

大型LabVIEW应用程序 开发技巧及案例剖析

张南雄

上海其高总经理

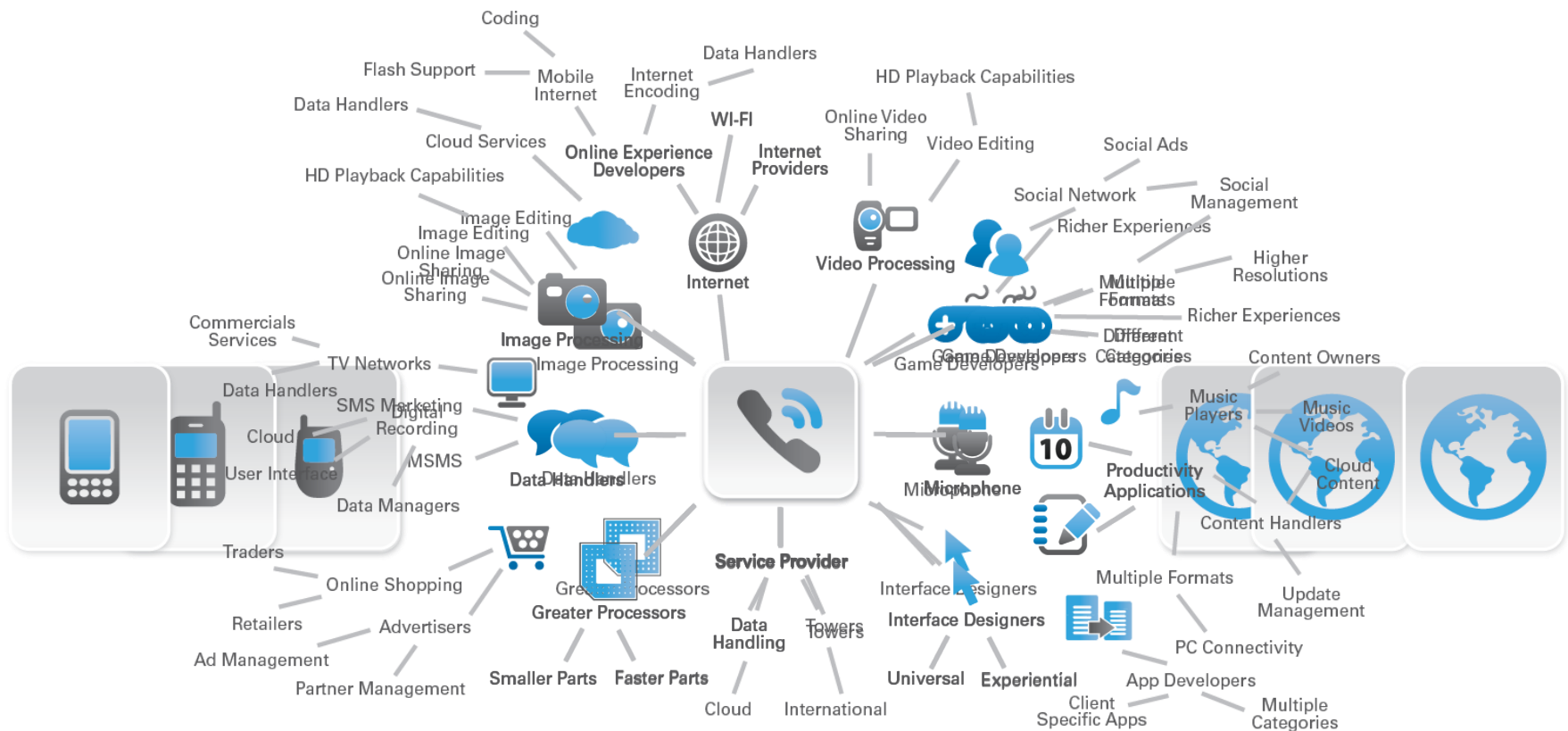
Certified LabVIEW Architect

当前工程应用面临巨大挑战

- 功耗要求最小化
- 竞争和项目期限
- 将更加复杂的产品快速推向市场
- 最大限度提高工作效率
- 更少资源、更少预算，更多项目
- 永不停止的技术变革
- 适应不断变化的应用需求
- 代码及系统集成
- 分布式团队间的协作
- 传统设备到系统集成



