
Prevas
Innovation for Growth

2017-11-09

Test System Development Using Frameworks

Jouni Karinen

CLA

Test System Architect, Prevas

Prevas

Outline

- Frameworks in software development
- Prevas SW Framework for Test and Measurement
- Example solution for a customer
- Framework as a reusable concept
- Example solution for high speed capacitor testing
- Summary and Questions

Prevas

Frameworks in Software Development

- Designed to represent a particular application domain
  - Should function as an architecture/design for creating customized applications on the same domain
  - Customization of the specific application is done using the abstraction and interfaces provided by the framework
- Promise of shorter application development time and increased productivity
- Enhances reuse of the code and the design
- Usually based on object-oriented design

Prevas

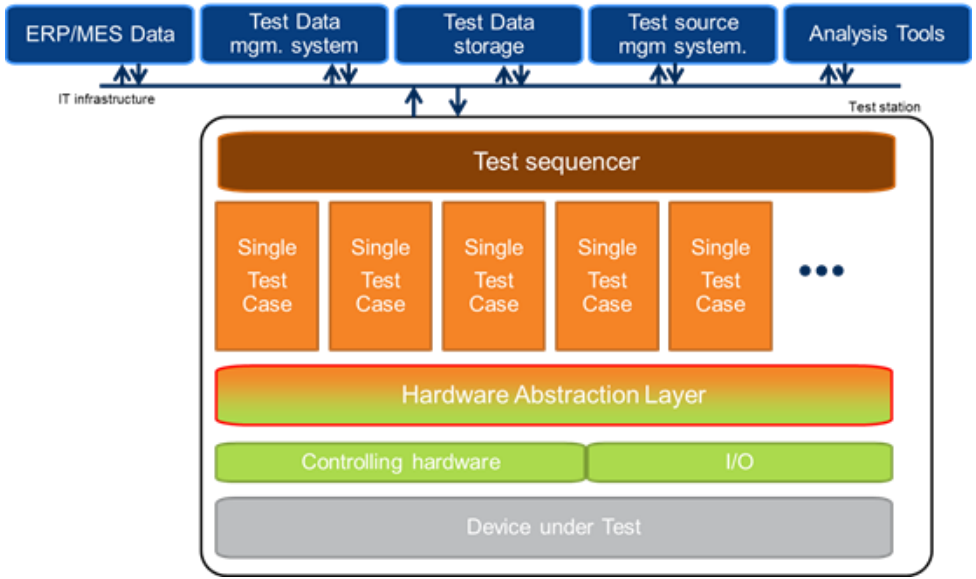
Test applications

- Testing could be thought of as one domain – but the applications needed in test can be very different by nature
  - One framework can hardly cover all the needs in test applications

- One clear difference is test sequencing
  - This can be the most crucial part of the application or fixed and almost irrelevant
  - There are good frameworks/programming environments for test sequencing such as TestStand

- Common to almost all test applications
  - The need of making physical measurements
  - Controlling something in the environment of the test
  - Creating a report of the test showing the results

