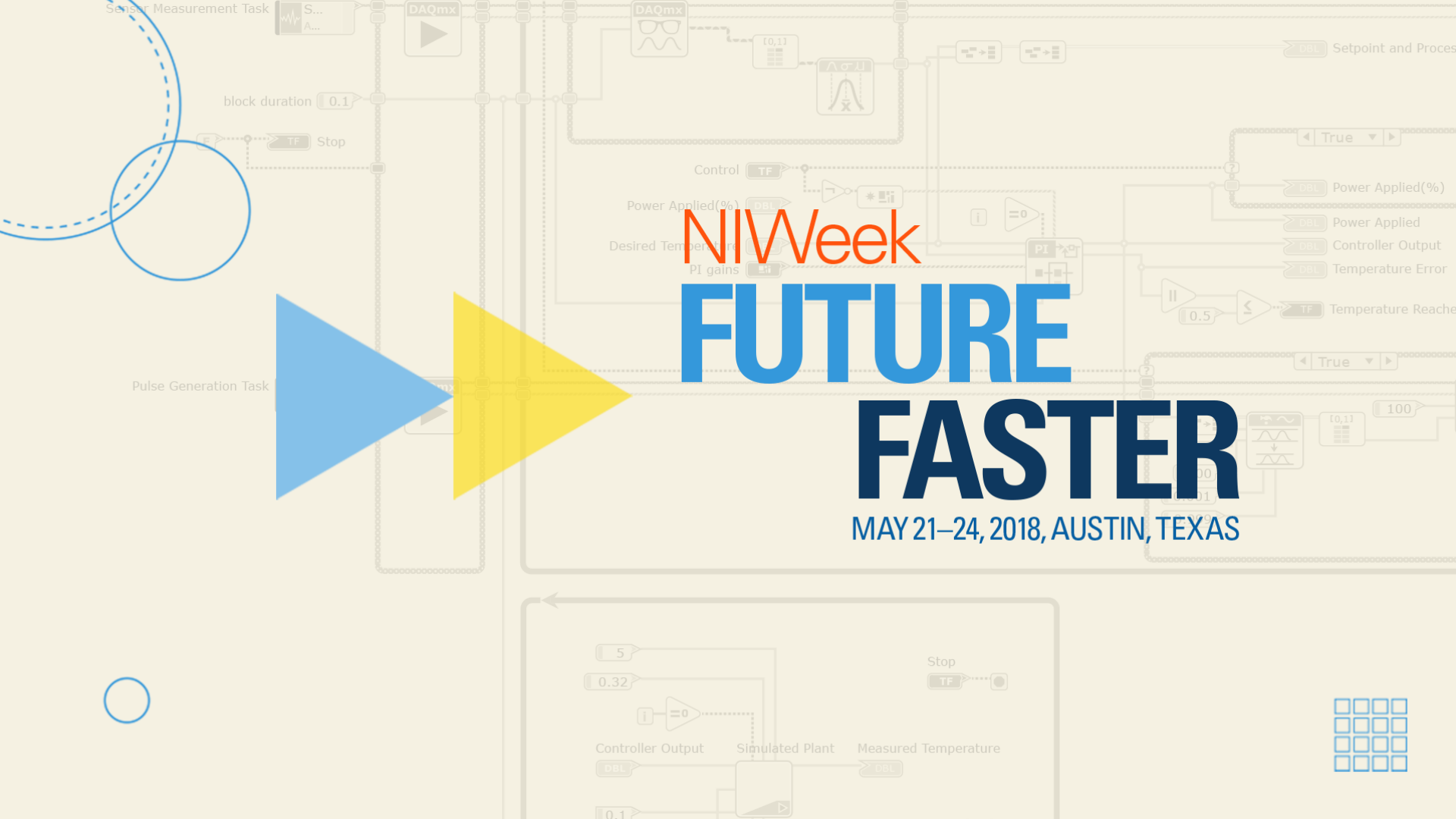


NIWeek

FUTURE FASTER

MAY 21–24, 2018, AUSTIN, TEXAS

Bringing Down The Barrier - A Pragmatic View on Software Design

Jeffrey Habets

VITechnologies

Founder / CEO

jeffrey@vi-tech.nl twitter.com/JeffreyHabets linkedin.com/in/vitech

Quote

> “Perhaps the greatest strength of an object-oriented approach to development is that it offers a mechanism that captures a model of the real world.”