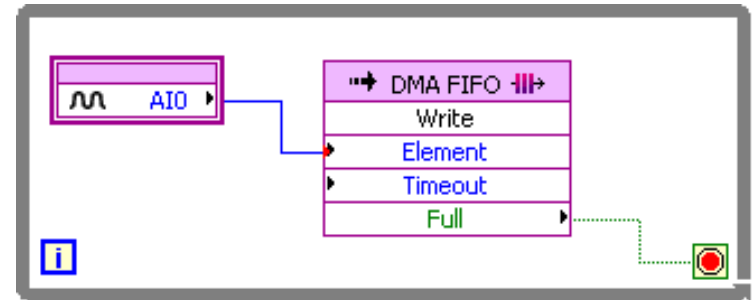
系统复杂度不断升级



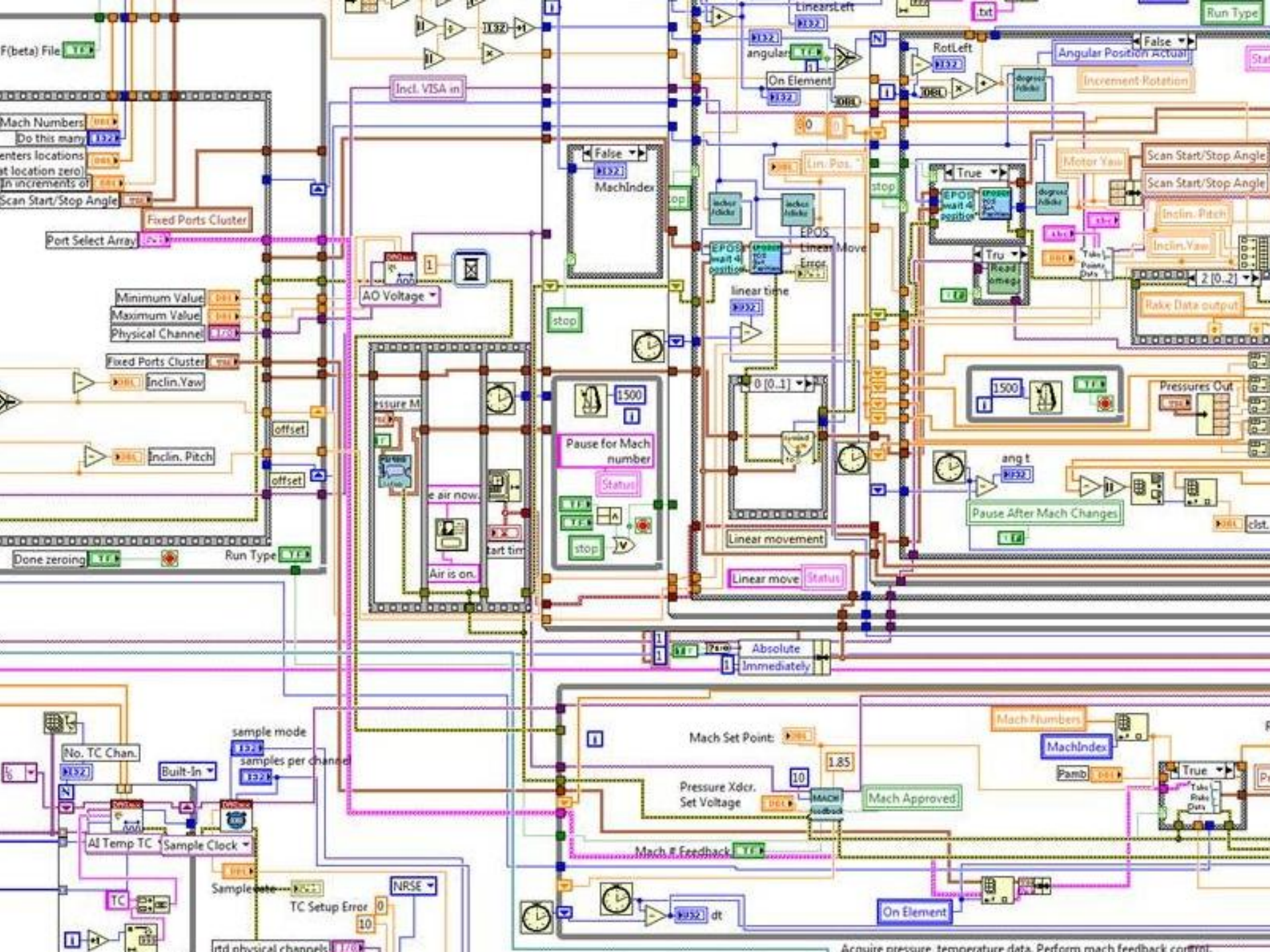
图形化设计简化开发流程



VHDL



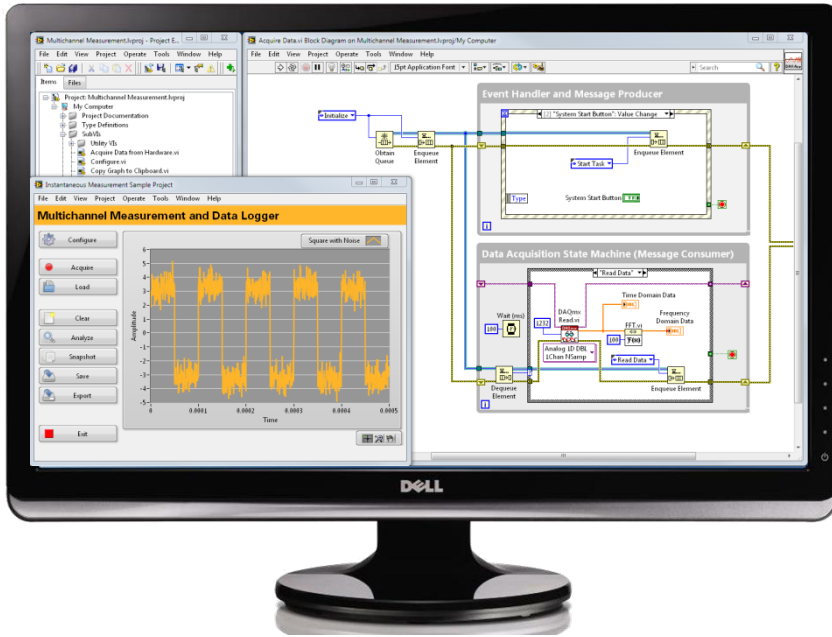
LabVIEW FPGA



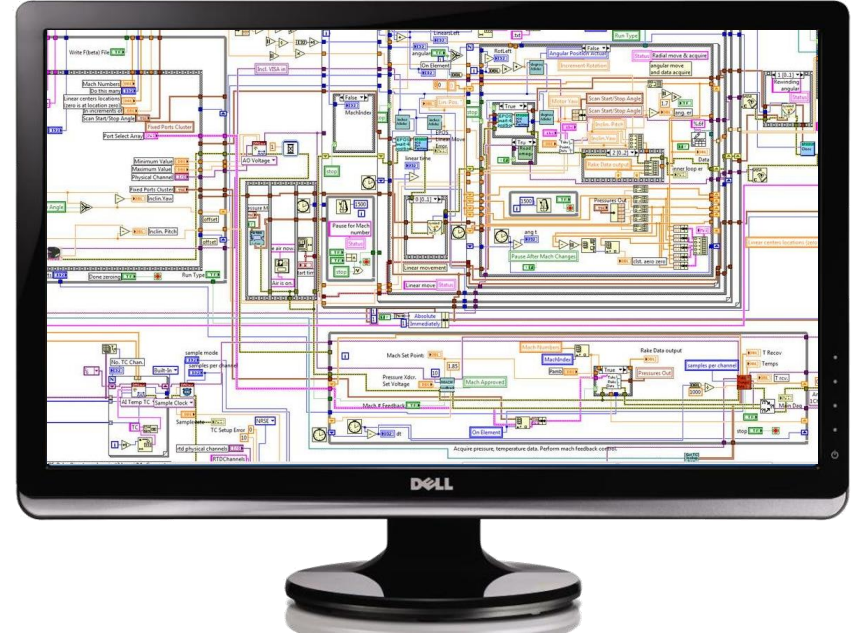


NATIONAL INSTRUMENTS

LabVIEW™



Build This

