RSS and Atom to the various offerings that we provide under the, you know, Lotus portfolio, as well as WebSphere portals?

MEHLAN: Okay, so if you look at the history of WebSphere portal you'll find that it was one of the earliest adopters of these technologies. There's been an RSS portlet for example in portal map for sometime. That's a way of bringing into a portal site content published by RSS providers.

But it's also now being used in a much more general way. It's really being used pervasively in the Lotus technologies. So in Lotus Quickr and Connections you see heavy use of feeds for any time you have to read a list of content.

So for example if you're using Quickr and you're looking at a list of documents, that's available as a feed. And if you're looking at the items in an activity that's available as a feed. So basically anywhere where you see a list of content, that's where you would be looking for a feed.
JANSEN: Does that mean that it's easier to get access to content on different kinds of devices let's say that it might have been in the past?

MEHLAN: Yes, it is. That's really an important part of the feeds as well, is because it's such a simple protocol, it's easy to implement a feed reader on different kinds of devices.

It doesn't require heavy duty application to process the feed. Anything that can read XML, anything that could issue an HTTP request and then read the XML coming back from that is potentially a feed reader.

In the case of Quickr and in the case of Connections, you'll see the ability to get at content in both of those sources through other venues than just the Web as well.

So for example in Quickr, there's a connector that allows you to see documents from Quickr from inside of Lotus Notes or Sametime or the Office environment. And those connectors are using feeds as a way of getting the content and also as we talked about with Atom as a way of publishing content into the system.

JANSEN: Right, yes, and as I look at my Notes client here there's also a new feed section here on the the sidebar which allows me to go ahead and subscribe to various feeds and blogs that I may monitor. So I guess that's another way that we've incorporated that ability to access those feeds into Notes 8. So why would developers want to leverage RSS and Atom in their custom applications?

MEHLAN: That's another good question. Really with feeds it gets a developer access to a much wider variety of content than they would if they were tied to a particular
repository or programming model.

Now as I mentioned before there are hundreds of thousands of feeds available now on the Net that make their content available to consumers and if you want to just read them, there's RSS and Atom for example another conduit for publishing information.

So the basic idea there is that by using feeds in your application it gives your component or the application you're building greater reach.

JANSEN: What you just described is kind of your application making its data available to other applications. So kind of providing a service. Correct?

MEHLAN: Well, Peter that's one way of looking at it. Your application, it should be making its content available as a feed make its content usable by many kinds of other applications or consumers.

JANSEN: Right, okay. So there's not only making your content available, it also, you know, consuming content from other services that are, that have made their content available as feeds.

So I can see, you know, if a developer is building a custom application and there's a service out there that's providing information, they would want to be able to bring that in and present that to their end user in the context of the work that they're doing at that time.

MEHLAN: Right. It's this concept of information on demand that IBM has been describing. And it's basically another way of talking about an SOA architecture. So as
that information or content should be made available as you need it. And having done that now your content can be mashed in to your other application as it's needed.

JANSEN: I see. So you mentioned, you know, mashups that are, content being mashed together a couple of times now. So in a mashup, is the technical way that one of the widgets or components makes the information available to another one via RSS or Atom?

MEHLAN: That's one of the mechanisms. In general with Web 2.0 what you want to do is to make your content available so it could be fetched on demand using HTTP, using a ReST based services.

So that means that with these simple protocols you can do feeds Atom or RSS it's just being a specialized kind of such protocol. Your content can be used easily by different applications.

JANSEN: I see, okay. We actually have another topic in this series of podcasts that's going to address ReST but I was just wondering if you could give us a description of the relationship between RSS and Atom and ReST.

MEHLAN: Well this view that Atom and RSS is just being an example of a ReST style protocol where there's a URL pattern that you can use to fetch and with Atom be able to publish content as well.

JANSEN: Okay. So for developers who are interested in leveraging these technologies, both RSS and Atom, what skills would you say they need to have?
MEHLAN: Well, that's the good news about Atom and RSS, is that you can get started very easily. Since they're just using HTTP as the protocol and XML as the payload, if you have an introduction level knowledge about Web and XML programming, you can get started with feeds.

And another part of this is beyond the basic skills you're free to choose your favorite programming language and editor. So if you have a very simple text editor and you can get started with that. You don't need a development environment to get started.

But, you know, there are frameworks out there and some of them are described on developerWorks, but there are frameworks like Abdera that allow you to build Atom based application without having to understand even the details of that specification.

So there are tools open source frameworks that are available for developers and there's just a lot of shared expertise on the Web on feeds development in general.

JANSEN: Excellent. I see. So there really is a low barrier to entry.

MEHLAN: Yes, and there's a community of, you know, an active community around feeds that has evolved. So a lot of help.

JANSEN: Okay, great. Could you give some examples maybe how developers might utilize RSS and Atom feeds to enhance their development processes? You know, we've talked about how they might implement them in their custom applications, but are there some examples where they, you know, as part of their job to develop solutions they might benefit from this.

MEHLAN: That's another good twist on this, is that, because it's a way of getting
information that changes quickly and because it's a world, in the programming world, things are changing quickly, it provides developers with a way of getting updates on technical information that's pretty focused on their interests.

So for example you can go to the developerWorks site and you can build your own feed where you can say, you know, I'm interested in Java and Linux or podcasts like the one we're building right now. And you can get very targeted information at your own pace.