Grady Booch

Your takeaways

- Starting with Object Oriented Development is not as hard as it may seem at first glance
- LabVIEW classes are easy!
- Confidence to take that first step
- You know tools that help you visualize the big picture and give you a huge productivity boost with OOD / OOP in LabVIEW and NXG
- You'll know just enough UML to be (not too) dangerous

Disclaimer

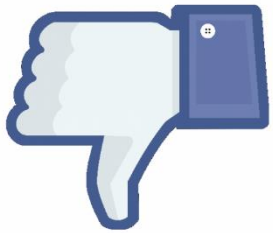
- I am not a UML or OO guru
- I use UML in a pragmatic way *(which is “just about correct”)*
- Yes, I hope you will start using the tools I will show you later on ☺

NATIONAL INSTRUMENTS

But wait…

- It's such a lot of extra work, all these extra files
- Creating classes and it's components is a hassle, a lot of extra clicks
- I don't need it! (*But, you do… You just don't* ***know*** *it yet!*)

- All these frameworks I hear of all the time, the terminology, principles, design patterns… Where do I start?



The Why

- Maintainability (be nice to future you!)
- Extensibility (Ideally only add code to add functionality)
- Testability (modular code is easier to (automatically) test)
- Reliability (Result of all of the above!)

- You can use UML to communicate your ideas!
- Just need to stick to some basic rules

Quote

> “Perhaps the greatest strength of an object-oriented approach to development is that it offers a mechanism that captures a model of the real world.”

Grady Booch

Basic Object Oriented Design Concepts

Three Pillars:

- **Encapsulation**

  An object:

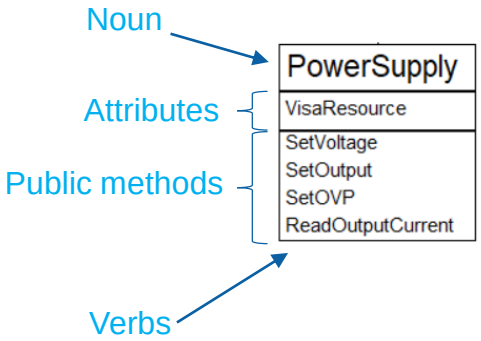
  - Encapsulates data and the methods that act upon that data
  - Group of VIs with a common responsibility

- **Inheritance**

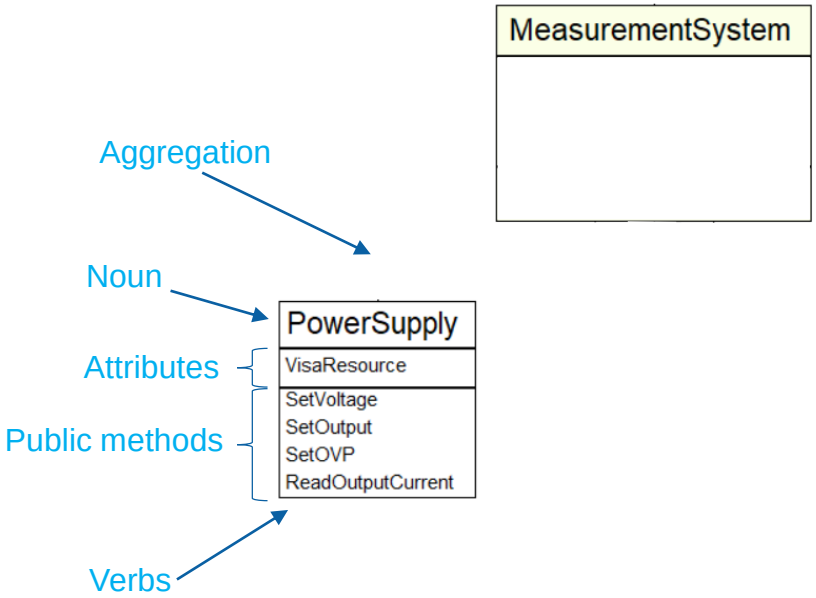
- **Dynamic Dispatching** (Polymorphism)