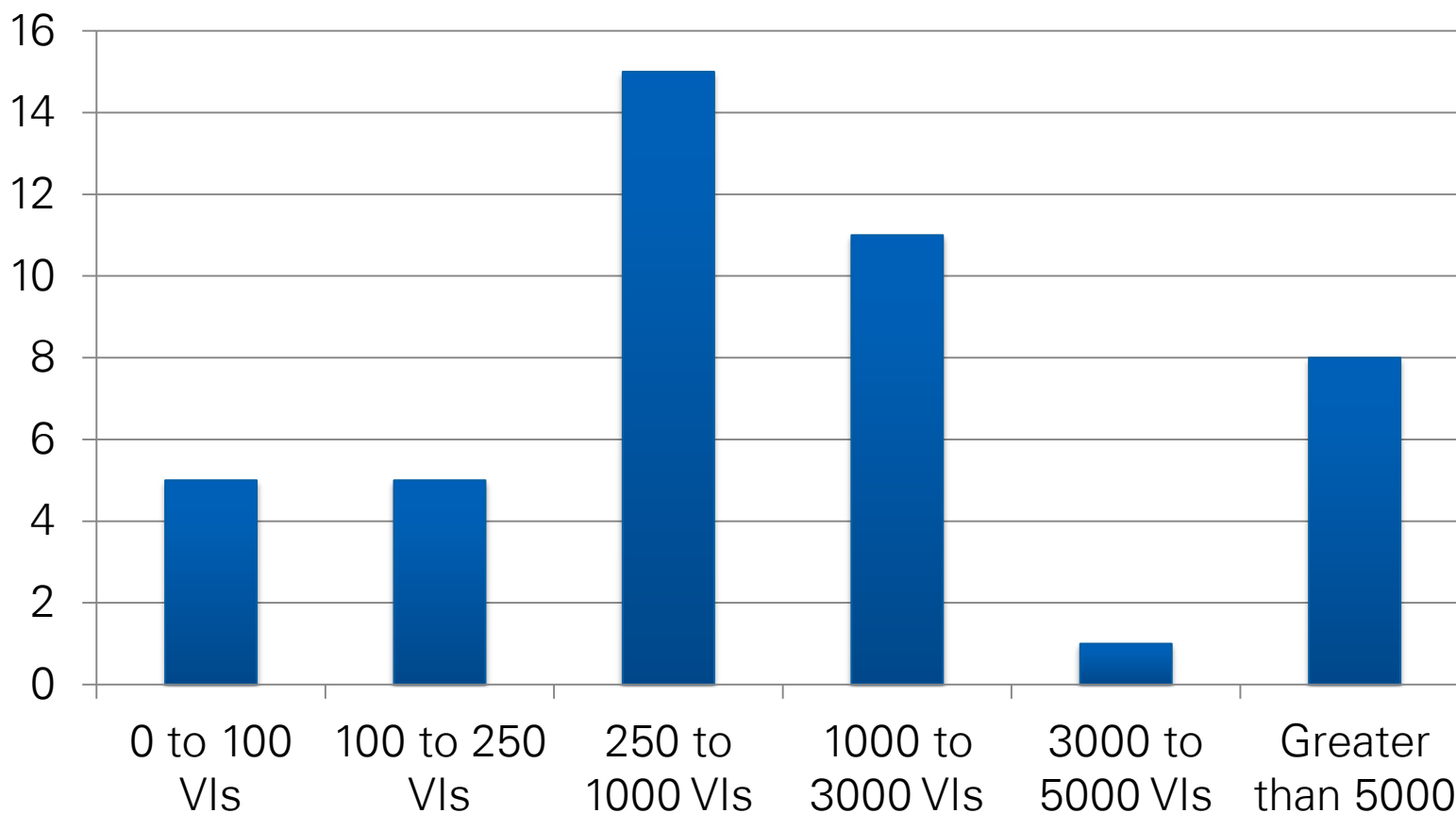


Not That

内容提要

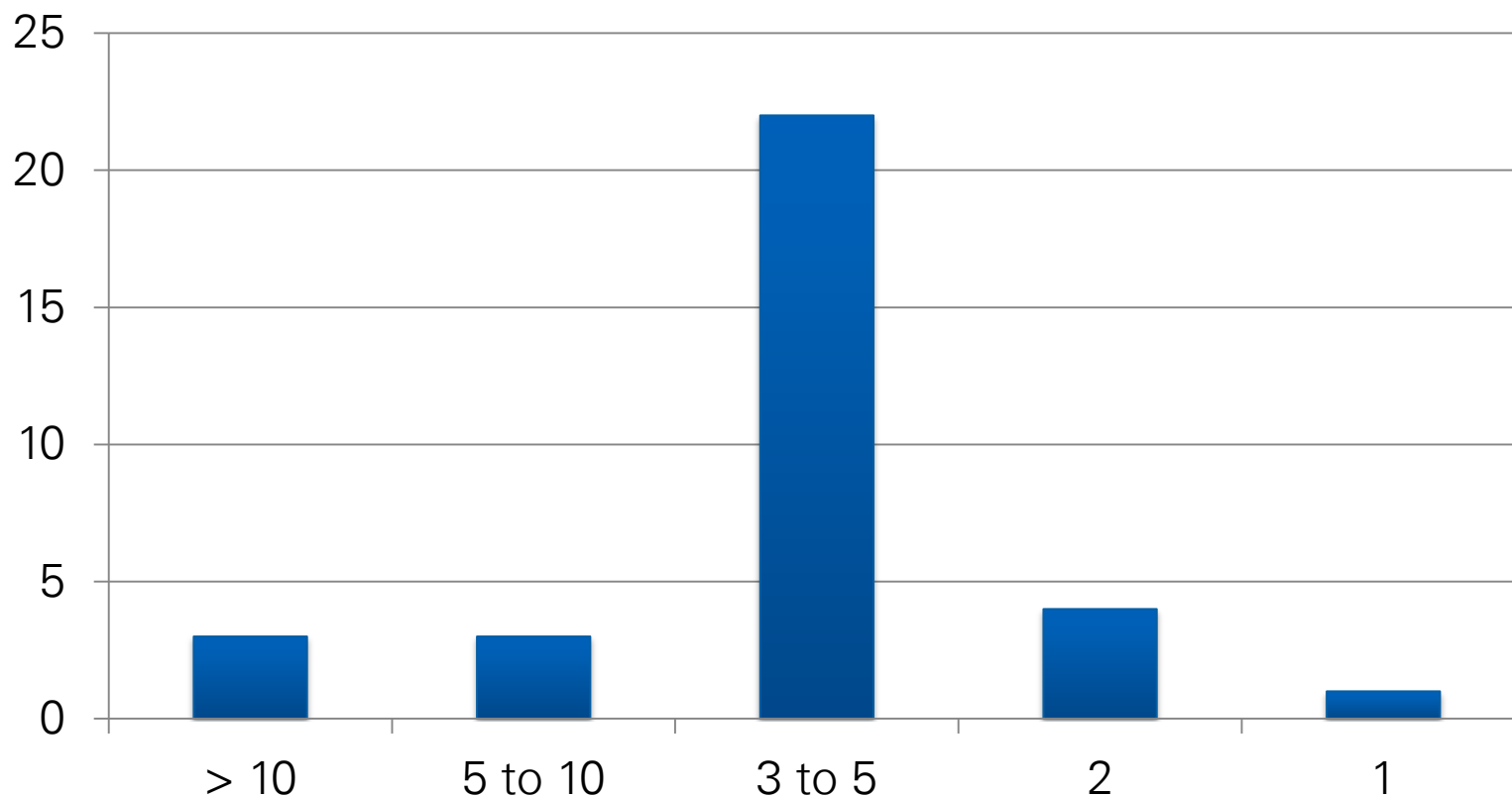
- 大型LabVIEW应用程序开发的挑战与应对措施
 - 团队合作
 - 程序架构
 - 数据传递、接口
 - Code reuse/并行开发
 - 发布与维护
- 案例分析 – 其高SignalPad测控软件
 - 功能介绍
 - 架构设计与实现
- 开发工具与团队合作
 - 源代码管理
 - 测试与缺陷跟踪
 - 软件测试与发布
- 若干LabVIEW实用技巧

