

Для этого нужна вся деревня

Почему вам нужна программная платформа, подпитываемая экосистемой

“Для этого нужна вся деревня”. Хотя эта африканская поговорка обычно ссылается на необходимость надежной опоры из ближайших родственников, чтобы дети росли здоровыми, вы можете быть удивлены, когда узнаете, что такой же принцип может быть применен к миру инженерного искусства. Темпы развития и обещания единого мира резко обостряют требования к проектированию и тестированию инженерных систем и в настоящее время больше, чем когда-либо, нужна “вся деревня” – “деревня” программного обеспечения. Сегодня инженерное программное обеспечение, как правило, эволюционирует по двум различным направлениям. На первом направлении вы ищете специализированное решение – программу, спроектированную для конкретной задачи или типа оборудования. На другом направлении вы ищете абстракцию – упрощенную связь с одним сложным принципом, которая облегчает более быстрое выполнение задачи, но ценой потери управления на низком уровне. Любой разработчик, который утверждает, что решит эти задачи, используя один тип программных инструментов, наивен или заблуждается. Я не знаю, какое из направлений хуже, но знаю, что какой бы путь вы ни выбрали, вам нужно обеспечить совместимость, иначе риск ответственности за сложность интеграции ляжет на вас.

История учит, если вы готовы слушать

История неоднократно учила нас, что использование для обеспечения функциональной совместимости сочетания разнородного программного обеспечения от конкурирующих разработчиков трудоемко, дорого и очень сложно. Совместимость проще достигается при построении сложной программы на одной платформе, а затем с помощью средств экосистемы надстраивать эту платформу специализированными IP и функциональностью. Эта концепция лучше иллюстрируется Операционной системой.

Десятилетия тому назад Microsoft Windows доказала преимущества унифицированной платформы, определив стандартные протоколы обмена данными на верхнем уровне общего фундамента – персонального компьютера (PC). В результате появился общий интерфейс, который легко расширяем, что сделало аппаратную платформу PC высокоэффективной. Со временем мы увидели точечные решения, такие, как кассовые аппараты, замененные персональным компьютером общего назначения со специализированным программным обеспечением. Можно привести аргументы, утверждающие, что Apple iOS выбрала концепцию платформы для следующего уровня. Надежная экосистема iOS, построенная на вершине стандартной инфраструктуры, была катапультирована в будущее.

Для достижения максимальной совместимости и эффективности проектирования разработчикам нужно было создать программное обеспечение, используя правильное направление. Подход компании NI состоит в том, чтобы использовать преимущества Операционной Системы настольного компьютера, достигнутые на рынке PC, и применить их в инженерном программном обеспечении.

Значимость инженерной операционной системы

В то время, как оба предыдущих примера были примерами потребительских операционных систем (ОС) общего назначения, аналоги могут быть перенесены вперед на проектирование и тестирование инженерных систем. Рассмотрим концепцию “Инженерной операционной системы” – программной платформы, созданной для стандартизации и упрощения взаимодействия между технической системой и человеком, ее спроектировавшим, точно так же, как традиционная ОС упрощает взаимодействие между человеком и PC.

<Рисунок 1> - Сравнение традиционной и инженерной ОС

Истинная инженерная ОС – это набор технологических компонентов или фундаментальных строительных блоков, которые могут быть использованы для создания различных программных продуктов, предназначенных для решения различных проблем. Эти фундаментальные строительные блоки являются общими практически для любого инженерного программного обеспечения – элементы интерфейса пользователя для визуализации данных, IP обработки сигналов, структуры компиляторов для оптимизации кода под разные архитектуры процессоров, профессиональные языковые конструкции, API драйверов и модули для развертывания. Однако в пакете эти компоненты приспособлены для достижения максимальной продуктивности и эффективности при решении задач. Например, комбинация этих элементов в программу, спроектированную для управления высокоуровневой архитектурой выполнения последовательности тестов, не только выглядит специфично, но и реализует разные потоки операций, оболочки, IP и дизайн взаимодействия в программе, созданной для интерактивного конфигурирования аппаратуры и экспериментального сбора данных.