Think in terms of objects and actions, instead of VIs and sub-VIs

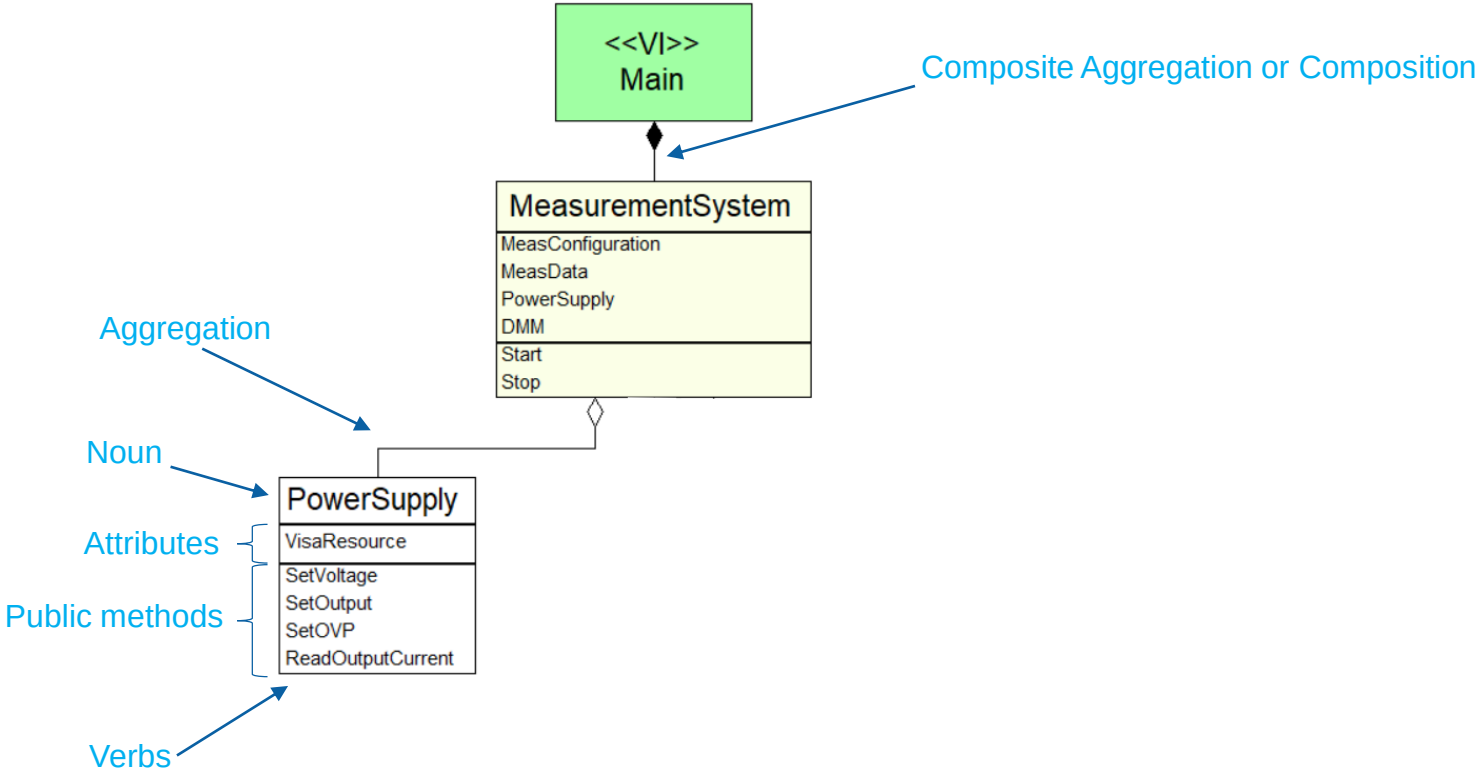
Class diagram – A Measurement System



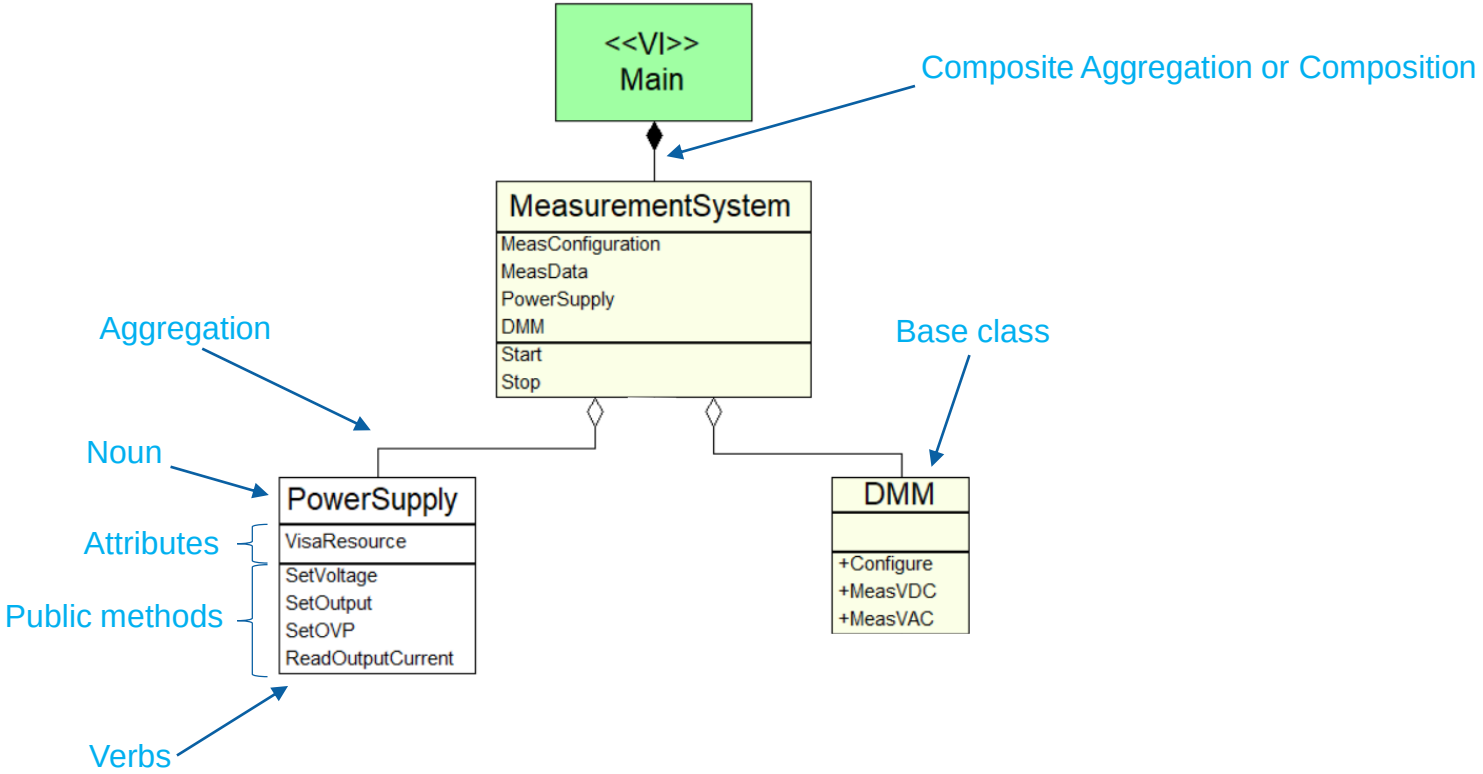
