



[HTTP://www.dotnetrocks.com](http://www.dotnetrocks.com)



Carl Franklin

Carl Franklin and Richard Campbell interview experts to bring you insights into .NET technology and the state of software development. More than just a dry interview show, we have fun! Original Music! Prizes! Check out what you've been missing!



Richard Campbell

Text Transcript of Show #496
(Transcription services provided by [PWOP Productions](#))



Dean Guo Takes Us Into Project Trident November 5, 2009

Our Sponsors





Geoff Maciolek: The opinions and viewpoints expressed in .NET Rocks! are not necessarily those of its sponsors, or of Microsoft Corporation, its partners, or employees. .NET Rocks! is a production of Franklins.NET, which is solely responsible for its content. Franklins.NET - Training Developers to Work Smarter.

[Music]

Lawrence Ryan: Hey, Rock heads! Beware of falling cliché and listen up! It's time for another stellar episode of .NET Rocks! the Internet audio talk show for .NET developers, with Carl Franklin and Richard Campbell. This is Lawrence Ryan announcing show #496, with guest Dean Guo, recorded live Tuesday, September 1, 2009. .NET Rocks! is brought to you by Franklins.NET - Training Developers to Work Smarter and now offering SharePoint 2007 video training with Sahil Malik on DVD, dnrTV style, order your copy now at www.franklins.net. Support is also provided by Telerik, combining the best in Windows Forms and ASP.NET controls with first class customer service, online at www.telerik.com, and by Grape City Data Dynamics, makers of ActiveReports.Net, simple, powerful and cost-effective reporting for Windows Forms and ASP.NET web applications, online at www.datadynamics.com. Support is also provided by CoDe Magazine, the leading independent magazine for .NET developers, online at www.code-magazine.com. And now, the man who would like to see just one movie trailer that doesn't start with "in a world," Carl Franklin.

Carl Franklin: Thank you very much. This is Carl Franklin. You're listening to .NET Rocks! I am in Malmo, Sweden, which is just across the water from Copenhagen, Denmark, and I'm here at Oredev. It's like 2:00 in the morning here. I'm recording the intro because Richard is asleep. We spent the night in the bar. I had my guitar, we were playing and singing songs and you haven't lived until you've heard American Pie sung my drunk Swedes, let me tell you, folks. But anyway, we don't have any email today or any Better Know a Framework so we're just going to roll the recording. But first, we want to acknowledge that we're going to see you at DevConnections in Vegas in a week or so and we're also going to see you at the PDC, and after that we've got a whole bunch of vacation time, you know, the holidays and all that stuff, and we're looking forward to the launch which is going to happen in March. After the launch, I can't tell you exactly when but we're thinking about doing another road trip. That's right, we're thinking about it so stay tuned for that, you won't want to miss it. We're going to, we're maybe going to go across country and talk to people like we did before in 2005. Stick around, lots of good things happening at .NET Rocks!

Our guest today is Dean Guo. He is a Senior Program Manager for External Research Division of Microsoft Research. External Research division is responsible for broad reach and in-depth engagements with academic and research institutions, related government agencies and industry partners. This division is also responsible for working with internal Microsoft groups to build future technologies and products that will transform computing for scientific and engineering research. Dean has been working with Microsoft for 10 years. Prior to joining the External Research division, he spent 8 years in Microsoft Consulting Services as a .NET application development consultant working with a variety of Microsoft enterprise customers. He specialized in enterprise application integration and web application development in .NET. He moved to Microsoft Research a year-and-a-half ago and has been working on developing solutions to support large scale science projects and reproducible research. Project Trident, a scientific Workflow workbench is an example. Dean has a Ph.D. degree in Medical Informatics from University of Utah. Welcome, Dean.

Dean Guo: Thank you, Carl.

Carl Franklin: It sounds like a lot of fun, whatever it is.

Dean Guo: Yes, yes.

Richard Campbell: And you strike me as a very unusual character because most folks I've ever talked to that are involved with Microsoft Research were scientists that came directly from university into research. You actually spent time as a developer.

Dean Guo: Yeah. I joined Microsoft as an application development consultant. I started in healthcare practice and then I moved to the public sector. So I work on different projects; I gained a lot of practical experiences using .NET. Actually, that was a great advantage moving into research because I have some practical experience and also my passion is research and it has been a great time.