<Рисунок 2> - Скриншот NI TestStand (Испытательный стенд) рядом с NI Measure (Измерения) – легко пожертвовать, если есть чем

Целостный комплект этих программных инструментов обеспечивает ценность инженерной ОС. Узкоспециализированные продукты, такие, как драйверы аппаратных устройств, средства управления развертыванием приложений, управления распределенными системами созданы на этой особенной платформе. На этой замечательной платформе созданы такие высокоуровневые, основанные на конфигурировании продукты, как система управления последовательностью тестов, интерфейс облачной аналитики и готовое к применению программное обеспечение для онлайн управления мониторингом технического состояния. На этой замечательной платформе созданы хорошо настраиваемые среды проектирования, в которых доступны высокопроизводительный графический язык проектирования или язык проектирования, основанный на ANSI-C. На этой замечательной платформе есть каждый отдельный компонент, необходимый для решения ваших задач, и создан он таким образом, что способствует и упрощает интеграцию между компонентами. Более того, поскольку технологические компоненты инженерной ОС созданы на этой же платформе, они обладают прямыми и косвенными достоинствами, которые вы, как пользователь, хотите получить.

Простейшая траектория освоения

Сегодня тренд развития программного обеспечения прямолинейный, но не простой – требуются большая вычислительная мощность и производительность в сочетании со стартовой точкой высокого уровня и возрастающей простотой использования. Одно из преимуществ платформенного подхода, при правильном применении, заключается в том, что конечный продукт представляется общей и простой траекторией освоения. Общие возможности создаются базовыми стандартными строительными блоками, которые обеспечивают два уникальных преимущества. Во-первых, унифицированность (совместимость) этих базовых стандартных строительных блоков означает, что пользователям не нужно изучать заново эти основные элементы при объединении нескольких программ в систему, поскольку парадигма совместимости является общей для платформы. Во вторых, команда разработчиков может внедрять эти основные элементы в адекватный сквозной процесс проектирования, который позволяет использовать для решения задачи накопленный вами специальный опыт.

Совместимость

С ростом сложности современных решений повсеместно приходят к необходимости комбинирования нескольких языков программирования, операционных сред и методов проектирования. Однако стоимость интеграции этих методов существенна и продолжает увеличиваться. Платформенный подход упрощает организацию взаимодействия по нескольким направлениям. Во-первых, встраиваемые IP и API являются общими для разных продуктов. Этому способствует повторное использование кода высокого уровня без сколь-нибудь значимой рефакторизации или перепроектирования. Это может радикально снизить стоимость интеграции внутри компании в дополнение к интеграции на уровне отдельного пользователя. Это также исключает бифуркацию между абстрагированием и специализацией. Узкоспециализированное программное обеспечение может быть

разработано в рамках одной платформы для выполнения конкретной задачи, например, регистрации данных, отдельной корреляционного анализа и обработки данных офлайн или даже управления испытаниями. Впоследствии специализированное программное обеспечение может быть расширено или интегрировано с помощью средств высокого уровня. В конечном счете, используя эту уникальную комбинацию возможностей, вы можете выбрать подходящий инструментарий для каждого аспекта вашего проекта и не обременять себя дороговизной решения задачи, связанной с интеграцией этих инструментов.

Экосистема

Становится истиной для удачной программы – без экосистемы программа должна бороться, чтобы считаться релевантной. Не будет удачным профессиональный программный инструментарий, который изолирован в бункере. Экосистема обеспечивает громадное количество ценных качеств для поддержки, создания примеров кода, сетевого взаимодействия и стороннего анализа. Наверно, наиболее важно то, что экосистема может расширять платформу узкоспециализированными IP сверх компетенции поставщика программного ядра или за пределы его экспертной области. Взгляните на Apple App Store – инженеры Apple не являются эксперты в фитнес-мониторинге или в считывании показаний с метеорологических РЛС, но такие возможности были созданы в основе их платформы.