LabVIEW应用程序大小



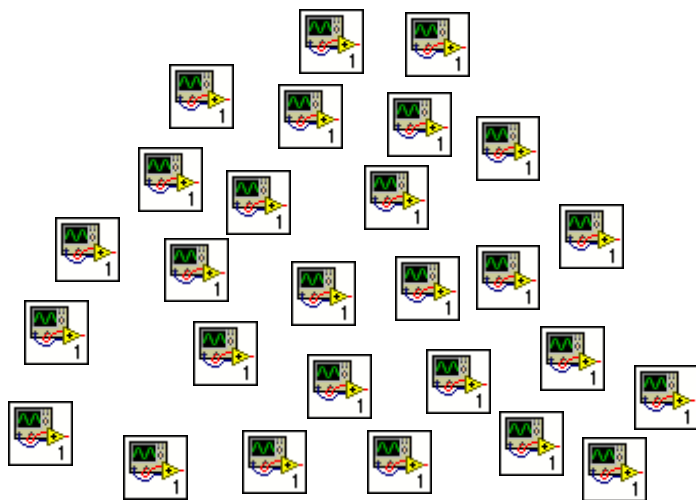
Source: 2010 ni.com/largeapps survey

LabVIEW应用程序开发团队人数

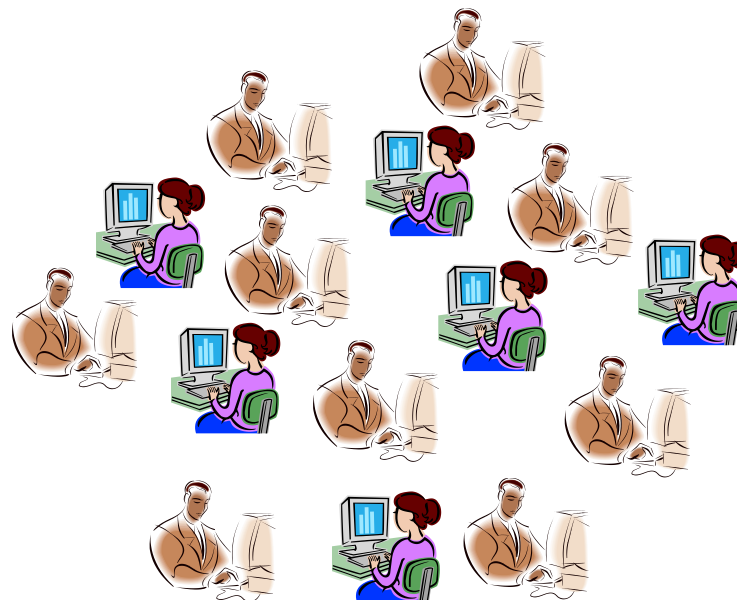


Source: NIWeek 2008 Software Engineering Survey

LabVIEW大型应用程序特点



VI数量多



团队工作

LabVIEW大型应用程序的挑战

- 软件工程

- 团队合作
- 需求分析
- 测试与质量控制
- 发布与维护

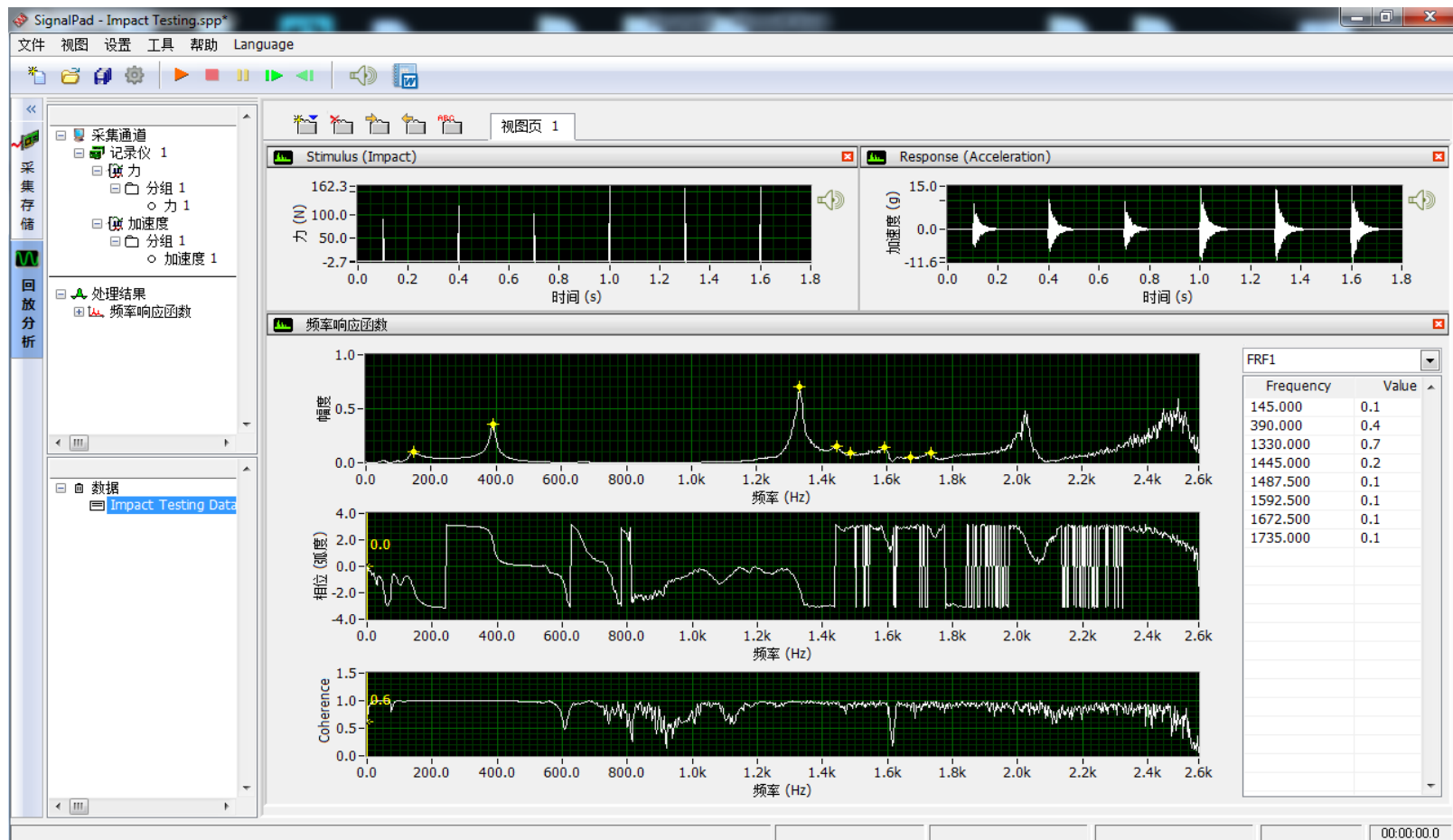
- 编程实现

- 程序架构
- 数据传递与接口
- 性能与用户体验
- 项目管理

LabVIEW大型应用程序开发原则

- 采用模块化架构
 - 适合团队合作开发
 - 减少功能模块间的耦合和依赖，使得并行开发成为可能