Class diagram – A Measurement System



Class diagram – A Measurement System

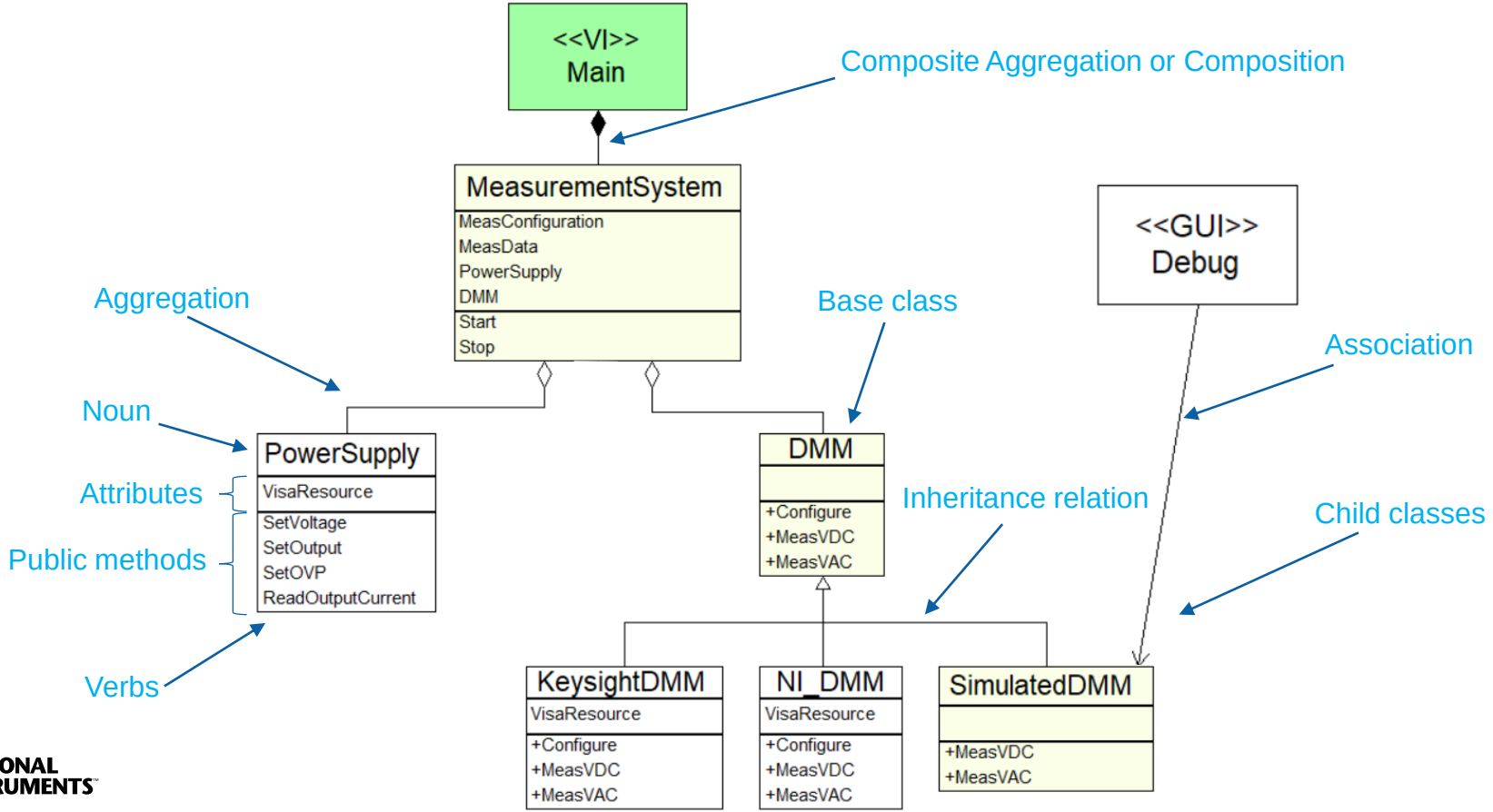


Class diagram – A Measurement System

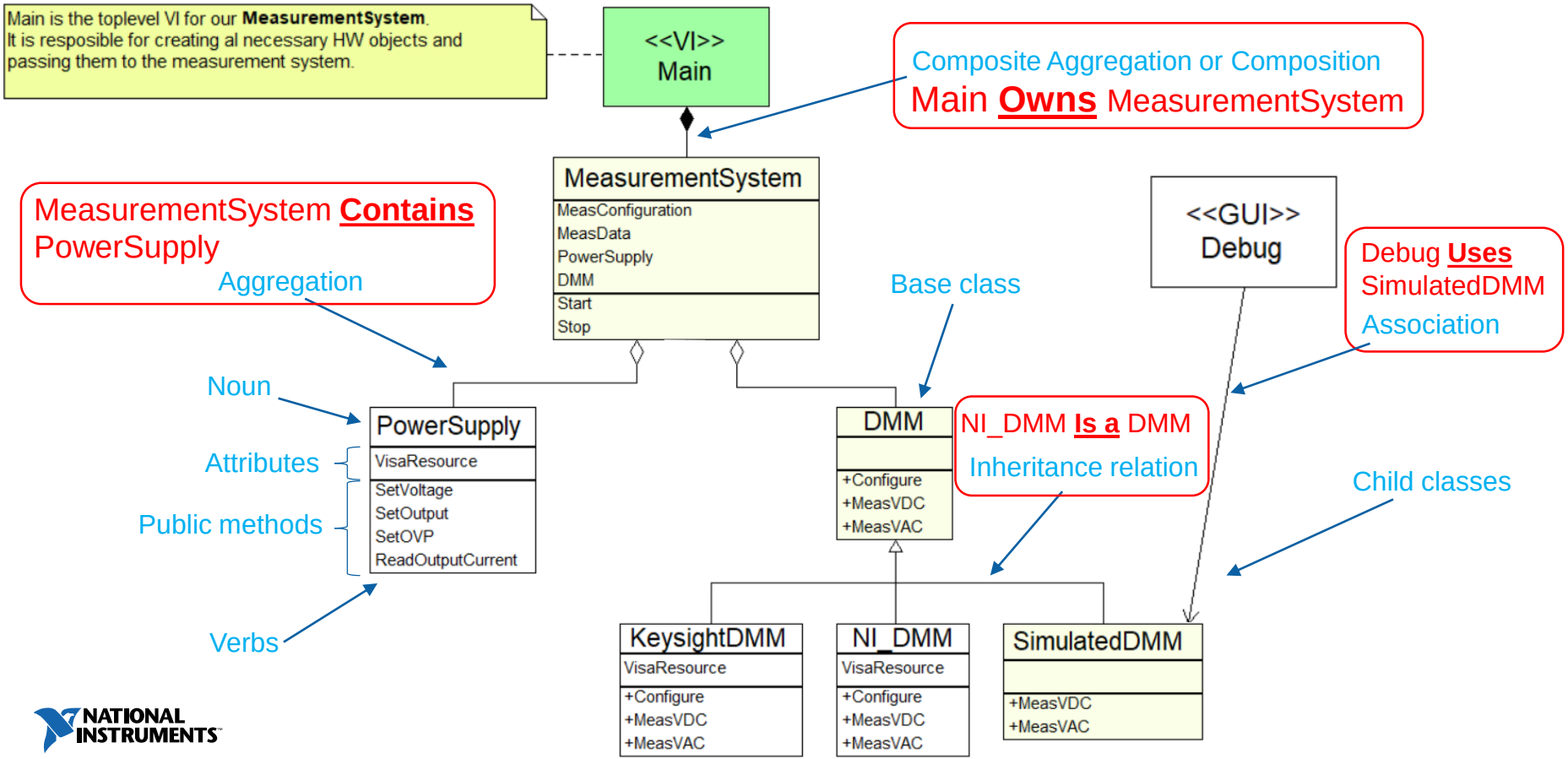