High-Level Test System Setup



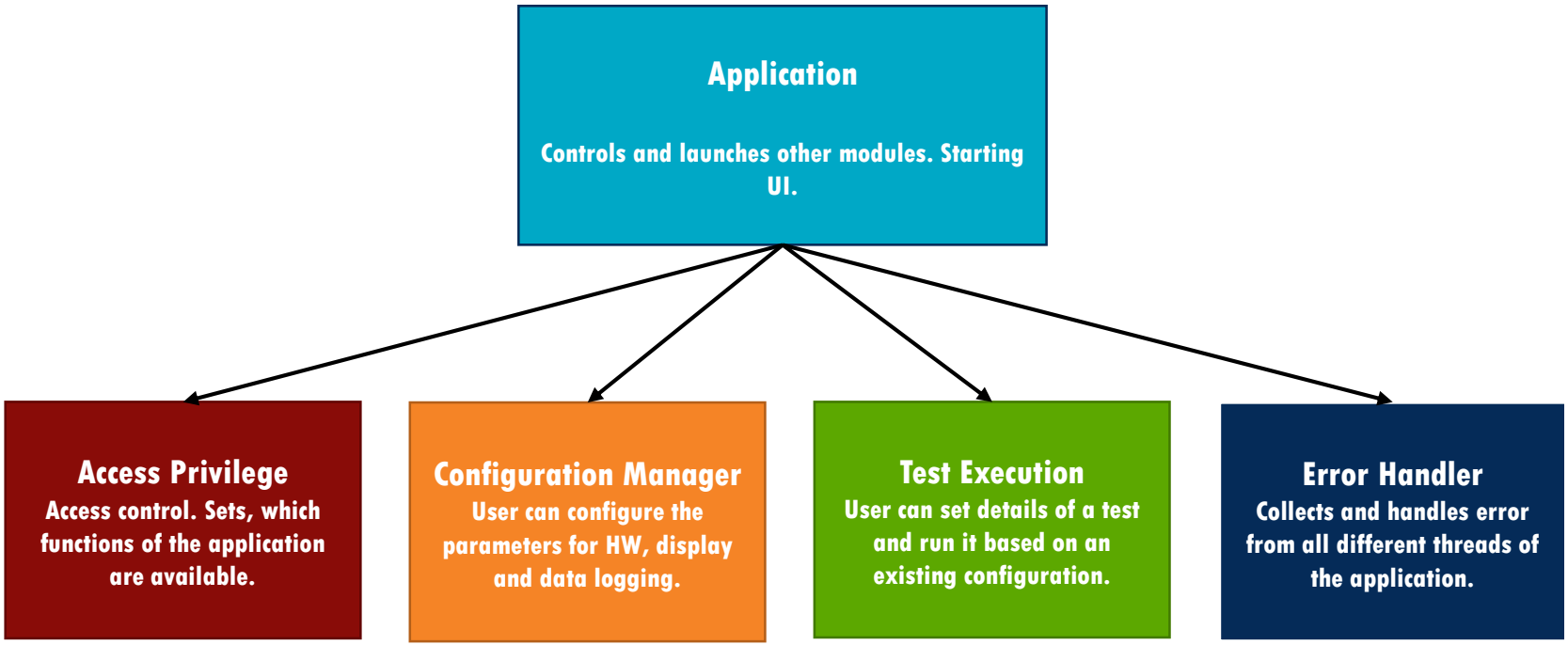
Prevas

Building Frameworks in LabVIEW

- Using classes and inheritance for abstraction
- Using actor framework for controlling code modules and threads
- Using libraries for common functions as enhanced error handling
- There are existing frameworks available in LabVIEW tools network – no need to write code that has been done already
  - http://www.ni.com/labview-tools-network/

