

La programación gráfica impulsa el diseño intuitivo

Artículo cedido por National Instruments



www.ni.com

Autor: Jeff Kodosky, Co-fundador y socio del negocio y tecnología de NI

La visualización del diseño es esencial para que los diseñadores puedan evolucionar desde el concepto hasta la creación; ya que, puede proporcionar una comprensión más profunda de un proyecto, ayudar a los demás en la comunicación de un diseño y ayudar a identificar los problemas a tiempo. Las visualizaciones de los diseñadores deben abarcar todos los detalles para llevar a cabo con éxito el acabado de los diseños, sean o no las herramientas de los diseñadores quienes les ayuden a llegar allí.

El método tradicional consiste en plasmar un diseño mediante documentos y cifras detalladas. Por ejemplo, para especificar un objeto que se hará en una fresadora, un diseñador puede utilizar varias vistas ortogonales del objeto. Esto es eficaz para la anotación y dimensionado, pero puede ser difícil la integración mental de estos puntos de vista para obtener una visualización del objeto.

Los diseñadores también pueden visualizar un objeto mediante un entorno de diseño interactivo. Resulta mucho más productiva la edición directa de un modelo en 3D cuando se diseña un objeto. La visualización no se limita a las proyecciones discretas y los diseñadores pueden ver fácilmente cómo una edición en un modelo 3D afecta a diversas proyecciones.

En general, la visualización es una parte crucial del diseño, pero visualizar el diseño de sistemas que unen la cibernética con el mundo físico (también conocidos como sistemas ciber-físicos) plantea un reto particular debido a las funciones críticas que

juegan el software y las comunicaciones. El diseño del sistema físico puede ser sencillo, pero la parte cibernética requiere herramientas especializadas para la visualización del diseño del software. La visualización del diseño del software es muy difícil debido a su carácter abstracto, flexibilidad ilimitada y gran complejidad. Una parte importante del diseño de sistemas ciber-físicos está en el software. Como resultado, una mejora en el software de diseño y en el desarrollo de la productividad puede tener el mayor impacto en un proyecto.

Esa es la razón por la que inventamos LabVIEW. LabVIEW es un entorno para la creación y edición de la visualización del diseño del software. Utiliza diagramas gráficos del flujo de datos y paneles frontales interactivos para crear una jerarquía de instrumentos modulares definidos por software. LabVIEW es altamente productivo porque la visualización del diseño es también la implementación.

El código fuente de LabVIEW consta de diagramas gráficos de flujo de datos que muestran claramente las relaciones de dependencia de los datos. Un cable muestra la fuente del valor de los datos y los destinos que los utilizan. No hay ninguna posibilidad de acceso invisible o modificaciones de los valores en un cable antes de que lleguen a los destinos. Los bucles y otras estructuras de control son cajas en el diagrama que separan claramente el interior del exterior. No hay manera de que un error de sintaxis pueda mover inadvertidamente el límite de un bucle.

Gracias a LabVIEW, la visualización es la implementación. Esto lo convierte en una excelente herramienta para la creación rápida de prototipos y el desarrollo incremental. La facilidad con la que los diseñadores pueden cambiar las visualizaciones implica que pueden explorar de manera más efectiva los espacios del diseño de sus aplicaciones, lo que se traduce en soluciones más óptimas y tiempos más cortos de lanzamiento al mercado. La semántica por valor del flujo de datos garantiza la seguridad y la escalabilidad de los diseños. La composición jerárquica es uniforme en todos los niveles y también lo es el comportamiento de la ejecución. Los paneles frontales por módulos facilitan la depuración, las pruebas unitarias y la interacción del usuario en todos los niveles del diseño. Todos estos atributos y otros más, contribuyen a la productividad de la utilización de LabVIEW.