 - 加强程序的可扩展性和可维护性
- 积极使用开发工具
 - 源代码控制
 - 缺陷跟踪
 - 调试工具
- 重视开发文档

SignalPad测控软件



关于其高

- 概要信息

- 2009年11月正式运营
- NI系统联盟商
- 5名LabVIEW工程师
- 2名CLA和1名CLD

- 主要业务

- 数据采集：大通道、高采样率、车载
- 振动、噪声、模态、麦克风阵列、ANC



其高数据采集与振动噪声测试解决方案

交钥匙完整解决方案



SignalPad

测控软件

- 基于配置，无需编程
- 数据采集、存储、回放
- 更多振动噪声测试功能

NI DAQ



SignalPad应用案例

- 车载数据记录

- 振动、温度、压力、速度、应变



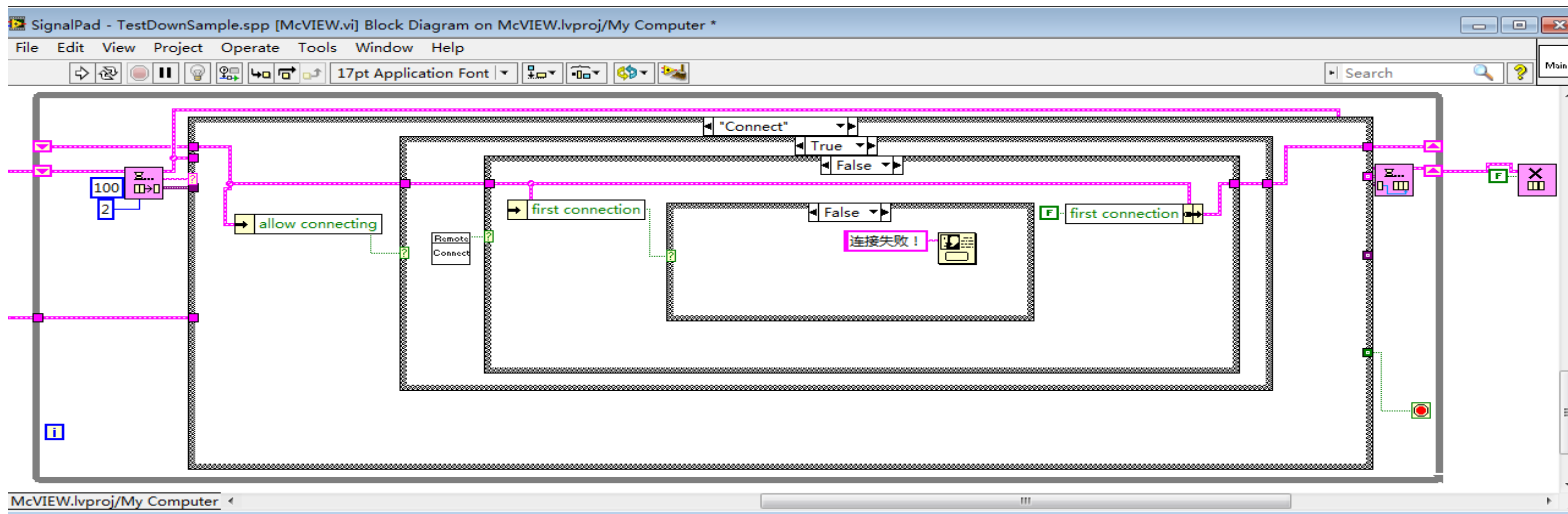
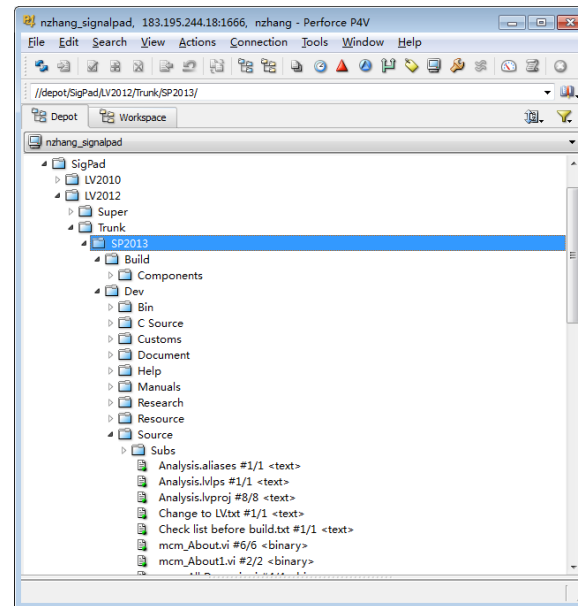
- 风力发电机组噪声测试