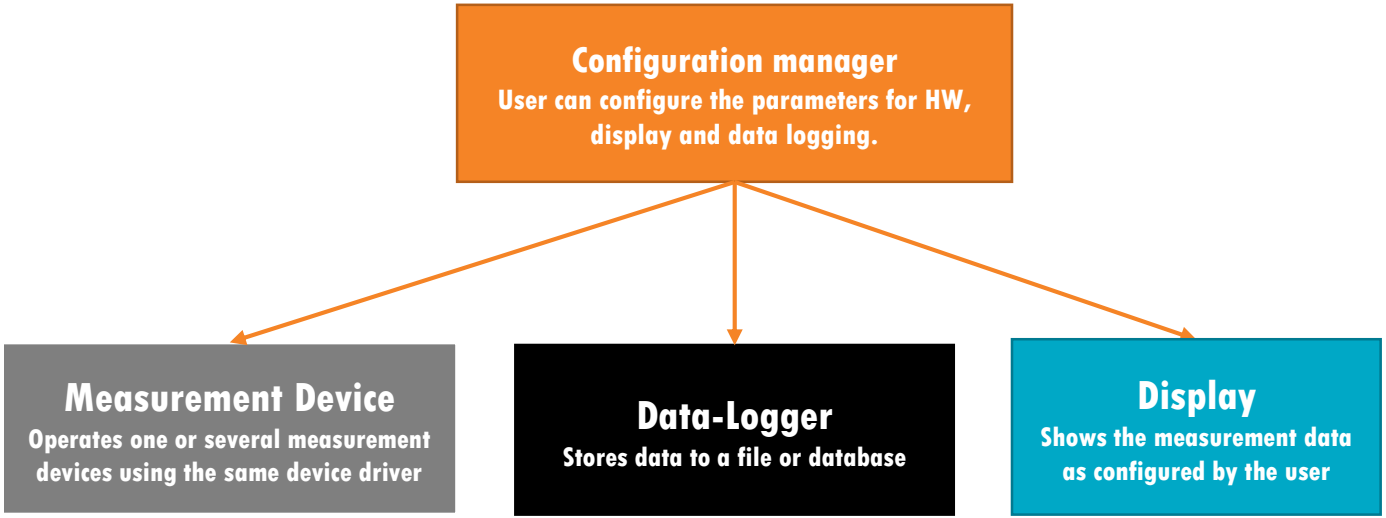
Prevas

Prevas Test SW Framework



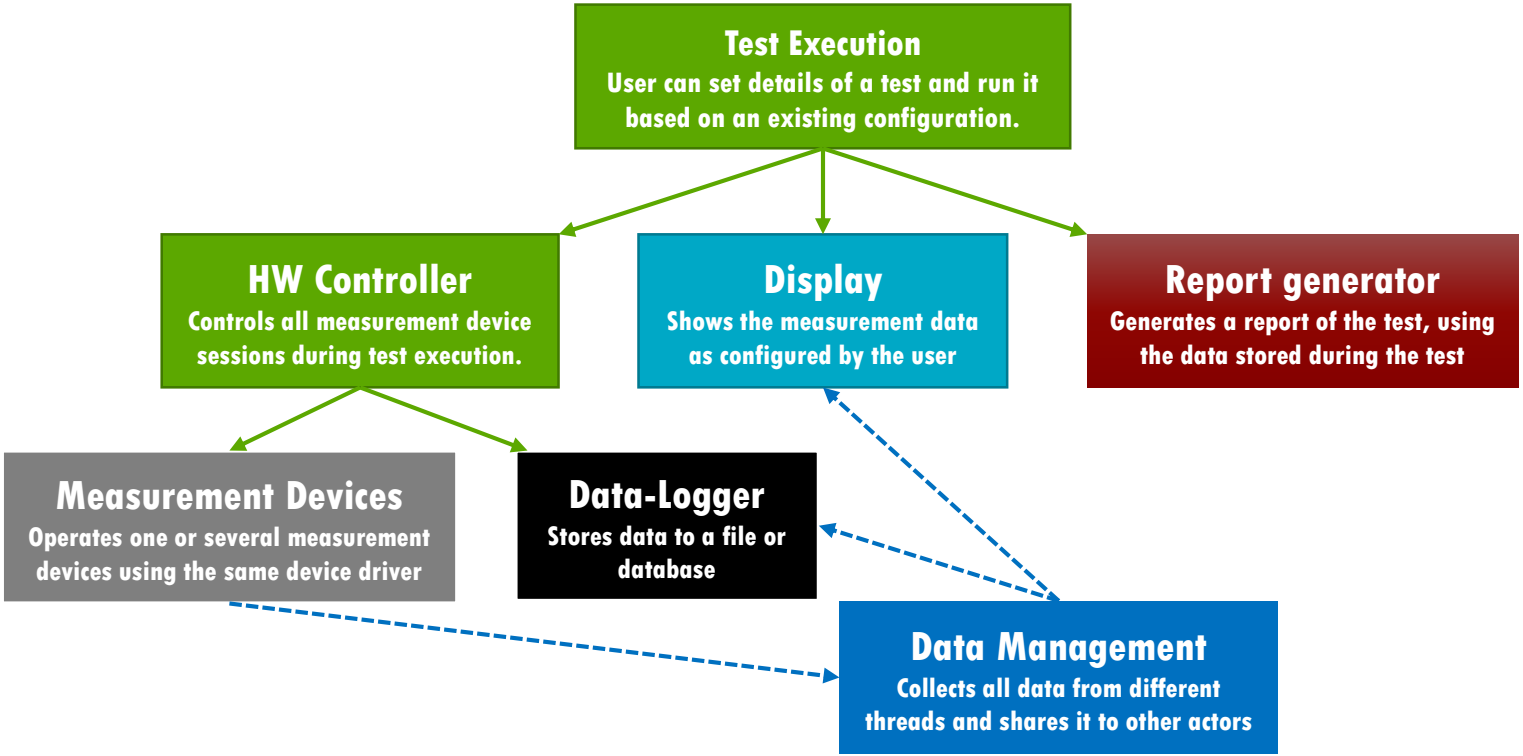
Prevas

Prevas Test SW Framework
- Configuration Manager