NATIONAL INSTRUMENTS

Class diagram – A Measurement System



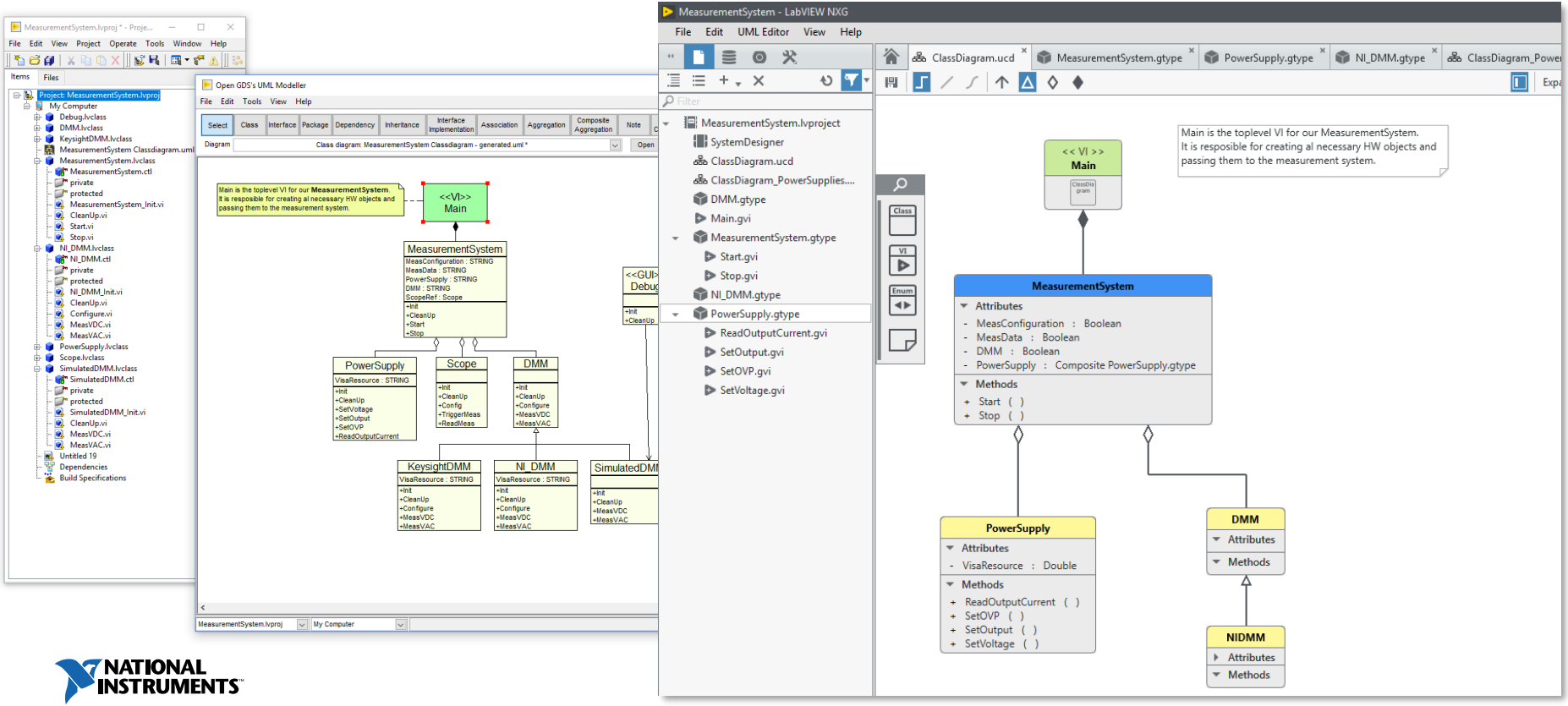
Class diagram – A Measurement System



NATIONAL INSTRUMENTS

OpenGDS

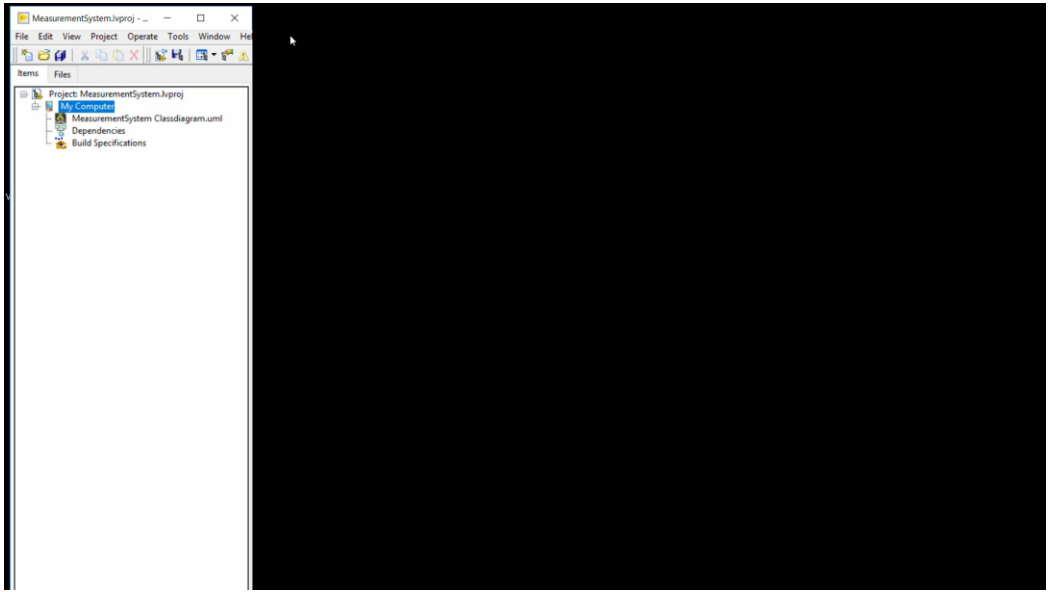
UML Class Editor Add-on for LabVIEW NXG



NATIONAL INSTRUMENTS

OpenGDS Demo

opengds.github.io/ or
http://sine.ni.com/nips/cds/view/p/lang/nl/nid/209038 for the NI supported version.



NATIONAL INSTRUMENTS

UML Class Editor Add-on for LabVIEW NXG

Demo

Stay in the loop!

For state-of-the-art fully integrated UML Class Editing and code synchronization for LabVIEW™ NXG!

