technalysis
RESEARCH

WHITE PAPER

On Behalf of NATIONAL INSTRUMENTS

Building a Tools Platform for a Specialized IOT World

By Bob O'Donnell, TECHnalysis Research Chief Analyst

SUMMARY

The demands placed on engineers and the tools they use to do their jobs continue to rise. The broadening of technology-related products as well as the increasing specialization puts a whole new level of requirements on engineers whose job it is to design, prototype, and test new capabilities. In order to address new needs, tool makers must evolve their offerings to match the demands of this quickly changing market. Given these conditions, it's becoming increasingly clear a flexible, platform-style approach is necessary in order to deliver a set of tools that can enable a faster time to market, while at the same time building on a common core of well-established features and functions. A platform-based solution enables the creation of more specialized, targeted offerings that package together the specific capabilities needed by different segments of the market.

> "By building a set of specialized, powerful tools across a number of different disciplines, all of which leverage a similar visual approach and philosophy, and enabling these components to share their output with others, it's possible to provide a framework for teams of engineers to work together like never before."—Bob O'Donnell, Chief Analyst

INTRODUCTION

Look into the heart of any engineer and you'll likely find someone who loves to build, design and create. Turning ideas and concepts to reality is just part of who they are. From working with LEGOs as a child (or even an adult!) to designing mobile communications devices, rocket ships, or anything in between as part of their professional lives, engineers are uniquely focused on making things. To put it into the terms of a growing cultural movement, they are the original Makers.

In fact, the concepts driving the popularity of the Maker Movement—the phenomena in which individuals are building clever, interesting or fun things from low-cost components—has rejuvenated the field of engineering. Whether it's building simple robotic projects, creating modern art installations, or manufacturing specialized products that other people can use, the Maker Movement has made people realize that there's a bit of an engineer in many of us.

Part of the recent impetus for the Maker Movement is the growing awareness of how you can integrate the low-cost compute and connect resources now commonly available in an enormous range of devices. In addition, we've seen the creation of more complete, user-friendly tools that are more specifically targeted at completing tasks. For example, LabVIEW for Lego Mindstorms is much more well-suited to the task of programming the Lego brick computer than, say, a general purpose programming language like C or C#. The concept of IOT, or the Internet of Things, is based on leveraging the capabilities of these types of newer, simpler tools into a wide range of specialized projects, some of which are perfectly suited to the Maker mindset and others which require more advanced engineering skills.

THE INFLUENCE OF THE MAKER MOVEMENT, COMBINED WITH THE INCREASINGLY SPECIALIZED VARIETY OF POTENTIAL PROJECTS, IS STARTING TO CAST A NEW LIGHT ON THE TOOLS USED TO HELP BRINGS THESE PROJECTS TO LIFE.

Regardless of the difficulty level, however, the influence of the Maker Movement, combined with the increasingly specialized variety of potential projects, is starting to cast a new light on the tools used to help brings these projects to life. For example, many of the simple, straightforward tools used to design and create the software driving things such as LEGO Mindstorms, or the FIRST robotics competition projects that many students compete in, offer a compelling new way of tackling the problems and challenges that professional engineers face. Rather than complex tools that only engineers can understand, many of these tools provide a simple set of visual building blocks that even non-engineers can understand on a conceptual level.

But that doesn't mean more advanced tools built on some of these same principles aren't powerful enough for serious work. By diving into the depths of the offerings, hard-core engineers can tackle just about any kind of project—from prototyping and designing 5G network infrastructure to building the software running the Large Hadron Collider.

The key is having a fresh perspective on how to tackle these problems, a consistent approach and methodology to address them, and a powerful set of visually-based tools that allow big projects to be broken down into their component parts. Particularly with today's increasingly complex, technology-driven world and the increasingly specialized products that populate it, the need to have powerful tools that not only let individuals achieve impressive things on their own, but also work effectively with others has never been more apparent.

This is a particularly problematic issue for today's technical work force. Many engineers are stuck using a variety of legacy or proprietary tools that may be well-suited to one task, but don't necessarily work well with others. In other cases, they require completely new tools to integrate the efforts of co-workers or to test their work on real hardware instead of just a simulator.

What today's technical professionals really need is the ability to leverage the work of their fellow engineers, work across a variety of different disciplines, deploy code to hardware in order to interact with the real world, and augment their toolset as new technologies become available. In short, they need an operating system-like environment with support for multiple applications and hardware platforms to achieve their work, not just a singular tool. They need a tools platform.

A TOOLS PLATFORM

Companies build software platforms for a variety of different reasons, but the fundamental goals of most platform efforts are similar:

- Create a flexible environment to enable a wide variety of solutions
- Provide a common set of tools to reduce training time
- Build a consistent interface to enable ease of learning
- Offer a consistent way of working across tools
- Enable extensibility so that new capabilities can be added as required
- Address the widest range of specialized requirements

By providing these kinds of capabilities, software platforms reduce the need for companies to have to rebuild core tools from scratch. This enables new features and functionality to be added more quickly, while at the same time providing a consistent working environment for existing users. In addition, a platform environment can enable faster time to market, reduce training costs, and provide a host of other benefits.