Uno de los atributos más importantes que distancian LabVIEW de otras herramientas de software es que el flujo de datos es inherentemente paralelo. Los lenguajes de programación tradicionales son secuenciales, están modelados de la misma forma en que trabajan los ordenadores, como las máquinas de Von Neumann. Simplemente, los lenguajes secuenciales no son adecuados para las máquinas que trabajan en paralelo. Este es un obstáculo importante en tanto que el mercado se mueve hacia las máquinas multinúcleo y FPGAs. Ya en 1977, John Backus señaló las limitaciones de la arquitectura de Von Neumann, no sólo como un cuello de botella de datos, sino también como un cuello de botella intelectual.

Treinta y cinco años después, los lenguajes secuenciales tradicionales basados en texto continúan perpetuando este cuello de botella intelectual.

El lenguaje de flujo de datos de LabVIEW trasciende a esto, lo cual lo hace ideal para afrontar los retos de diseño que los sistemas ciber-físicos y recientemente, el "Internet de las

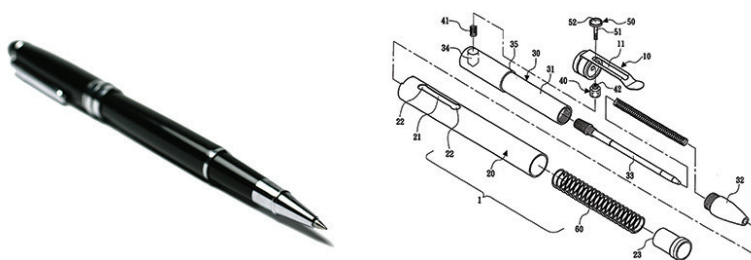


Figura 1. Visualización del usuario con respecto a la visualización del diseñador.

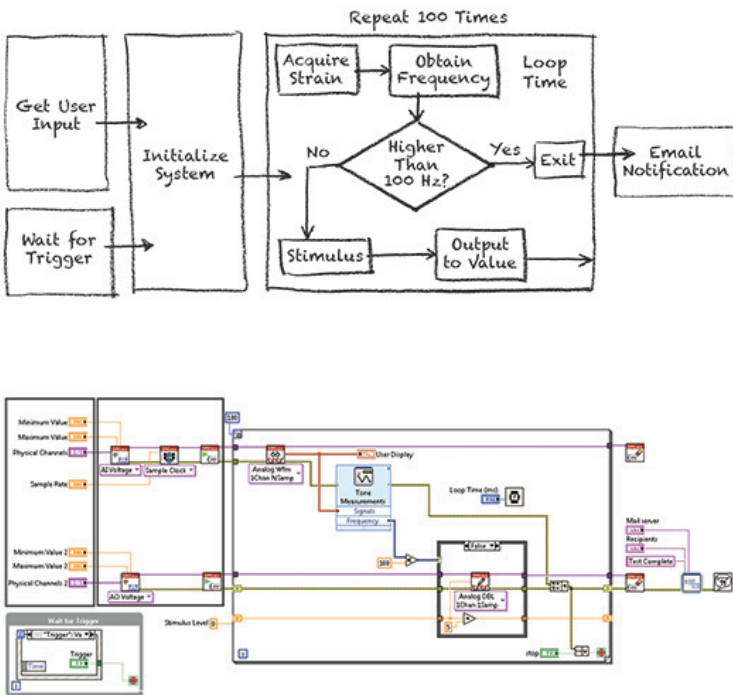


Figura 2. Los diagramas gráficos de flujo de datos sirven para la visualización del diseño y la implementación.

cosas”, plantean hoy en día. Este método gráfico sigue evolucionando gracias a las muchas áreas de investigación activas centradas en la visualización del software y las capacidades subyacentes del hardware.

Una de las áreas en curso de investigación es el tiempo. La mayoría de las aplicaciones de software están optimizadas para el rendimiento general más que para la sincronización exacta.