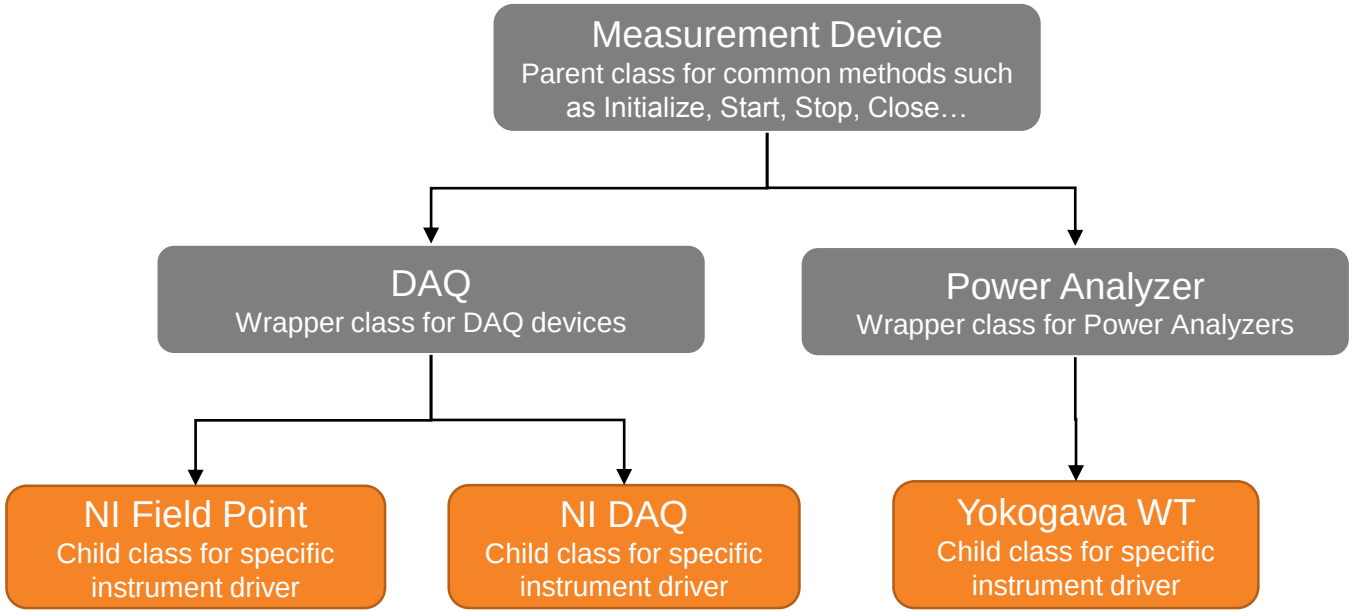
Prevas

Prevas Test SW Framework - Test Execution



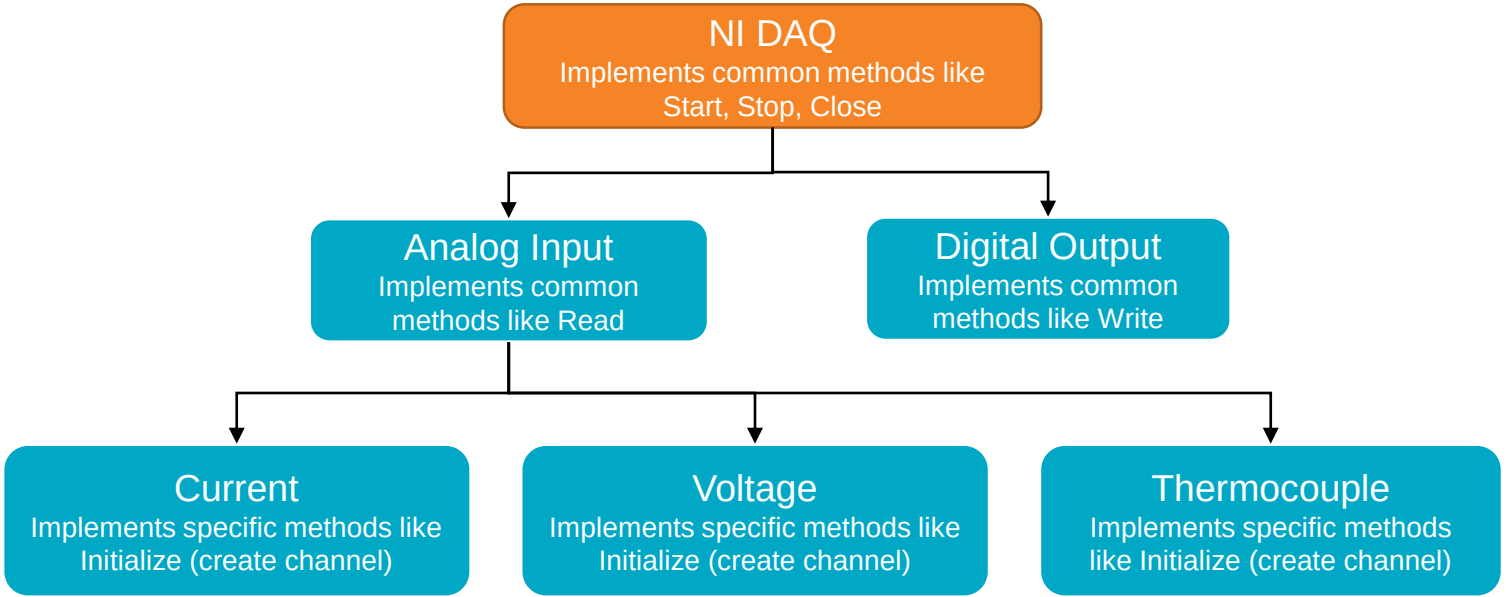
Prevas