So rather than sort of cluttering up your e-mail with lots of subscriptions, instead you could build feeds that are targeted toward your interests and then visit those feeds with your favorite feed reader and pull in the content as you need to. So it gives you a way of keeping abreast of rapidly changing technology.

JANSEN: Okay, that was a great point there, the fact that people that do subscribe to, you know, the podcast series on developerWorks will be made aware of this podcast because of the nature of the capability around RSS and Atom feeds.

MEHLAN: Exactly.

JANSEN: So what we were just talking about kind of dovetails nicely into my next question is, you've just gone ahead and released Quickr 8. Now I was just wondering if you could share some of the examples about how Lotus Quickr has leveraged these technologies as part of your 8.0 release.

MEHLAN: Sure, I mean, as I mentioned before really all over feeds with Quickr and Lotus in general, but in Quickr's case, we made a decision to build the connectors, these things that give Quickr reach into your applications, using published feeds.
And so for example when you’re in your desktop and you’re working with Quickr documents and you add a document just by dragging a document into a folder in your desktop what's happening behind the scenes is that Atom is being used as a way of publishing that document into Quickr.

And then the good news here is that we're being very above board in how we’re doing this. So we really, documenting the protocols that we are ourselves are using sort of inviting people to build other kinds of connectors as well.

We also are working to use feeds to sort of marry Quickr with existing enterprise content systems that customers have already deployed. So we'll be using Quickr as a way of getting at content from systems like Filenet by having a sort of feeds wrapper on top of those systems that allow us to see into those systems using this general mechanism.

We're also using it in other ways, like for example if you issue a search in Quickr one of the things you get back is a feed of that of those search results. So this is another way of sort of reusing the plumbing and saying that search results can also be seen as just another example of a feed.

Wherever you see blogs and wikis in Quickr of course that information is also available in the form of feeds. So basically wherever you see lists of content, wherever you see content being published into Quickr, look for feeds as the sort of mechanism for doing that.

JANSEN: So it sounds to me like you've, the development team has realized a lot of efficiency by, or increase their efficiency, by implementing this technology for, as a way to present their content to a bunch of different, you know clients or other
applications, let's say.

MEHLAN: That's another good point. It's really just good engineering that says you should be looking at something like feeds because it's a way of abstracting the definition of content and ways of getting at content.

So it's just good engineering by building on this kind of common protocol you can give your application the ability to reach lots of content without having to build many kinds of point to point solutions.

JANSEN: Great. Okay, I was just wondering if you can kind of share some thoughts about where you see us going here in the future with RSS and Atom in our various Lotus offerings, as well as WebSphere portal.

MEHLAN: Well, I think we can continue, you will continue to see this great use of feeds for syndicating and for publishing content. So we've done a pretty good job for example with Quickr and using feeds as a way of getting and putting documents. And you'll see more of that over time with Quickr and with Connections and Lotus in general.

Wherever you see a list of content, just look for a feed. There more than likely will be one. And whenever you're looking at an API or services to publish content, there will be feeds that will be the likely mechanism for doing that.

You'll see more use, increased use of feeds in portal as well. We will be enhancing the feeds portlet in portal so that it can be, you can use that portlet to pull content in from feed providers that are using Atom, like Quickr and like Connections. So, you'll see more refinement of existing technology and portal that already uses feeds.
JANSEN: Oh, I see. So really ensuring that we have coverage for both RSS and Atom across the breadth of our portfolio to ensure that we do have interoperability certainly across our portfolio, but then extending the reach to other applications that may be build by third party vendors et cetera.

MEHLAN: Right, I mean we’re very interested in sort of getting the partner community and IBM and customer community of IBM also involved and using feeds, extending Lotus products through these feeds approach.

JANSEN: Great. And I think since I’m coming from the app dev tooling perspective, I know that one of our missions going forward to ensure that we have the appropriate level support within our application development tool to easily intersect and consume feeds and services and call ReST based APIs.

So I think we’ll round everything out by providing it both out of the box in our platforms as well, tooling that developers may use to build custom application.

And I think feeds is certainly the right way to go with that. I can remember when I first, when I wrote my first feed reader, it was in 2001 where it was RSS based and it was for Web content system.

At the time it was really quite difficult to find feed providers. And now, as I mentioned, there are almost a million of them I believe at last count on the Web.

So it’s, it seems like it’s the technology that has caught on. And so by having the tools and having the applications or embracing the technology that makes it a lot easier to
build applications that can reach a lot of content.

JANSEN: Excellent. Well thank you very much Greg for coming and sharing your expertise on RSS and Atom today. I think you conveyed a lot of good points about why RSS and Atom are very pertinent to Web 2.0 and how we make use of this technology across the various offerings we have from Lotus, as well as our WebSphere Portal offering.

I believe that developers now understand how they can certainly benefit from using RSS and Atom in their custom applications, as well as helping them throughout their development process as far as a way to stay up to date with current information and to make content that they're working on available to other developers as part of their community.

I would just like to remind our listeners today that this is one in a series of podcasts on Web 2.0 as it applies to our Lotus and WebSphere Portal offerings and invite you to listen to others so that you can expand your knowledge around Web 2.0. Thanks for listening, and have a great day.

[END OF SEGMENT]