Chair’s Corner

It’s hard to believe that November is already here. It seems like we just started planning our PDS in Santa Fe and now it is over. By all accounts it was a big success. We had 77 legislative staff representing 27 states in attendance—104 total counting the private sector attendees. I want to thank the New Mexico IT staff for hosting us and for the tour of the state Capitol. I also need to thank the PDS planning committee for putting together a great agenda, all of our presenters for stepping up and doing a great job and of course Pam and Stacy for all of their hard work keeping the PDS running smoothly.

We missed those of you who were unable to attend and hope that we see you at our next conference. Handouts and presentations from the PDS are located on the NALIT website.

I know it is a busy time of year for all of us. An election time always causes more work, and most of us will be going into session in a couple of months. My hope is that you all are able to take time for yourselves and enjoy the holidays. Please let me know if you have any suggestions for NALIT for this upcoming year. I am looking forward to serving as chair and will do whatever I can to make it another successful year.

THE SESSIONS WERE FUN, INFORMATIVE AND VERY WELL PRESENTED.
With the trend continuing toward paperless legislatures, information technologists need to prepare for the challenges of authenticating and preserving electronic, legislative legal materials. The technology we use today to draft, present and store legislative material will become obsolete and unusable in its current electronic format if we do not plan for its long-term trustworthiness and preservation.

Several organizations, including the American Association of Law Libraries and the Uniform Law Commission, have been working on addressing these issues.

In July 2011, the Uniform Law Commission approved The Uniform Electronic Legal Material Act (UELMA), and in February 2012, the act was approved by the American Bar Association.

UELMA establishes an outcome-based, technology-neutral framework for providing online legal material with the same level of trustworthiness traditionally provided by publication in a law book.

The act requires that official electronic legal material be: (1) authenticated, by providing a method to determine that it is unaltered; (2) preserved, either in electronic or print form; and (3) accessible, for use by the public on a permanent basis. UELMA has been enacted in several states and an updated list of current enactments can be found at the Uniform Law Commission or NCSL websites. The outcome-based framework allows each state the flexibility to choose its own technology for authentication and preserving materials and allows for future technologies as the current technologies become obsolete.

A discussion of the different technologies and the associated cost can be found on the Minnesota Historical Society Preserving State Government Digital Information website.

California uses hash values for basic authentication, but uses Adobe CDS to sign documents digitally. These documents can be retrieved from the California Legislative Information website.

Both states continue to review their preservation options. For more information, see session resources for “Ensuring the Authenticity of Online Legal Information” from the 2014 Legislative Summit.
Customer Support at the California Legislative Data Center

Ellen Donati, Manager, Customer Support Division, Office of the Legislative Counsel, California

I work in the customer support division of the California Legislative Data Center. I’ve been there for almost 14 years. Before that, I worked at AAA. I’ve trained staff in customer service skills for many years, and I learned that customers are the most important part of any business. I am fortunate that the culture here at the Data Center is customer service-oriented and expects everyone to treat each other with respect and to put the customers’ needs first.

Customer service is my favorite subject. I love hearing from customers and solving their problems. I find customer service to be one of the most important aspects of the service center and desktop support services. Our customers are often under the gun and need immediate attention when they have a problem. If we deliver exceptional service, we are more apt to develop trust and get computers up and running quickly.

We brand our services, for example, by answering the phone the same way every time. We identify our group and our individual names and ask how we can help the person.

Listen Before Providing a Solution

We can’t provide solutions if we don’t know what might be causing the problem. Listening includes asking probing questions, clarifying what the customer tells us and verifying information. We teach never to have dead air. We want technicians to chat with customers while their PCs are rebooting. This way we build a rapport with our customers.

A Pleasant Finish

We always finish up by asking customers if there is anything else we can help with or by telling them what to expect if additional service is needed.

Follow Up

Last but not least, we email all customers once the work is completed to make sure they are satisfied with the solutions. For example, one of our senior technicians ends her emails with, “have a groovy day.” Our customers love it and appreciate the attention they receive.

All of this happens with consistent training, coaching, monitoring and reward programs. We use customer surveys to identify training needs and motivate the staff. We have fun, and we celebrate successes.
2014 NALIT Professional Development Seminar
Santa Fe, New Mexico

How many legislative IT professionals does it take to make learning fun?

104, if attendance at the 2014 PDS is any indication!

Attendees toured New Mexico’s unique Capitol.

Networking and sharing ideas about new technology between sessions are NALIT Chair Joel Redding (Ky.) and Mark Guillen, Host State Chair (N.M.).
Why Attend a NALIT Conference

Erica Peharda, Associate Application Developer, Legislative Council, Colorado General Assembly

I’m what some may call a “NALIT-newbie.” Actually, I just made up that term, but it does describe how I felt going into this PDS. I had no idea what to expect besides hearing from coworkers that, “it’s such a great experience; you’ll learn so many things and gain a lot of new ideas.”

I can now confidently say that what they said is true. Being a relatively new staff member of the Colorado legislature, I was afraid that I wouldn’t understand many of the presentations or that I would not be able to contribute. I could not have been more wrong.

All the sessions were well done, and the presenters were very good at breaking things down in a way that would allow most people to follow their discussions.

The conference began on Tuesday with some demos and exhibits from states that set up tables to showcase their newest projects and explain their struggles, successes and future enhancements. There were ideas that I would not have even imagined being showcased.