Hardware Abstraction Layer Example



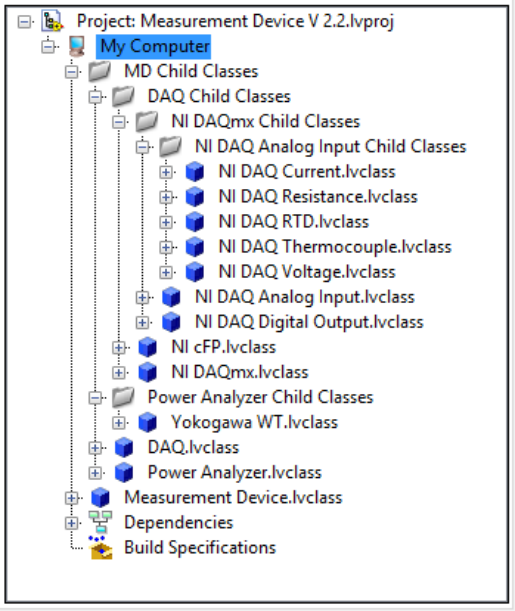
Prevas

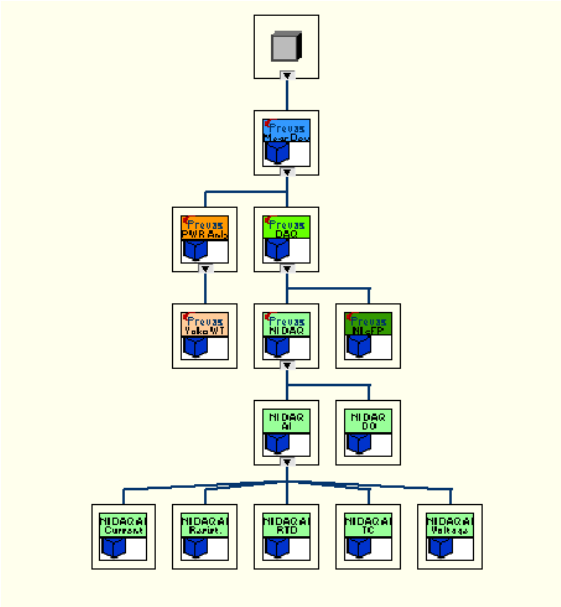
Hardware Abstraction Layer Example - lower level