- 风速、功率、噪声

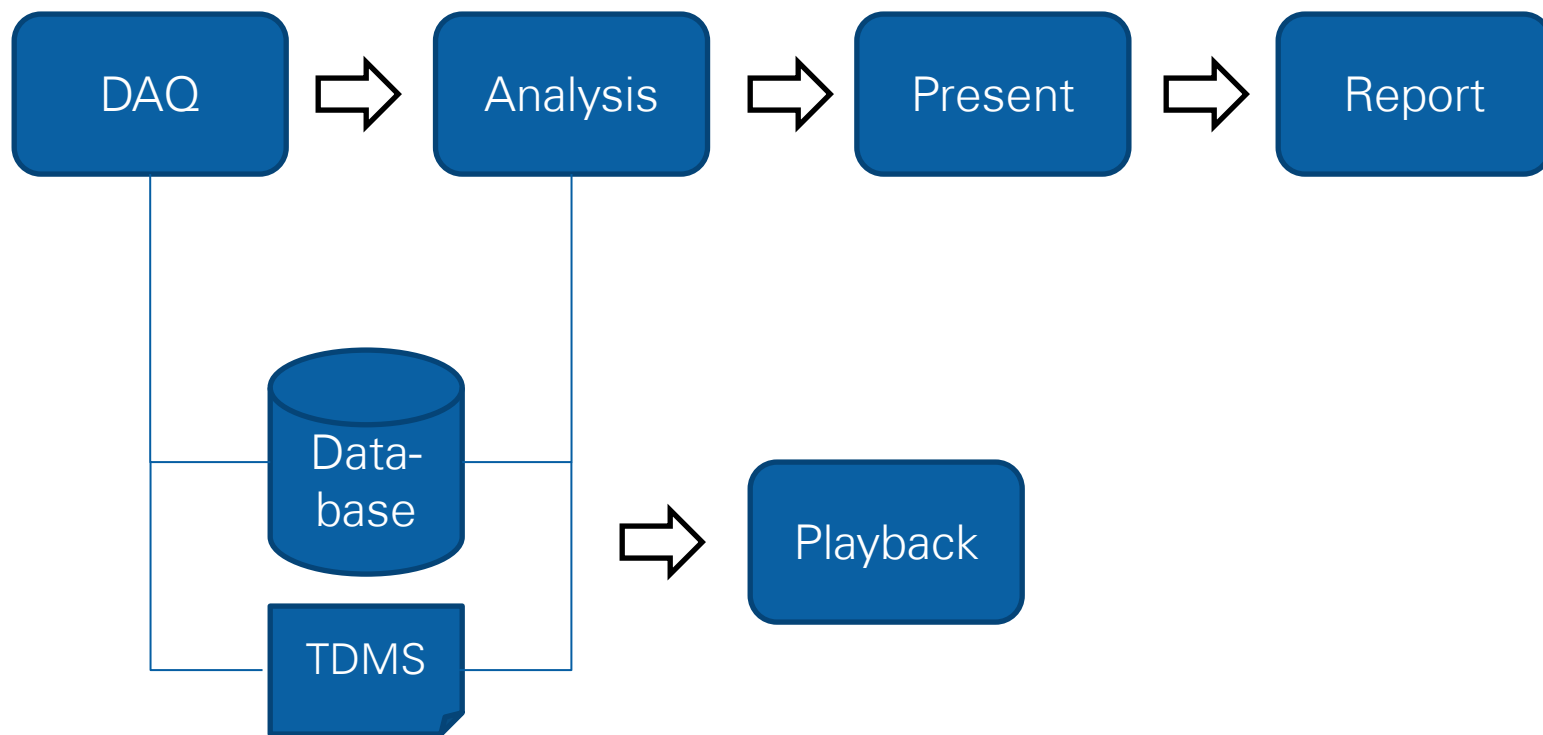


SignalPad开发

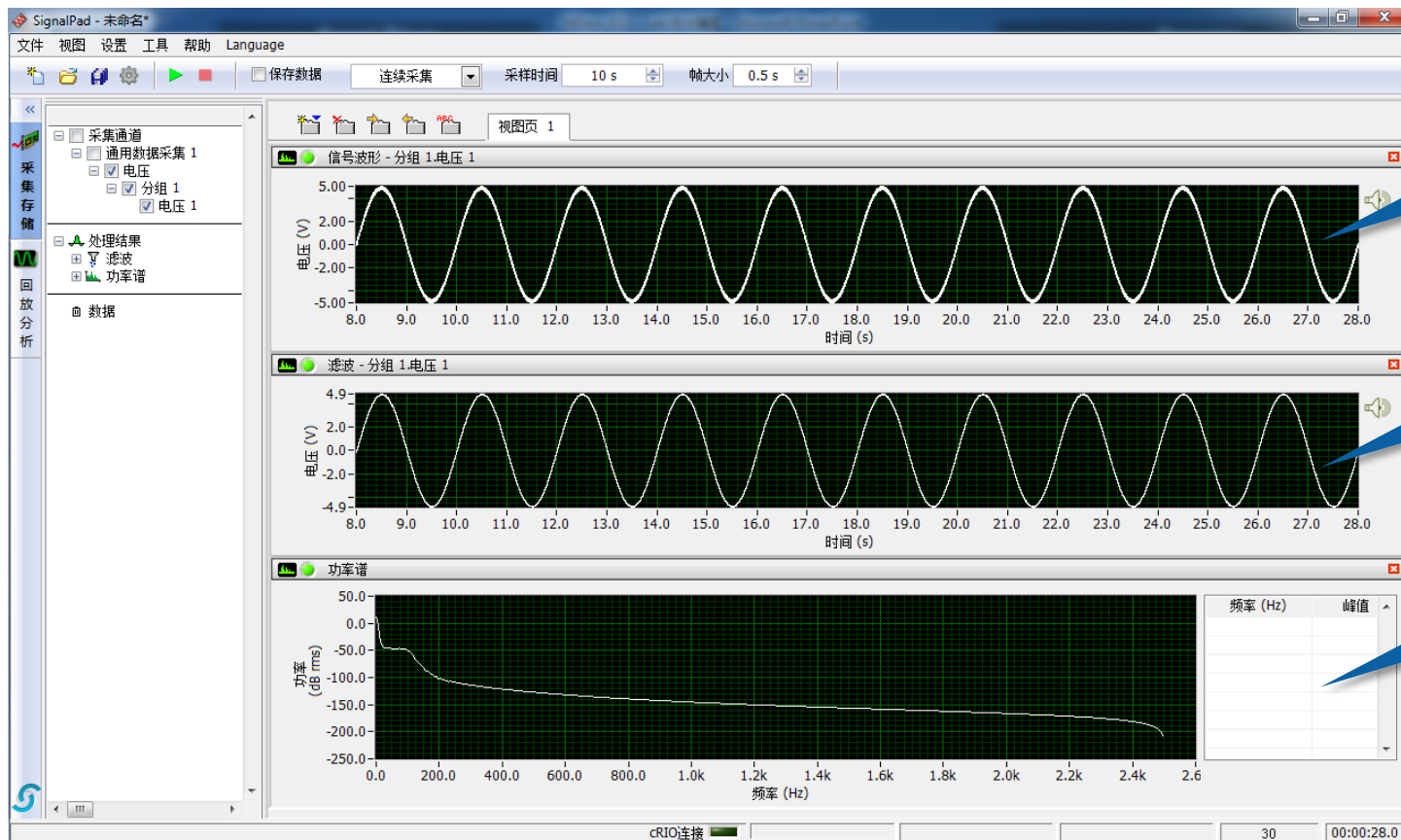
- 程序
 - >10,000个文件
 - ~1GB
- 开发人员
 - 5名工程师已经投入近10人年