Applying these principles to the world of engineering and testing tools, it's not hard to imagine the benefits of an engineering tools software platform that links directly to a supported hardware ecosystem. Conceptually, this is very much like a general computer or mobile operating system working in conjunction with a PC or mobile device and connected peripherals. In fact, you could even call it an Engineering OS.

The idea of an Engineering OS (eOS) is not literally an operating system, but rather a unified platform of tools that incorporates an array of different applications and hardware components, provides simple connectivity across both hardware and software, and is extensible. It does share some important conceptual similarities with a traditional OS, such as a valuable abstraction layer, standard APIs (application programming interfaces), inter-application communications, and a set of drivers to talk to connected hardware, but is not intended to replace a traditional PC OS. Rather, it's best thought of as a "metaOS" that sits on top of traditional operating systems, like Microsoft Windows or Linux RT, and provides an important set of shared capabilities that are specifically targeted towards engineers.

THE IDEA OF AN ENGINEERING OS (EOS) IS NOT LITERALLY AN OPERATING SYSTEM, BUT RATHER A UNIFIED PLATFORM OF TOOLS THAT INCORPORATES AN ARRAY OF DIFFERENT APPLICATIONS AND HARDWARE COMPONENTS, PROVIDES SIMPLE CONNECTIVITY ACROSS BOTH HARDWARE AND SOFTWARE, AND IS EXTENSIBLE.

There are a number of important advantages this OS-like platform offers versus a loose connection of different tools. Not only do all the applications in a platform have a consistent look and feel across the user interface, they have a consistent logic behind how they operate. Even more importantly, they enable code-reuse and component leverage. Chunks of code written by one set of engineers in one app can be easily integrated into the work of another set of engineers in another app.

For example, if one engineer is working on a signal processing algorithm that will ultimately end up in a piece of communications hardware test equipment, he or she can create that algorithm leveraging a visual environment that's best suited to their work, save that work as a component, and pass it along to his or her colleagues. In turn, this second group of engineers can import this component directly into their hardware design tool, and continue building whatever pieces of the product they're responsible for in an environment that's been tailored to their needs.

There's no need to worry about using intermediate code translation or porting tools. Plus, if the second group wants to dive into the details of the algorithm in order to do some troubleshooting, the consistent file formats between tools—a key characteristic of a platform-based approach—makes that task very straightforward.

Another indirect benefit of using a consistent platform-based set of tools is the ability to both simplify and specify when it comes to training. All engineers can be trained on the overall structure and operation of the platform and then companies can focus on hiring and training specialists that have the right sets of expertise for their specific needs.

Platform Benefits

One of the key benefits of a platform-based approach is the ability to create tools that are specialized for different audiences. Much like how you can now purchase cooking kits that include all the raw ingredients you need to make a particular dish, so too can a customized configuration of a platform-based software tool be adapted to fit the particular needs of a given application. Sure, you could simply find a recipe and then individually purchase the different ingredients, but it's more efficient to make a particular dish if some of the initial "configuration" work has already been done for you.

With the growth in more specialized markets, including IOT, as well as the more focused requirements in other fields, there's a growing need to package some of the raw capabilities that you can find in more general-purpose tools, into more specific application clothing that's targeted at different industries.

WITH THE GROWTH IN MORE SPECIALIZED MARKETS, INCLUDING IOT, THERE'S A GROWING NEED TO PACKAGE SOME OF THE RAW CAPABILITIES THAT YOU CAN FIND IN MORE GENERAL-PURPOSE TOOLS, INTO MORE SPECIFIC APPLICATION CLOTHING THAT'S TARGETED AT DIFFERENT INDUSTRIES.

The telecommunications industry, for example, is currently in the midst of a fundamental evolution from today's 4G networks to tomorrow's 5G networks and there are a variety of specific requirements that engineers in this field have. These engineers must find new ways to handle more users and more data in existing infrastructure, while simultaneously developing new algorithms and finding new spectrum to handle the growing demand. By building an application that's tailored specifically for real-time communications algorithm prototyping and deployment, it's possible to create a more compelling solution for this group.

Taking a platform-approach to tool-building also enables higher feature velocity—the ability to bring new capabilities to several different applications more quickly. For example, by building libraries of pre-built plug-in like components that can be used to easily connect to simple data acquisition devices, more complicated instrumentation platforms, or anything in between, it's possible to offer these new capabilities more quickly than having to rebuild them for each application.

For engineering organizations that use multiple tool sets, this type of architecture can actually save money as well because part of this platform-like approach enables certain functionalities to be shared across different tools. No longer do you have to buy one tool just to get access to a specific capability. Instead, it's possible to see those capabilities being added to other tools, because a platform-style approach makes it easier for the tool vendor to integrate that capability into their other offerings.