Carl Franklin: Where has it specifically helped you having that experience as a developer?

Dean Guo: As a developer, because I had a chance to work with customers. Because in external research, we are collaborating with scientists from universities, researchers, and also we need to know not only technology but also we need to know processes, we also need to know how researchers do their job. So with this consulting experience I feel like I can talk to the customers in their point of view, and also I can work with developers and design systems to meet their needs. Why write specifications? I can



talk in terms of customers' language and also developers' language.

Carl Franklin: Yeah. It seems like you have your foot in the best of both worlds.

Dean Guo: Yeah, yeah. Yeah, so let me give you some background in terms of scientific computing first and then I will talk about Trident in detail.

Carl Franklin: Sure.

Dean Guo: So in terms of scientific computing, the nature of scientific computing is changing. Data storage, datasets grow faster than more slow. We have sensors deployed on ocean floor. We have digital cameras taking pictures of the sky everyday and there are lots of data generated. So there's an explosion of data coming in for analysis. Typically, if you look at the data management cycle it involves data ingestion, data transformation, and data archival. You collect the data from different sources especially sensors. There's a large amount of data that comes in 7 days a week, 24 hours a day, and there are also different types of data, the land data, water data, ocean data and all kinds of data sources that come from government agencies, research institutions. They all need to be ingested before use, and then after data is ingested we need to do data transformation, have data analysis with different models. You create interactive graphs based on results from different models and then you perform aggregation of results to generate research products, and at the end you also need to do data archival because you want to publish papers, you want to reproduce experiments and all these need some tools to manage. So the scientific community is in great need of scalable solutions for data intensive computing.

Carl Franklin: So it sounds like it's not just the data but the logistics as well.

Dean Guo: Yes, yes. It's the whole cycle of getting the data, doing analysis where you'd put the data you'd run different experiments to generate all kinds of results and then a few days later you're going to publish papers. How did I generate that piece of data? What algorithms did I use? What parameters did I use? There are all kinds of things. You can't manage this manually anymore. Traditionally, people have been using scripts to manage. We call workflows to manage the data collection, data analysis, but the problem is that you--some scientists can do this with computer science background, but most scientists they want to focus on science instead of computer programming.

Richard Campbell: Right.

Carl Franklin: Right.

Dean Guo: Yeah. So that's where Trident comes from. The idea of a Trident is open source workbench for scientific workflow, and so that provides a workflow environment in which scientist can compose, run, and catalog experiments as scientific workflows.

Carl Franklin: So does this also include collaboration? Because I know that you had something to do with iBird or one of those projects that was a real-time collaboration thing. Is that a piece of it?

Dean Guo: Yes, we do have collaboration as part of the feature-set because collaboration, as you know, in the scientific community is very important. Increasingly, a big science project is done across universities, even across countries and collaboration is a critical component. So in Trident, we do have a feature-set that enables scientists to share Workflow, to upload Workflow through a portal and people can download, or even embed Workflow definitions and experiment results in the Word document so that people can share the Word document and annotate, write additional descriptions and everything is self-contained within the Word document for reproducible research and for other purposes.

Richard Campbell: So are we using Windows Workflow here? Would you build in Workflow like this?

Dean Guo: Yes. So we are using Trident on top of Windows Workflow Foundation because we don't want to reinvent the wheel to build another Workflow engine. The value at of Trident is we have a custom designer. The custom designer is totally different from Visual Studio. Traditionally as a developer, if you develop Windows Workflow application you use Visual Studio, you open up a project, you create objects, classes and you reference different libraries. The problem with that is it's very intimidating for a scientist to use.

Richard Campbell: Right.

Dean Guo: And another thing is its not data flow centric. If you develop a Workflow application in Visual Studio, you can drag and drop activities in the designer surface. When you come back to different activities, it's not data centric in the sense that because for a scientific workflow it consist of a series of activities. Each activity has well-defined data inputs and outputs and you can have several inputs and several outputs, and then you want to

chain these activities together so the previous output will become the next input. In Visual Studio, there's no way to do that. So in Trident custom Workflow design, we enable that so that it's natural for scientists to connect different activities together and make a workflow. So Trident workbench provides the library of workflows and libraries of activities and users can develop their own. So that will increase the reusability emitted. For example, you have a well-defined algorithm. You can develop, you can embed that algorithm inside an activity. That activity is a Windows Workflow concept. Think about it as just say a piece of code. It follows Windows Workflow interface definition, and the key method is the execute method. So you put your code inside that method, it does its own thing and then you import it to Trident workbench and anybody can reuse that in different workflows. The idea is we have a catalog of activities, catalog of workflows, you can have domain specific workflows, you can have general purpose activities, domain specific activities ranging from reading data, data analysis, visualizations and so the scientists can just drag and drop and connect them together with minimum programming experience required to do that.