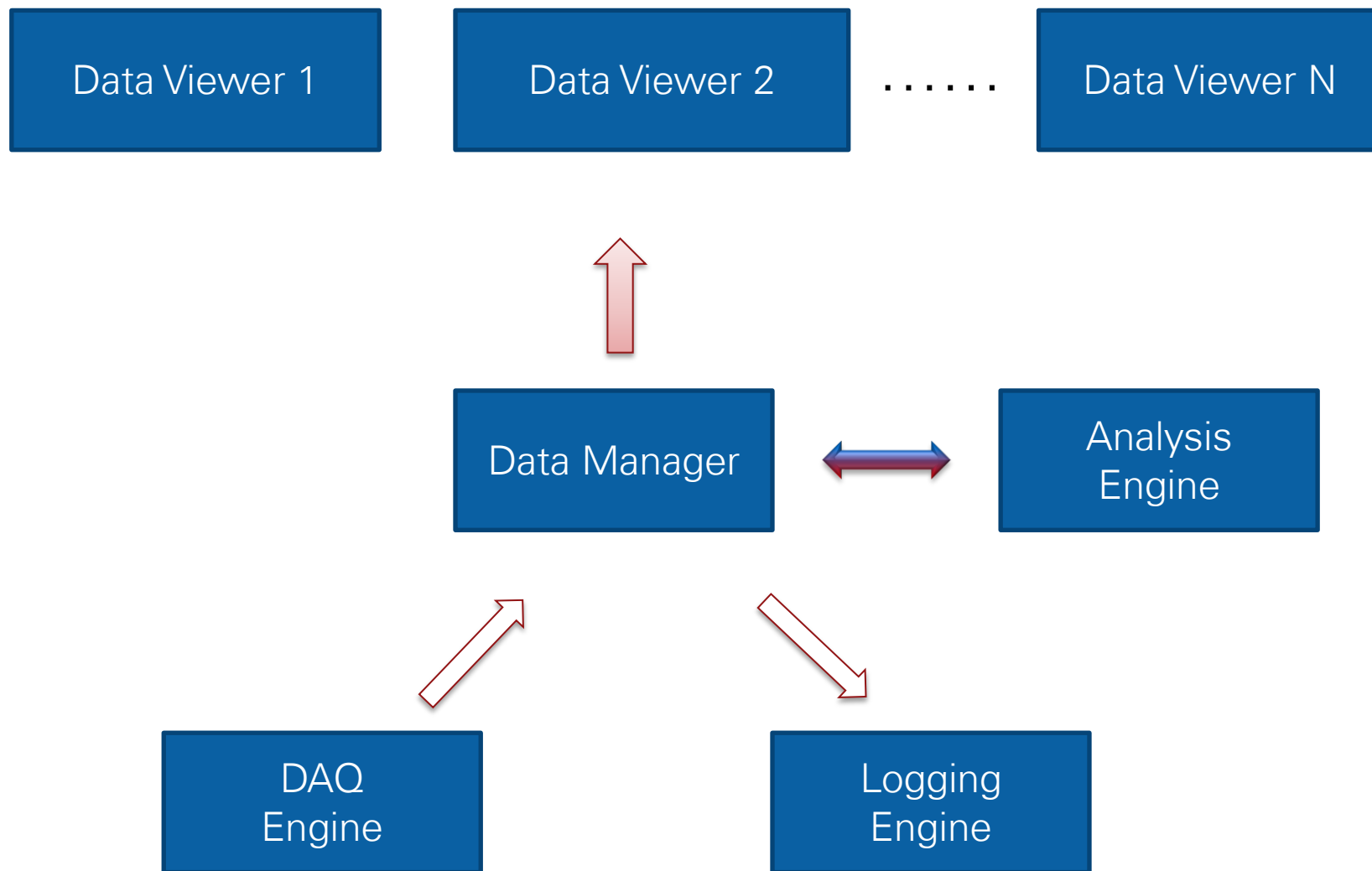
程序功能



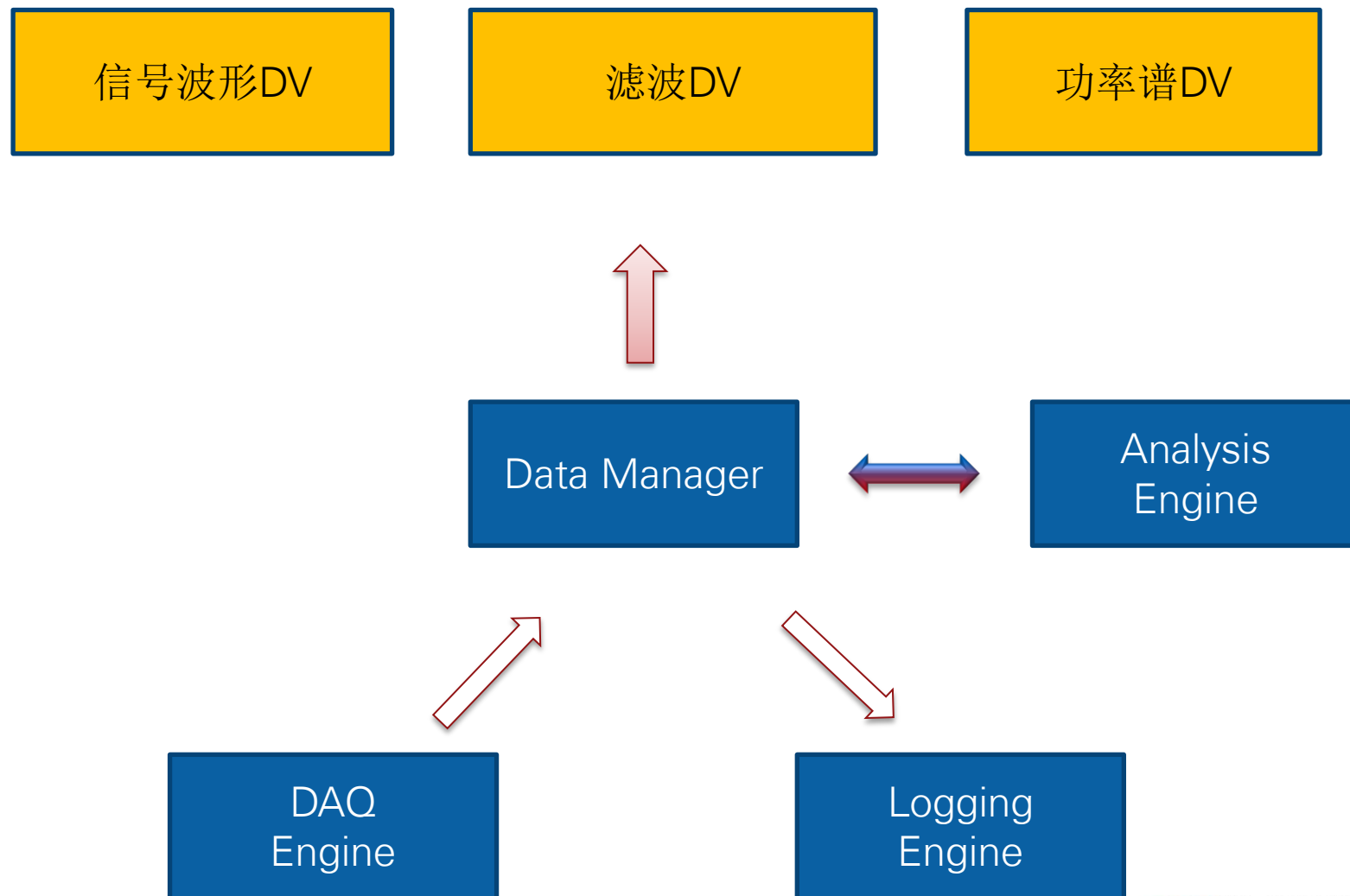
执行机制



执行机制

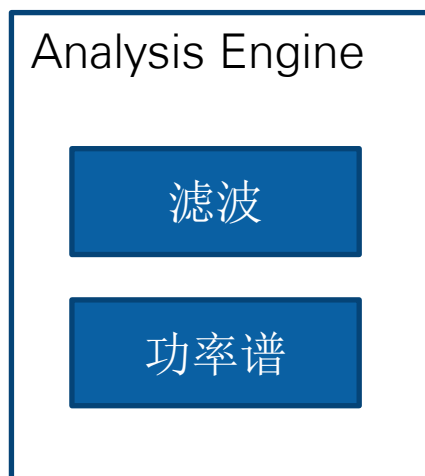


SignalPad执行机制举例



SignalPad执行机制举例

- Analysis Engine - 根据视图配置确定信号处理步骤