Sin embargo, los sistemas ciberfísicos dependen críticamente de la sincronización exacta. LabVIEW puede representar visualmente la sincronización requerida utilizando un bucle temporizado (Timed Loop) y producir una ejecución en tiempo real. Aunque los diseñadores pueden visualizar la ejecución en curso con detalle utilizando Real-Time Trace Viewer, no tienen una manera de anticipar el rendimiento detallado de la temporización por adelantado como parte del proceso de diseño.

Dado que es necesario especificar sólo la temporización precisa de los sensores y actuadores, el compilador debería programar el resto del software para adaptarse a las E/S. Las innovaciones en la representación pueden hacer que sea más fácil especificar la sincronización precisa de

las E/S sin restringir abiertamente la temporización del resto del software. Es posible construir los sistemas distribuidos ciberfísicos de hoy en día con una temporización precisa, pero será mucho más sencillo en el futuro con una representación de software más explícita del tiempo, junto con mejoras en el hardware para la comunicación isócrona.

Otra área de investigación hace referencia a la configuración de los recursos del hardware y software implementado. LabVIEW muestra actualmente esta información orga-

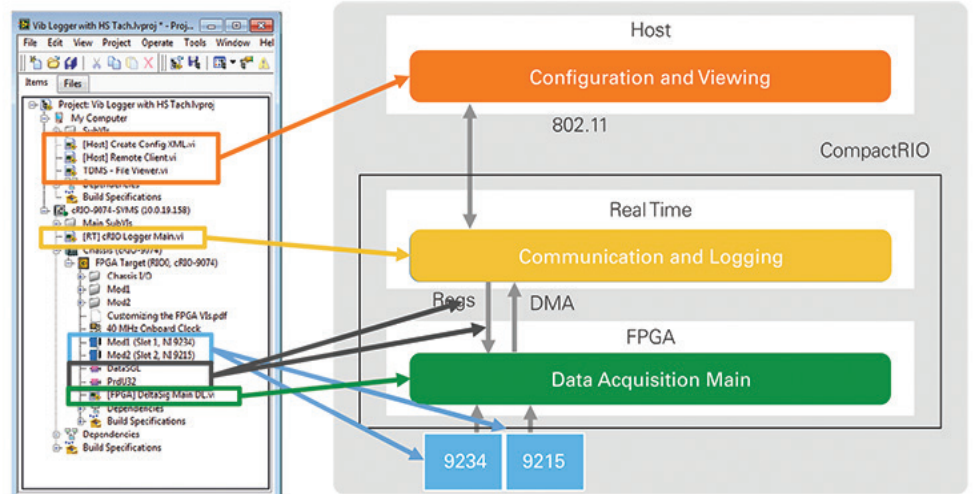


Figura 3. Un árbol de proyectos no ayuda a los diseñadores a visualizar las conexiones entre componentes.

nizada en un árbol en la ventana del proyecto. Puesto que un árbol no puede mostrar las conexiones entre los componentes, los diseñadores tienen que usar su imaginación para visualizarlos. La ventana del proyecto de LabVIEW necesita una representación esquemática para mostrar la configuración de componentes de hardware, interconexiones, sensores y actuadores; así como, potencialmente una representación de la máquina física.

Será posible realizar una transformación sin problemas desde una visualización física foto-realista del sistema de visualizaciones lógicas a múltiples niveles de abstracción. La representación lógica puede mostrar el software implementado en cada componente programable.

La temporización y la configuración son sólo dos de las muchas áreas de investigación en las que estamos trabajando activamente para ampliar las capacidades de nuestra plataforma. El uso de una pizarra borrable es un indicador que ayuda a estimar cuánto queda por hacer.

Si tengo que dibujar en una pizarra borrable para explicar mi diseño, entonces sé que hay más oportunidad de plasmar datos que con mejores herramientas.

El diseño de sistemas basado en plataforma produce sistemas flexibles y escalables con una mayor productividad y un menor coste total que los métodos tradicionales. Y la continua evolución de un método gráfico significa aún más que el rendimiento y las ganancias de productividad están por venir. 📌