Carl Franklin: Wow.

Richard Campbell: So really, like you said workbench works in the way that scientists think this progression of workflows, of gathered data, correlating a data, and doing reporting on it and so forth.

Dean Guo: Yes. So that's one piece of it. So a scientist can build Workflow and then after you build a Workflow you want to make sure you can run it and you can maintain the history of the Workflow because the reproducibility of research is very important. For example, you do wide experiments with one algorithm and then you want to try another algorithm to try different data input. We automatically capture all this and then we call Provenance. Let me just talk about Provenance a little bit because that's a very essential component in terms of scientific workflow. So the goal of Workflow Provenance is to validate the workflow and to determine the quality of generated data products. What that means is you want to maintain the history, what was done, what data inputs were done, intermediate steps, what data would generate it, what data output would generate it. So when you back to previous experiments, you can trace this is what I did, and if you want to let another research team to validate what you did, Provenance is a very important record for research reproducibility and also for publishing papers that gives you the record.

Carl Franklin: I just want to be clear about the word you said, it's Provenance.

Dean Guo: Yeah.

Carl Franklin: Provenance, right? Like the history of an artifact.

Dean Guo: Yes, yes.

Carl Franklin: Back story.

Richard Campbell: Yeah, think of it as provenance, it's the path of proof.

Carl Franklin: Right.

Dean Guo: Yes, yes. Path of proof, and that's a critical piece. So Trident provides the mechanism to automatically capture the Provenance. Unlike if you develop a workflow system by yourself, you have to manage all this by hand. If you program all this, it has a lot of overhead to change the experiment to try different things. So the reason we can do this is really the Windows Workflow Foundation, because Windows Workflow Foundation is a state machine.

Carl Franklin: Right.

Dean Guo: The engine knows, okay, where do I need to start, which is the activity to start. And when executing a different activity it knows, okay, I'm initializing, I'm executing the error. So it knows the different stage and we have this mechanism of capturing this, and then we have this publish and subscribe service to capture all this Workflow engine generated data because while Workflow services is executing it keeps publishing all these data to this publish/subscribe service and there are all kinds of subscribers. We have monitoring service, we have provenance service. Users can add additional services to the system and it has minimum impact on existing components. So with this overall runtime environment it makes it very flexible to capture provenance data, monitoring data during runtime, error handling, and you can also customize it to fit your own purposes if you want to get additional data or you want to filter out data that you don't want. You can do all kinds of things.

Richard Campbell: I get to think that most of these custom modules are about dealing with different sensor collection methods.

Dean Guo: Yes. Part of it is to deal with data ingestion. You have data from sensors, also data from web services and there are certain standards and different domains. For example, in oceanography community they use less tedious file format and they have different ways, standard ways of retrieving the data from a server, and for astronomy



data it will be different and usually the telescope, those telescopes generate a lot of files and they have to read those files in and filter the data and put the data into a database and do data transformation and do some analysis. So different domains have different needs...

Richard Campbell: Right.

Dean Guo: And different algorithms. For example, in gene alignment area that you have had different algorithms, some algorithms are more accurate than others but they're slower, and you generate the matching scores like you have gene sequence, you want to determine whether they match and then you can generate different matching scores and you try different algorithms and there are different results that's why you try all these things and then you want to generate some outputs and then you can visualize. So in general, in the workflow environment, you want to easily capture the Workflow steps and you can easily run it and the system would generate those provenance records and generate all these records so that you can validate it later.

Richard Campbell: Cool.