As a programmer for the legislature, I have found that many users do not know what they want. They struggle to find ideas, mostly because they do not know what the IT staff can build. Coming to this conference helped me address that area immensely. I was given so many ideas, including, but not limited to, help-desk solutions, committee notifications, e-filing and legislative drafting solutions, iPad applications and floor activity reports.

On Wednesday, we attended a talk on cyber security. Although I do not work much on the network administration side, this was still enlightening. I learned a bit more about what our network administration staff does on a daily basis and the types of roadblocks they face.

“Five Minutes of Fame” was a very fun presentation. Volunteers were asked to explain what they have been working on for the past year in just five minutes. It was a struggle for many states to condense one full year’s worth of technology into just five minutes. This was a great time to listen for any projects of interest you might want to follow up on later.

Connections like this were made throughout the conference.

I spoke a little bit about our state’s Granicus integration, and I was surprised by how many people later came up to me with ideas or questions. Although I was initially nervous about presenting during my first year, afterwards I realized how beneficial it was to me and others. I am also extremely thankful to other people who volunteered to present for the same reason.

The remaining part of the conference was equally impressive. I e-mailed Virginia afterwards to learn more about their “Leveraging the Cloud” presentation.

The connections made with other legislative staff are invaluable. I left this conference with an immense feeling of pride in working for the legislature. I felt like I was part of a larger team, which I imagine is much different than a private-sector job where competition is very high. Legislative IT staffs across the states are ready and willing to both receive and give ideas to each other.

Manish Jani, my supervisor, described it best when we were discussing this article by saying, “it’s like drinking from a fire hose.” My mind was swimming with ideas for creating new applications and improving our existing ones.

For anyone who has not yet attended one of these conferences, I highly suggest doing so the first opportunity you have. You will not be disappointed.
Re-engineering a product on a new platform has benefits and disadvantages. Being able to leverage new capabilities is a plus, but it’s not always possible to carry over previous strengths. In the West Virginia Legislature, we’re trying to update our existing, paperless bill drafting system, itself enmeshed in older technologies, to a current programming architecture—a client-side click-once application in C# and Windows Presentation foundation (WPF).

Some advantages from the development side of the new project are using a model-view/view-model architecture, which allows definite separation of user interface (UI) operation and core program function. This way, the program can focus on being a program, and the UI can be imbued with enough intelligence to automate its own appearance, which is handy. That said, there are quite a few helpful things that can be done in WPF—like custom templates for items in a list that permit the presentation of an object to be defined apart from the definition of the object itself. Additionally, using Microsoft Entity Framework allows the developer to concentrate on everything else in the program and not have to care about the T-SQL composition going on in the background, as well as all the security gains to be had by using EF as the interface to the database. Additionally, Visual Studio has standard features that almost effortlessly work into the project in development, like XML comments and auto-update upon deployment, which greatly assist us in taking care of the time-consuming little things that add up in the long run.

Being a .NET project means we are able to leverage all the support from Microsoft and from other state legislative IT departments that working in this environment can bring about. This helps in situations when we don’t have the time to figure out the nitty-gritty solutions ourselves. Also, being part of the Microsoft ecosystem allows interoperability with other Microsoft products, like Exchange and Active Directory, making it possible to add in more functions than the previous project was capable of achieving. From a management perspective, these new capabilities present an exciting set of possibilities for better cohesion with existing services and products alongside the projects’ deployment environment.

For the end user, this has been a great opportunity to approach the scope creep of the previous project with the expectation of future scope creep, and in allowing for that, to create a user experience (UX) that handles the previous requirements, picks up the needs of the day, and is built in a way that allows future expansion. This creates a clean and simple user experience from the outset and maintains that simplicity despite complications that future requirements may bring.

From a management perspective, this allows that same philosophy—accounting for the future just as much as the present—and adds years to the project’s lifespan. It also mitigates costs associated with major changes and avoids outgrowing the capabilities that are available to leverage in the new project. Using newer technologies and practices allows for better innate security, strengthening the software environment by upgrading a potentially weaker link. Gathering and developing the code in a centralized, modular way provides a lower entrance threshold for anyone associated with the project and easier maintainability in the future.

All that said, there have been some difficulties along the way. Some strengths of the older system may not
necessarily translate over to the newer one, like the robust, innate, reporting engine freely available in the older system. Also, as a new project using newer technologies, a certain amount of time is necessary to acquaint the personnel involved in the project with the newer technologies. For instance, WPF is extensible and powerful, which means that there’s a potential for pieces of the UI to be interchanged and merged in powerful and intuitive ways that may not be clearly evident while in the middle of a development cycle. Incorporating these benefits into the project along the way has the potential to cost time. It is also the nature of a project like this—given that the original project is still in use and being extended while its replacement is in development—to have some additional costs associated with such a development dynamic.

**Pros**
- MVVM architecture
- Entity Framework for SQL database integration
- WPF, extensible UI elements
- Optimizations wrought by Visual Studio (auto-updates, XML comments, etc.)
- C# and WPF interoperability with Exchange, Office, AD, etc. (developer, administrator)
- .NET support, alongside other states’ .NET usage (developer)
- UX/UI overhaul (user)
- Opportunity for major design choices (manager, administrator, developer)

**Cons**
- Deficient, innate, non-WinForms reporting engine
- Steep learning curve
- Shifting requirements (product in use, so it changes while in development)

---

**Call for Newsletter Articles**

The NALIT Newsletter would not be possible without articles from our members. If you have ideas for articles—short or long—please contact Linda Wettstone, Virginia Senate, NALIT Secretary and Newsletter Editor, at lwettstone@senate.virginia.gov.