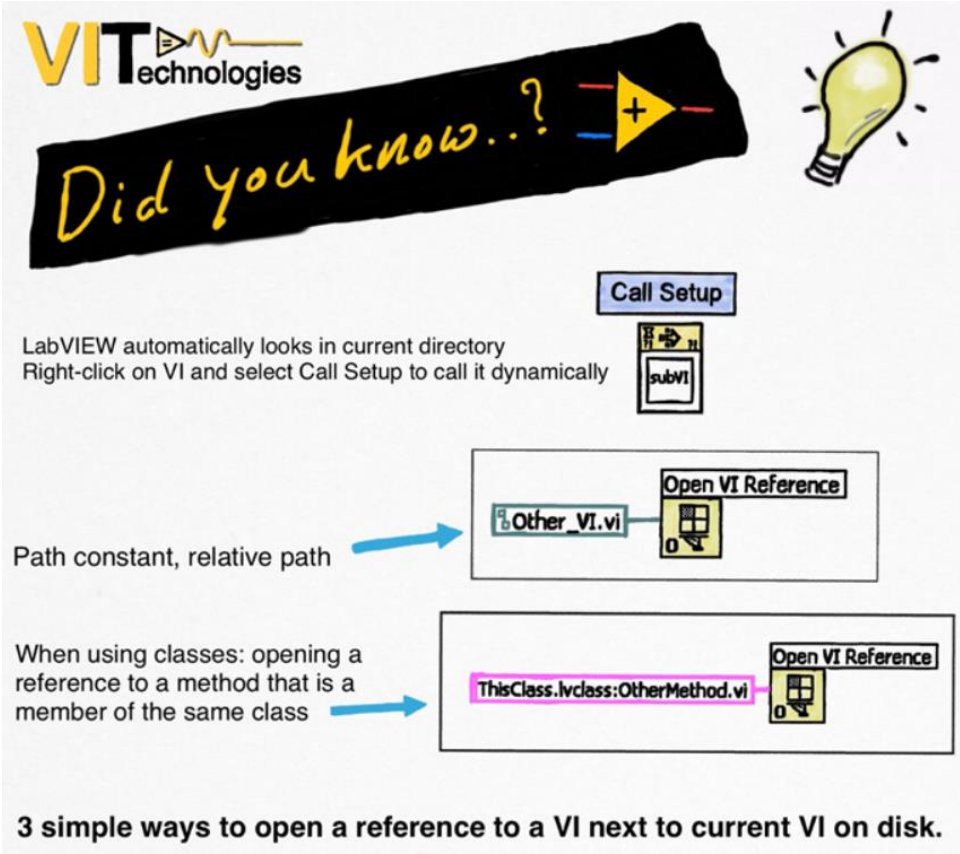
uml-addon.com

NATIONAL INSTRUMENTS

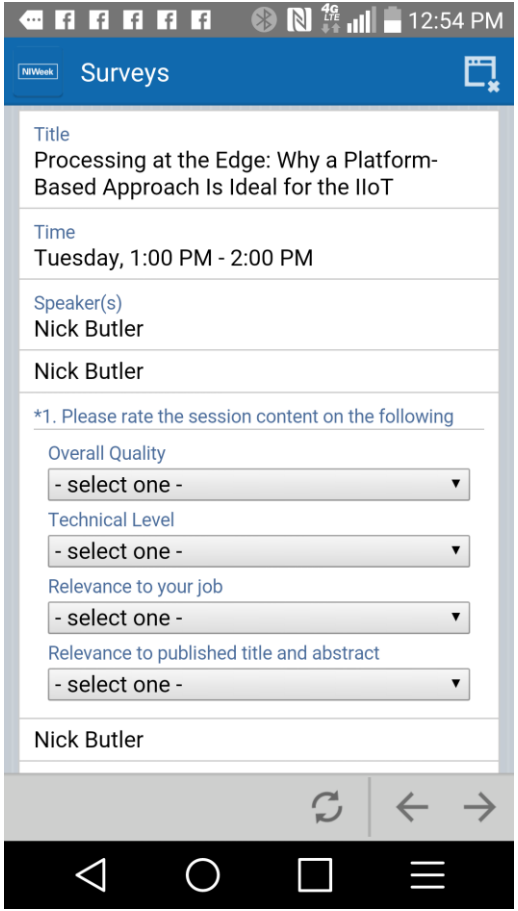
Did you know..?

You can follow us on social

- twitter.com/VI_Technologies
- linkedin.com/company/vi-technologies/
- facebook.com/vitechnologies/



NATIONAL INSTRUMENTS



Before you go, take the survey.

NATIONAL INSTRUMENTS

Stay Connected During and After NIWeek

ni.com/niweekcommunity

facebook.com/NationalInstruments

twitter.com/niglobal

youtube.com/nationalinstruments

Please provide feedback on this session via the NIWeek Mobile App