Carl Franklin: This portion of .NET Rocks! is brought to you by our good friends at Telerik who brings you the new TFS Work Item Manager and TFS Project Dashboard. So if you're spending a lot of time on organizing the cluttered pile of work items in TFS, get ready for a fresh and intuitive experience. The guys at Telerik just launched the TFS Work Item Manager and Project Dashboard, a couple of free tools designed to make working with Team Foundation Server faster and easier. Unlike the standard TFS Explorer, the Work Item Manager lets you take advantage of powerful capabilities like filtering as you type search, grouping and aggregation, and iteration scheduling. You can even see all the work items in a Scrum dashboard view as if watching the whiteboard in your own room. Project Dashboard is a unique tool for visualizing TFS data useful for both developers and project managers. It helps you keep track of the latest TFS project activity like current iteration progress, build history, recent check-ins, assigned tasks and bug history, and to understand the health of the project as a whole. The TFS tools are brought to you by Telerik and Imaginet, the experts in application lifecycle management. Build with RadControls for WPF, they're both amazingly flexible and responsive. Go to telerik.com and download the TFS tools for free. And hey, don't forget to thank them for supporting .NET Rocks!

Richard Campbell: So I'm wondering if this sort of sensor collection methodology interpretation in sorting a data wouldn't work nicely in the regular business

world too because this sounds like something that a factory would really want.

Dean Guo: Yeah. I mean, this is not limited to a scientific domain. Of course, we start from a scientific domain...

Richard Campbell: Right.

Dean Guo: That is data centric. In other fields, I'm sure anything that is data centric, you know factory assembly, plans, or even in let's say weather forecasting, anything related to data.

Richard Campbell: Right.

Dean Guo: And then you need to do analysis and they need to filter the data, you need to reliably record what's going on and make it easy for user to quickly try different things and that's really important.

Richard Campbell: Yeah, this is going to be terabytes of data in some cases. Like you're talking about astronomy, those images are huge.

Dean Guo: Yeah, so let me give you an example. There's a project called Pan-STARRS and it's developed at the University of Hawaii Institute for Astronomy. So the Pan-STARRS system is able to capture the entire available sky several times each month. Essentially it's a big digital camera. So it generates 2.5 terabytes of data per night, just the raw image data. So in terms of a year, it will be one petabytes of data. So think about that.

Richard Campbell: That's a lot of data.

Dean Guo: That's a lot of data. So the goal of Pan-STARRS is to discover and characterize those approaching objects. Imagine they take a picture of the sky, they compare images so you can identify whether there are any approaching objects like asteroids or comets that pose a danger to our planet. There's a huge volume of images produced by this system so that's why they implement it. Another is the ocean, oceanographer, or study of the ocean. There's a project called NEPTUNE led by the University of Washington. This is a science and education program associated with the study of northeast Pacific Ocean of the Washington and Oregon coast. Sensors are being deployed on the ocean floor to enable real-time data collection. So each sensor you can think of as an IP address. It's really a data source stack. It's collecting data 24 hours a day, 7 days a week. There's a huge amount of data that can be collected to study the global climate change, greenhouse gases, earthquakes, tsunamis, and study life in extreme environment of the sea floor, volcano.

Richard Campbell: Right, yeah.

Dean Guo: A large amount of data and that's the trend, a lot of data being collected from those sensors, from the sky, from satellites, even from web services. How do you deal with that? And that is a great challenge.

Richard Campbell: Well, especially when you're talking round the clock recording like that, most of that is not going to be all that interesting. You're going to need to be able to collate down to the bits that are during the eruption and that kind of stuff.

Dean Guo: Yes, yes, yes. So that involves how you filter the data, how you do the archive, or what data you want to store and there are certain data that can be generated. You don't want to store the final product that you can't use Workflow to generate certain products based on data input.

Carl Franklin: Dean, can you tell me, it sounds like this Workflow application has a potential for a nice plug-in architecture so you could add modules to actually do some things with the data that you're collecting and working on that might be specific to each particular implementation. Is that how it works? Do you use a plug-in architecture? Do you use MEF for example or another...?

Dean Guo: Plug-in architecture in this case as we have this, you can look at two ways. One is at the Workflow level, a Workflow consists of activities. Activity is really a block of code that you implement, either you read it or do data analysis or visualization, and you can embed web services in there. That's like the building block. If you build those activities within the same environment you can connect them together so it's like an integrated set of design in metaphor. You have all these blocks of things that you can build a bigger block. So Workflow can have another embedded Workflow and have another embedded Workflow.

Carl Franklin: Right.