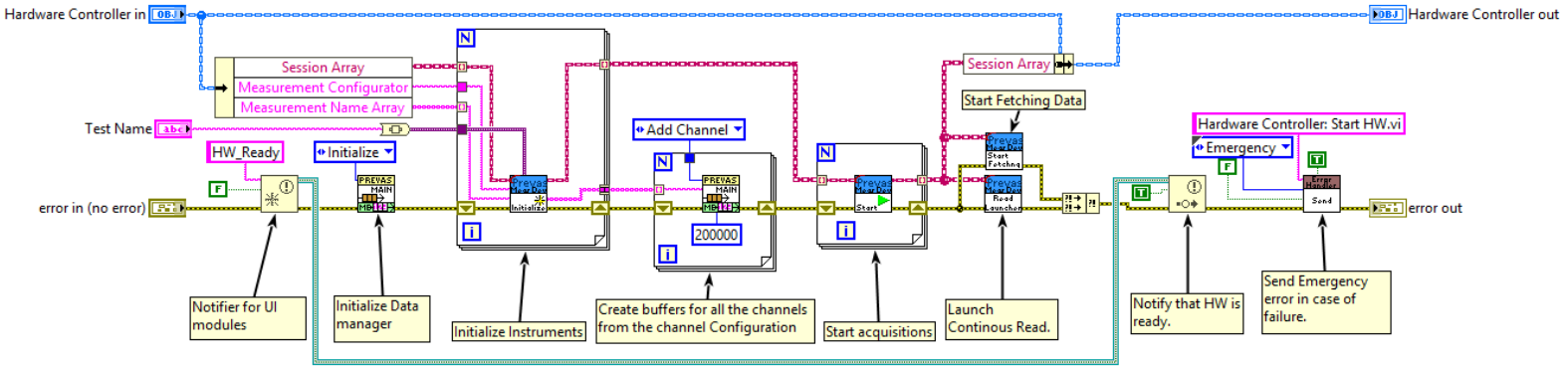
Prevas

Hardware Abstraction Layer Example

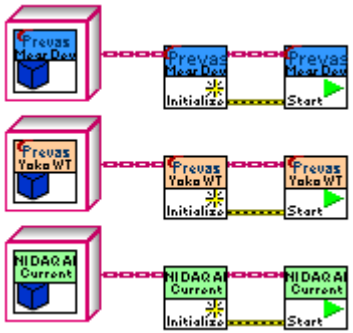




Inheritance and Overriding Methods



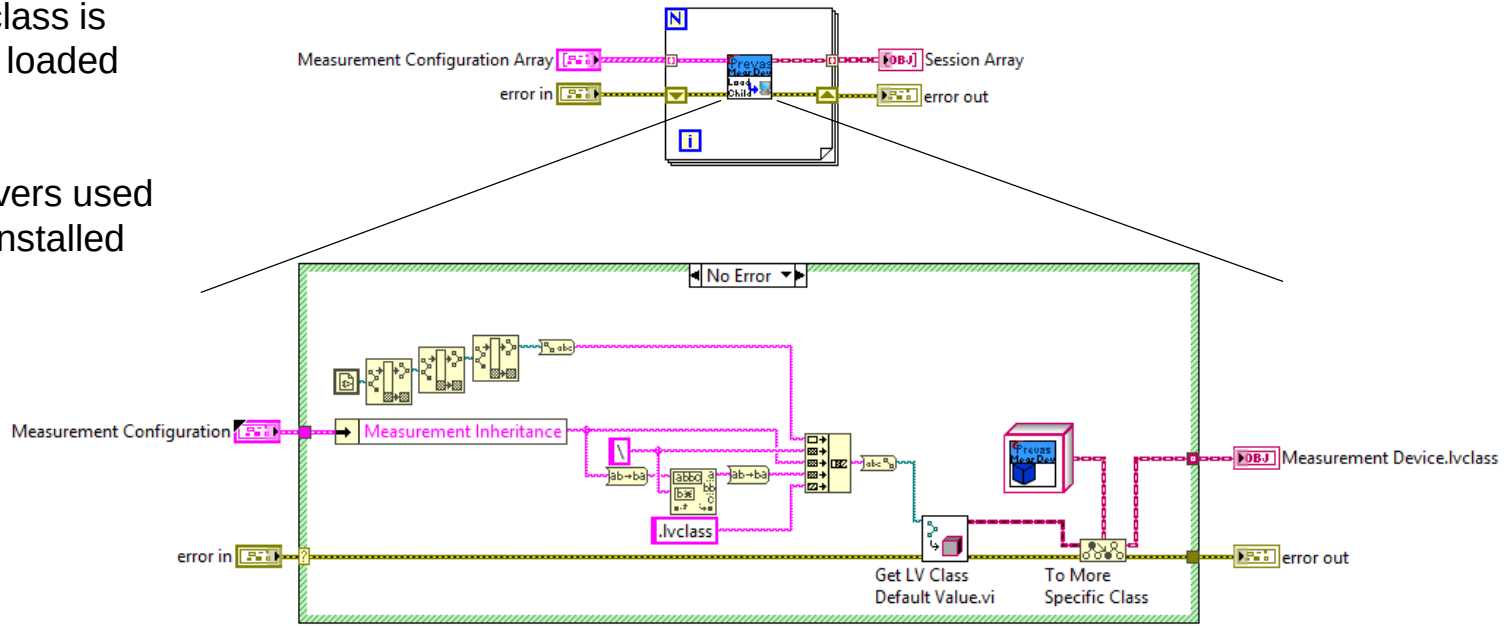
- Only parent class methods are used in the application
- Depending on the object type in Session Array – a child method is dynamically called



Prevas

Dynamic Loading

- Each child class is dynamically loaded
- Only the drivers used need to be installed on the disk



Example Project – the Task

- Customer has an existing production test system that needs new features
  - Application is developed with LabVIEW – but is pretty much spaghetti code
  - Production is not mass production and test is fairly simple
  - There are several different physical locations and hardware setups needed

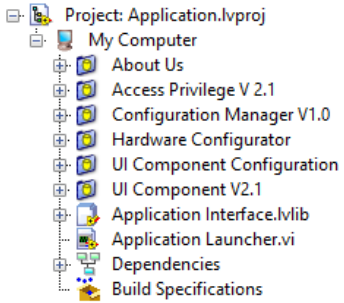
- Existing hardware still needs to be used – and new has to be added
  - Some of the old hardware needs to be updated, but this can't be done in the first phase so the new software must support it as well.