Related to this, a platform-based approach also allows toolmakers to create a common, consistent user experience across their tools. This, in turn, provides a more consistent experience for users. Moving from tool to tool is usually significantly easier if all of the tools come from the same company, thereby reducing training costs and allowing engineers to be more productive more quickly. For example, with a set of consistent, built-in analysis libraries, test and measurement professionals can start piecing together application test suites as well as other types of projects very easily and very quickly.

Thinking back to the communications industry example, companies who are currently prototyping 5G components and creating test beds for next generation networks are in the midst of some very difficult engineering work. Traditionally, the product design work is done with a different set of tools and platforms than the work done by its test engineers. In fact, a great deal of time and money is often spent transitioning between the tools used in these two stages. This is particularly true in the communications world, where testing often has to be done across a wide range of devices, from handsets to base stations to cellular infrastructure and beyond.

However, if a communications company did their work on tools that are all based on the same platform, they could potentially save enormous amounts of time and energy. They could move their IP over from testbeds built in one tool to real-world product development built in a separate, but related tool and then back again as testing and product development progressed forward. Throw in the ability to work on both network infrastructure and the devices that will talk to that network, and the savings could multiply even further.

Equally important, a tool provider that lays out a vision for how it is investing to enable these connections can provide confidence to its customers, assuring them that they can be a viable partner for many years to come.

REAL-WORLD PLATFORMS

The concept of “platform” gets discussed a fair bit, but it’s important to look at how it can play out in the real world. At a very simple level, if you think about Black and Decker tools, the company offers a core “engine” onto which you can attach an enormous variety of different “peripherals” to get a certain task done. From simple drill bits to screwdriver heads to saws to lights, you can leverage the same engine to do several different jobs.

Expanding out to software, if you think about a company like Microsoft, they have built not only a true operating system platform like Windows to enable an enormous range of applications, they’ve also built out smaller, more focused ecosystems around things like productivity—notably Microsoft Office. While each of the tools in Office perform a specific task, the ability to easily share elements from one into another is part of what has made it such a dominant player in office productivity. But even beyond the basic Office concept, Microsoft has also built web-based extensions of Office, with things like Office 365, that make it even easier to share and edit key Office files.

Applying the same principles to engineering tools, it’s clear that there really isn’t a great analogy. Instead, you find numerous standalone applications that, while very powerful, are not particularly adept at being leveraged across different parts of an engineering organization. What the engineering world needs is a set of tools, based on a common platform and common file formats, which can provide an Office-like level of sharing and reuse.

> WHAT THE ENGINEERING WORLD NEEDS IS A SET OF TOOLS, BASED ON A COMMON PLATFORM AND COMMON FILE FORMATS, WHICH CAN PROVIDE AN OFFICE-LIKE LEVEL OF SHARING AND REUSE.

A CALL TO ACTION

Engineers love to create and make things, but they can only achieve their visions and goals with a powerful set of tools that they can use both to complete important work on their own as well as collaborate with their co-workers. The ability to tap into a shared platform of software and hardware tools that can tackle any level project—from the simplest Maker Movement-type project to the most complex, multi-level engineering challenge—is key to achieving those dreams.

National Instruments (NI) has a long history of providing a range of hardware and software development equipment, as well as visual programming tools, for test and development professionals. As an industry analyst who observes key technology trends, it's clear from watching the company that they are evolving from a sophisticated tools vendor to an organization that's focused on building a platform for engineers: an Engineering OS, if you will.

NI holds an important place within the engineering community and has offered critical components that could be considered part of an engineering platform for some time. From the company's core LabVIEW development environment, to their extensive line of test and measurement hardware, NI has been a critical contributor to empowering engineers to solve the Grand Engineering Challenges for nearly 40 years.

But as any business's financial boilerplate commonly says, past performance is no guarantee of future success. The key to maintaining relevance and value with your customers is to evolve with the market and provide them with what their needs are today and into the future. In the world of engineering tools, those evolving needs include the ability to design, build and enhance products at a quicker pace; support a wider range of more specialized requirements; and improve the usability of the tools so that a larger, wider variety of customers can take advantage of those capabilities.

National Instruments is tackling these issues by extending some of their most popular tools and adding more specialized tools. At the heart of these efforts is a new approach to building software from a platform perspective, leveraging the familiar visual style of the company's popular LabVIEW programming tool.

With this new platform approach to their software, and an Engineering OS-like mentality, NI is extending their capabilities even further, bringing together several of their different components into a unified platform that will allow their engineering customers to tackle even more challenging problems in a more efficient and more effective manner. By building a set of specialized, powerful tools across a number of different disciplines, all of which leverage a similar visual approach and philosophy, and enabling these components to share their output with others, NI is providing a framework for teams of engineers to work together like never before.