Dean Guo: So that's one level of integration. Another level of integration is we can have different data sources. So we not only work with SQL Server, we also work with, let's say, Cloud data storage.

Carl Franklin: Right.

Dean Guo: Or Amazon S3. We're doing a research to enable, to work against Microsoft's Cloud, Azure data storage.

Carl Franklin: I'm thinking in particular about using the data once you have it, something that would allow you to do an analysis of data and particularly to the business domain that you're working with, or whatever the domain is.

Dean Guo: Yes and then you build a library of this. The build library of this is like a catalog, and then once you catch those you validate those and you put them in these catalogs. So Trident has this registry that maintains all the catalog of activities or workflows and that's become the reusable component. And then coming back to the plug-in model, in Windows Workflow you can add services to the runtime. So you can tell the runtime I have this service, please call during the runtime. That service can be anything. It can be a publication service, it can be a special data handling service and that's in another level of integration that you can plug-in to this runtime. Again, this is really leveraging the existing Windows Workflow infrastructure to integrate different services. In our case, we're using WCF in our publish and subscribe. So the publisher side is really what we call the Workflow Execution Service. So we wrap around this building Windows Workflow Foundation Execution Service. We make it as a WCF service itself and like a Provenance service it is also a WCF service so we can publish the data to the Provenance service through WCF, and then for monitoring it's the same thing. We publish through this WCF to the monitoring service. So that will enable us to do monitoring locally or remotely. Another thing is we have this high performance computing cluster environment.

Carl Franklin: Yeah, my next question is so if you have a number crunching that needs to be done, you sort of ship that out to a farm connecting a little service to give you your results back.

Dean Guo: Yeah.

Carl Franklin: Yeah.

Dean Guo: So we can leverage Microsoft's high performance computing clusters to do really a parallel workflow executions and the user experience is the same. So you specify where you want to execute, either at the desktop computer name or the cluster name because for high performance computer cluster, as you know there is a head node or the compute nodes. You just tell the execution engine, okay, here's the cluster, here's the head node and these are the workflows that we want to run, and the system will use SPC scheduler to schedule those workflows and then run them across different compute nodes. So it's a highly scalable platform. It can do desktop computation scale up to the cluster environment.

Richard Campbell: Well, what's cool about that architecture then is that a university can have a computing cluster and you could easily share time on this between lots of different projects all off the same workbench.

Dean Guo: Yeah, that's exactly true. That's really the idea to make it seamless. It works the same way either on a desktop or a computer cluster environment. Research is being done to enable similar things on the Cloud, so the Cloud environment is even more flexible. The Cloud infrastructure will hide a lot of complexity from users if the computing demand is very high and will allocate more virtual machines to process the data.

Richard Campbell: Right.

Dean Guo: If it's low, then we'll use the minimum resource needed.

Carl Franklin: Hey, I just want to give a shout out real quick to our friends at Data Dynamics who make ActiveReports.NET among other really awesome things. ActiveReports.NET is great because it allows you to just build your reports with the Easy Editor, embed them right in your application, provide PDF and HTML output, give your end-users a Report Editor, royalty free of course, a great Access report upsizing Wizard and all this for a price that isn't going to break the bank. ActiveReports.NET from Data Dynamics, go check it out now at datadynamics.com.

Richard Campbell: I think about guys I've worked with in universities. The amount of time it takes to load up a workload onto a super computer to then run it for two or three hours and then get all the results back off before the next guy can come in just being able to pare that down so that you have as little downtime on those compute clusters as possible.

Dean Guo: Yeah. So that's a big challenge, that's a great question because with large amount of data, moving the data becomes an issue. So the solution for that is we try to move the computing resources closer to the data. Instead of moving the data around, we try to keep the data where that is and then we're going to move the actual workflow, the actual computation close to the machine. It is still a big challenge, it's a great research topic in terms of how to solve that problem. Another way of doing that is we have a product called Dryad, Dryad LINQ. Think about it as the map reduce mechanism so you can divide big files into smaller chunks.

Richard Campbell: Right.

Dean Guo: And you cluster them independently and then you consolidate the results back to the other. That's another. You can use Workflow to do that as well.

Carl Franklin: When you're dealing with these big sets of data, are there any technologies as a programmer that you have to rule out other than the obvious ones. You're not going to load up an entire set of data in memory and use LINQ, but what about LINQ to SQL or some sort of entities? Are there any technologies that standard developers would normally grab for that you couldn't because of the size of data?