- Timeline is very tight – only 2-3 months of time during the summer

Prevas

Example Project – the Approach

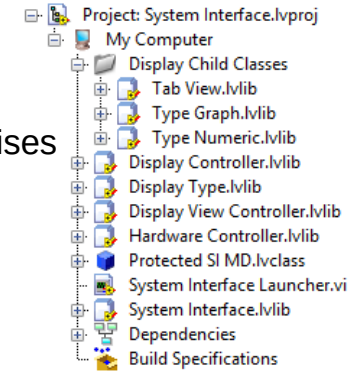
- The tests do not need much sequencing → LabVIEW based application is chosen
- Many different pieces of HW are used → HW abstraction layer is seen as a good solution
- Many different test setups are needed in the same application → Configuration based modular approach is chosen
- Timeline is tight → There isn't enough time to implement everything from scratch
- Our framework fits very well in the chosen approach -> We'll use it and develop the missing parts on the way



Prevas

Example Project – the Result

- We got the basic application done on time
  - We got a fast start by re-using code from our framework and reduced the risk at start
  - Due to lack of time and new emerging requirements during development, we had to change the approach of framework into application specific for some parts of the code
- We got part of the framework developed further
  - We were able to develop our framework forward, but had to make some compromises
- The application is modular and configurable
  - Adding new hardware to the application will be easy (hardware abstraction layer)
  - It will be easy to add new functionalities



Prevas

Key Learnings from the Project – Using Framework as a basis

- Developing and managing a complex project is easier when it's based on a framework
  - A new developer can jump in and understand the code easier – especially if he's familiar with the framework
- Time saving and a quicker start
  - Development can concentrate on solving the application specific issues from the start as the overall architecture is already in place
- Ensures modularity
  - If the framework is done to be modular, the resulting application will be as well
- If the calendar time for delivering solution is tight – do not plan to develop your framework in parallel with the project
  - It's better to use what exists and create a specific application on top of it. After finishing, applicable parts can be integrated to the framework.
  - If there is more time – do framework development first and start the application development after that

Prevas

Framework as a Recyclable Concept

- A single framework is not applicable for all test applications

- Design principles and style of a framework are recyclable
  - An application that is partially different can still benefit from an existing framework
  - Only one part of the framework can be used and modified for the need
  - The design principles and style can be used for creating a completely new code → This saves time and creates confidence in the architecture of the new application

- When starting a new project / application design, it is important to evaluate the possible use of the existing frameworks
  - A poor choice at the start of the project can cost a lot of time

Prevas

Thin Film Project Example – the Customer

thinfilm

“Thin Film Electronics ASA (“Thinfilm”) is a global leader in NFC mobile marketing and smart-packaging solutions using printed electronics technology.

Thinfilm creates printed tags, labels, and systems that include memory, sensors, displays, and wireless communication — all at a cost-per-function unmatched by conventional electronic technologies.”

Source: http://thinfilm.no

Prevas

Thin Film Project Example – the Task

- Create an application for production testing
  - Measurements of a new roll-to-roll process
  - Application should be designed in LabVIEW
  - Application should be modular, extendable and FAST
- Test system requirements
  - 1000 DUT's tested per 1 second
  - Test can run over 12 hours
  - Current measurement accuracy ~1 nA
- Pre-selected hardware by customer
  - NI PXIe-4141 power supply and and Pickering PXI Solid State Multiplexer
  - Needs to be evaluated for the purpose
- Mechanical/probing solution developed by another supplier
  - Communication between electrical measurement system and probe system needs to be defined and implemented

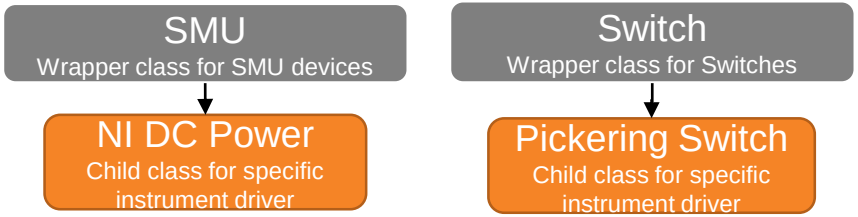
Prevas

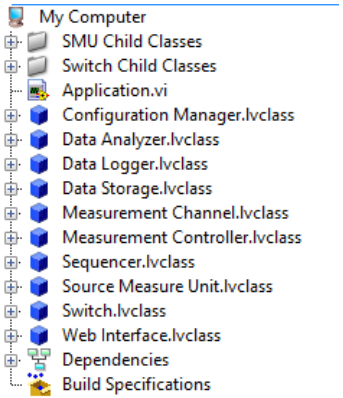
Thin Film Example - the Approach

- The test is simple “apply voltage – measure current” repeated for N times per DUT. → A simple LabVIEW based application is chosen
- The application has to be modular, flexible and extendable → HW abstraction layer is seen as a good solution
- Our framework is too complex for the simple task and is not designed for optimizing the speed
- Fast prototype of the SW is needed to test the speed of the HW & SW