Метод проектирования программного обеспечения, основанный на платформе, не только продвигает принцип расширений экосистемы, но действительно может быть использован для упрощения интерфейса между связанными критическими элементами, чтобы усовершенствовать расширение. Фундаментальный принцип платформы заключается в том, что возможности отдельно взятого ядра спроектированы, как повторно используемые сущности, охватываемые множеством продуктов, которые могут быть в дальнейшем адаптированы под конкретные требования пользователя. Путь к эффективности только в том, чтобы реализовать это при проектировании расширяемого набора API ядра, используемого командами разработчиков различных продуктов для развития и настройки функциональности. Это те же самые API, которые дают возможность дальнейшей настройки функциональности приложений, разрабатываемых командами, их компанией или их клиентами, если они построили бизнес на основе платформы.

Внедрение инженерной операционной системы в жизнь

В августе 2016 на нашей ежегодной конференции NIWeek мы впервые анонсируем доступный в любое время предварительный обзор программных технологий NI (NI Software Technology Preview) для того, чтобы продемонстрировать нашу приверженность инженерному прогрессу, который создавался на рынке измерений и тестирования в последние три десятилетия. Особенности некоторых программ NI Software Technology Preview в объединении связанных друг с другом возможностей. Вместе с другими особенностями эти возможности организованы вокруг решения ключевых проблем, которые стоят перед вами, как конечными пользователями, в том числе:

- Разработка интерфейса пользователя от настольного компьютера до Интернета
- Системное управление, визуализация и документирование
- Встроенное в продукт изучение с помощью интеграции развитых медиа технологий
- Обработка данных и управление данными на уровне настольного компьютера и на уровне сервера
- Создание способов программирования с помощью программных средств, основанных на конфигурировании

Такая постановка проблемы охватывает множество отраслей промышленности, и проблема может быть решена уникальными способами с помощью различных продуктов. В этом прелесть использования подхода, основанного на платформе. Возможности, демонстрируемые Software Technology Preview, созданы таким способом, что на платформе NI они могут быть интегрированы во многие программные продукты.

Несмотря на то, что Software Technology Preview сфокусирован на будущие нереализованные разработки и возможности, в нем есть современные решения, в которых реализованы преимущества инвестиций в эту платформу. VeriStand 2016 – программный интерфейс высокого уровня, созданный для упрощения тестирования приложений реального времени, включает инструментарий для проектирования графических интерфейсов пользователя, которые просто встраиваются и развертываются. В системе проектирования программного обеспечения LabVIEW 2016, выпущенной к 30-летию юбилею LabVIEW и являющейся непрерывным развитием предшествующих версий, начиная с 1.0 для Macintosh Plus, появился новаторский коммуникационный протокол, который использует один проводник для передачи данных между параллельно исполняемыми сегментами кода.

<Рисунок 3> - Администратор интерфейса пользователя VeriStand и LabVIEW 2016 Hero Shot

Требование платформ – опережать конкурентов

Конечно, не просто разработать эффективную программную платформу. Однако решение этой сложной проблемы – надежные инвестиции для поставщиков программного обеспечения, необходимые, чтобы помочь решать задачи возрастающей сложности, стоящие перед вами и вашими компаниями. Если вы по-прежнему используете точечные решения задач и тратите драгоценное время и энергию на интеграцию, а не на решения проблем бизнеса, вы, с трудом, но, в конечном счете, решите задачу, однако найдите время для обдумывания подхода на основе платформы. Нужно ли вам решить инженерную проблему прямо сейчас или необходимо подготовиться к предстоящим изменениям технологии, вам необходима платформа, которая предоставляет верную отправную точку, использует правильный метод проектирования программного обеспечения и поддерживается надежной экосистемой.