Dean Guo: Yeah. So obviously you don't want to load all the data into memory, and currently what I know of is we generally divide them up into different pieces and then you only pass the data you need. So the data ingestion part, the part I read the data in, is the tricky part, how you make sure that you only read the data that you need, and also you don't want to overload the system.

Carl Franklin: Yeah.

Dean Guo: Of course you can use files and move things around, but you don't want to physically move the data. You can use references, you can use references to reference the datasets. That's how you keep the catalog of datasets and so you avoid the copies of data along the way. So there are all kinds of techniques to avoid this data copying, loading large amount of data into memory and things like that.

Richard Campbell: You mention Dryad. You know, about this time last year we did a show on Dryad with Michael Isard.

Carl Franklin: Right.

Dean Guo: Oh yes.

Richard Campbell: At that time Dryad was not publicly available, but it is now?

Dean Guo: Well, yes. You can download a binary from research download. If you go to bing.com, you type Dryad/Dryad LINQ in the download site so you can download the binary free of charge. You can try it.

Richard Campbell: That's cool, and for folks who don't remember what Dryad is about, maybe you can talk a little bit about it because there's a LINQ provider for Dryad as well.

Dean Guo: Yes, the LINQ provider makes it much easier to create the data because the Dryad itself is in managed code. Dryad LINQ provides a

wrapper to make it really easy to manipulate the data through the .NET Framework.

Richard Campbell: Yeah. I keep seeing these situations where an experienced .NET programmer would really help out here because they know the libraries that are available and how to take advantage of them much more than some who are working purely in the scientific community would.

Dean Guo: Yeah, yeah, yeah. So there's definitely an integration point in Dryad and Trident. That's one of the things that I'm exploring right now, how we can leverage a Dryad capability in dealing with large datasets and then make it as a Workflow activity within Trident so that it can deal with the large datasets.

Carl Franklin: What do you use for visualization, for data visualization using third party tools or using graphs and charts from Dundas or something, reporting services?

Dean Guo: Visualization, we have this framework, it really depends on the activity writers. So out of the box we provide some basic charting visualization like a Dundas chart, but that is not enough. A lot of visualization in scientific community has 3Ds, it has images so that requires the different plug-ins. For example, you can have a custom visualization tool that can read data output from the Workflow. So the workbench doesn't -- you can use a third party visualization to visualize data. There's no restriction on that, but out of the box we don't really provide a kind of a domain specific visualization tool yet because that would take some time to make the library comprehensive.

Carl Franklin: But you do offer plug-in points though, right?

Dean Guo: Yes, yes. So you have plug-ins, the visualization component in the workflow.

Carl Franklin: Right.

Dean Guo: Depending on the type of data that you want to visualize. You could use Excel to visualize data and somebody wants to use chart, and somebody wants to use a totally different 3D imaging display.

Carl Franklin: Right. Excel you mention there before. We didn't quite hear what you said. I think you said somebody could use Excel if they want to.

Dean Guo: Yeah.

Carl Franklin: Yeah.

Richard Campbell: And I get to think scientists all live in Excel.

Carl Franklin: Sure.

Richard Campbell: That's the most familiar paradigm you could ask for.

Dean Guo: Yeah. So you can plug in different pieces. That's the beauty of the Workflow. So you have activities that wrap around different technologies either web services, or .NET classes, or even COMController and you can connect them together and then build the Workflow.

Carl Franklin: So is NEPTUNE like the sort of the project that you point to; to say, "Look, this is how Trident is really being used in the real world?" It seems like there's a lot of buzz on the internet about NEPTUNE and Trident. You know, when I'm searching for Trident, NEPTUNE comes up over and over again.

Dean Guo: Yes, yes. We're collaborating with the University of Washington and we feel like the oceanographers, the Trident workbench will be a great tool for the NEPTUNE project. Initially Trident, the idea started with oceanography community.

Carl Franklin: Yeah.

Dean Guo: You know Jim Gray? He is a famous database fellow in Microsoft and he was the first person who had the vision of an oceanographer's workbench. So he actually originally started this and then my manager, Roger Boje, he worked with Jim Gray and they started this workbench idea. So oceanography really was the first scientific discipline.