→ We decided to recycle two design approaches from the framework:

  – HW abstraction layer (with modified class structure)
  – The high-level structure of the application





Prevas

Thin Film Example - the Result

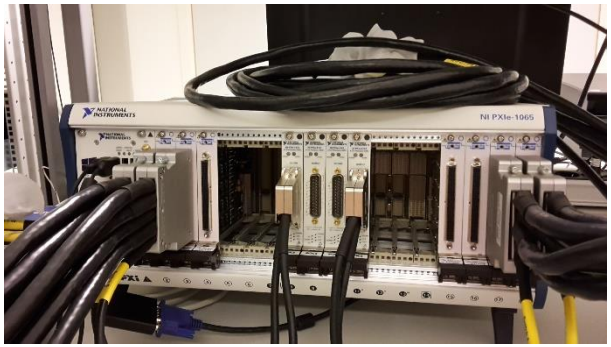
- We were able to create a simple software for the HW trials within 4 days
  - Due to resourcing issues – the work was split between two developers, who were familiar with the framework
  - It was a prototype SW but still with HW abstraction layer and multithreaded structure – thus simulating well the processing speed of the final solution
  - Some improvements with HW drivers were needed during the HW trials to increase the speed

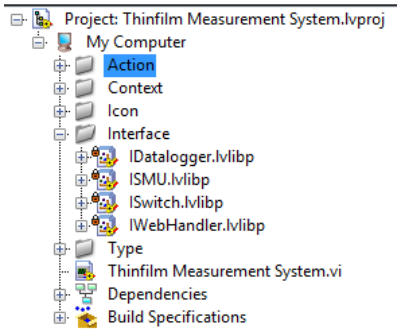
- After the initial trials and verification of the chosen solution we got good results
  - Running ~600 DUT tests per second
  - Noise level ~1 nA and measurement accuracy ~2-3 nA

Prevas

Thin Film Example - the Result

- A third developer was able to continue the work finalizing the application
  - The first version of code was embedded into another framework
    - E.g. the basic HW abstraction layer was improved into plug-ins
  - Configuration functions were designed specifically for this application
- After further implementation and with the final HW we reached the goals
  - Running ~1000 DUT tests per second
    - More efficient PXI controller
    - 14 parallel SMU channels with 69 switchable inputs to each
  - Running the test for 48 h without memory leakage or CPU load issues
- Communication protocol with the probing controller adopted to the application from the other sub-supplier
- The final testing of this solution is yet to come but it looks good!





Prevas

Key Learnings from the Project – Using Framework as a concept

- **Even solutions that have some very specific or unusual requirements, can benefit from an existing framework**
  - The framework can add significant value even if it's not used as such
  - Being able to concentrate in solving the special requirement – instead of designing the application
  - Several designers can work with the application, as the design has a familiar basis

- **By evaluating the existing framework/codebase significant time saving can be achieved**
  - Sometimes recycling is a good option to re-use

Prevas

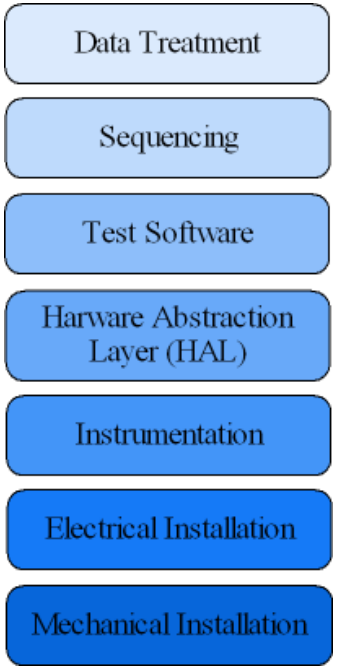
Summary

- Frameworks can be a great benefit for SW development
- The approach of using frameworks suits very well the test domain
- Usage of a framework should be carefully considered always when starting a project
- Frameworks can bring benefits also in forms of recycling ideas

Prevas

Prevas Knowhow in Test Systems

- **Data Storage and Handling**
  - WATS, NI Diadem, custom solutions
- **Test Cases and Sequences**
  - TestStand and LabVIEW based solutions
- **Software**
  - LabVIEW, .NET, embedded, Linux
- **Hardware Abstraction Layer (HAL)**
  - LabVIEW OOP concepts with dynamical loading
- **Instrumentation**
  - NI Alliance partner and in-house experience
- **Electrical Design**
  - In-house and partners
- **Mechanical design**
  - In-house and partners



Prevas

Thank You!

Prevas

Innovation for Growth

www.prevas.com