Carl Franklin: I was going to say maybe you could apply that to geology if you could put sensors all over in the ground to study seismic activity for example.

Dean Guo: Yes, yes, exactly. The pattern exists for lots of domains, even biology. You do gene alignment in mapping and you can do weather forecasting. It's everywhere.

Carl Franklin: Chemical analysis of water, I mean the list goes on and on.

Dean Guo: Yes, yes.

Richard Campbell: Yeah. I just don't see this is that much of a jump...

Carl Franklin: No.



Richard Campbell: From there to quality assurance controls on a factory. Like it seems very similar to me that you could build up a Workflow package around a factory process and do the same kind of analysis.

Carl Franklin: And not just one factory; but with factories all over the enterprise I could imagine.

Richard Campbell: Sure.

Dean Guo: So Workflow is kind of increasingly becoming an important vehicle for this integration, reusability, ease of configuration, and so it's natural for people to build something with minimum programming experience. It's a great tool for analyst to quickly come up with a workflow and then it can be executed and deployed.

Richard Campbell: You know, the whole Provenance concept -- I did a lot of work in electronic discovery on the legal side and they had this concept of chain of evidence that as things move around you need very clear records of how things were handled and so forth. It just reminds me of that same process.

Carl Franklin: Even in the financial system, in the financial software, or in medical software. I mean, anything where data changes, there has to be a reason, there has to be a story behind it.

Dean Guo: Yeah and if somebody challenges that you have to provide proof.

Richard Campbell: Right.

Dean Guo: And there are a lot of complexities that we see during the development because to make sure we can always go back to the original Workflow, we have to maintain different versions of the same Workflow.

Richard Campbell: Yeah. When you upgrade software.

Dean Guo: Yeah. We can't delete a Workflow easily. We have to maintain that because there are data products related to that. We want to be able to go back to that experiment and we produce what the Workflow looks like at that time. So that means we maintain versionings, versions of different Workflow.

Richard Campbell: Right.

Carl Franklin: What's next for the project? Do you have a new version in the works, and are there any particular new technologies that you're working with?

Dean Guo: Well, there are different things that we're working on. Obviously there's a lot of interest to enable Trident on the Cloud.

Carl Franklin: Right.

Dean Guo: That means there are two levels of leveraging the Cloud. One is to leverage the Cloud storage, another is to actually run Workflow off the Cloud.

Richard Campbell: Right.

Dean Guo: The second thing is we're working with Windows Workflow Product Team to migrate it to Windows Workflow 4.0 which is part of the .NET 4.0, and the 4.0 is faster and it has a closer modeling to the data centric mechanism. So the 4.0 supports data parameters, data input and output versus 3.5, that is the version we're using, and really it doesn't have this concept of activity, input parameters and output parameters. We need all dependency properties. There's no input and output concept so we have to model everything ourselves for this. With the newer version, we can leverage a built-in model and make it so we can reduce the codebase dramatically.

Richard Campbell: Cool.

Carl Franklin: Now is the source code available for developers who want to look at it?

Dean Guo: Apparently the binary is on the internet for free download. We're working on open source license. So the goal is to make the source code available to the developer community, to the scientific community so that people can in a way consult the existing codebase.

Carl Franklin: Excellent.

Dean Guo: Yes. So this year the goal is to build a community. I need your help to build a dynamic, a true community and it's not an easy job.

Carl Franklin: All right. Well, you know we have a lot of scientific minds who are listening to the show so go for it, people. It sounds like a very, very promising project. Thank you, Dean.

Richard Campbell: I've shrinksterized the Trident site where the CTP can be downloaded from. It's at shrinkster.com/19d4.

Carl Franklin: Thanks, Dean.

Dean Guo: Thank you, Carl and Richard.



Carl Franklin: Thanks very much for joining us today. And we'll see you next time on .NET Rocks!

[Music]

Carl Franklin: .NET Rocks! is recorded and produced by PWOP Productions, providing professional audio, audio mastering, video, post production, and podcasting services, online at www.pwop.com. .NET Rocks! is a production of Franklins.NET, training developers to work smarter and offering custom onsite classes in Microsoft development technology with expert developers, online at www.franklins.net. For more .NET Rocks! episodes and to subscribe to the podcast feeds, go to our website at www.dotnetrocks.com.