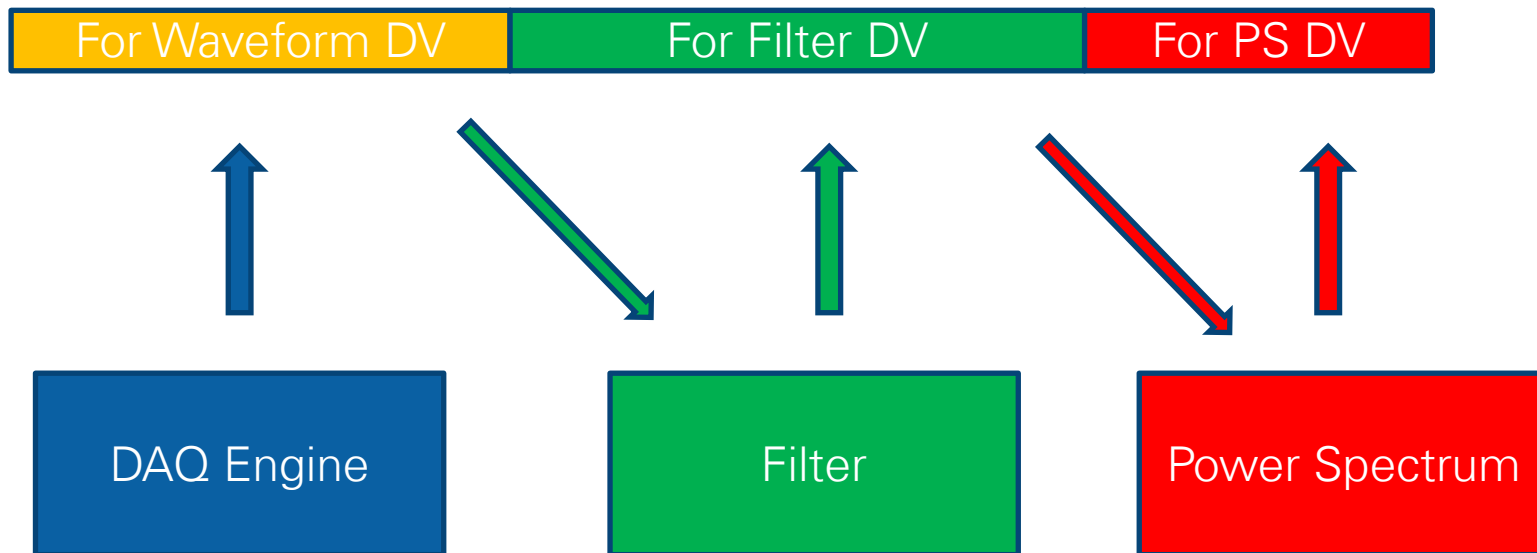
- Data Manager - 根据视图配置确定存储空间

Data Manager分配的存储空间



SignalPad执行机制举例

Data Manager分配的存储空间



SignalPad执行机制举例

Data Manager分配的存储空间



信号波形DV



滤波DV

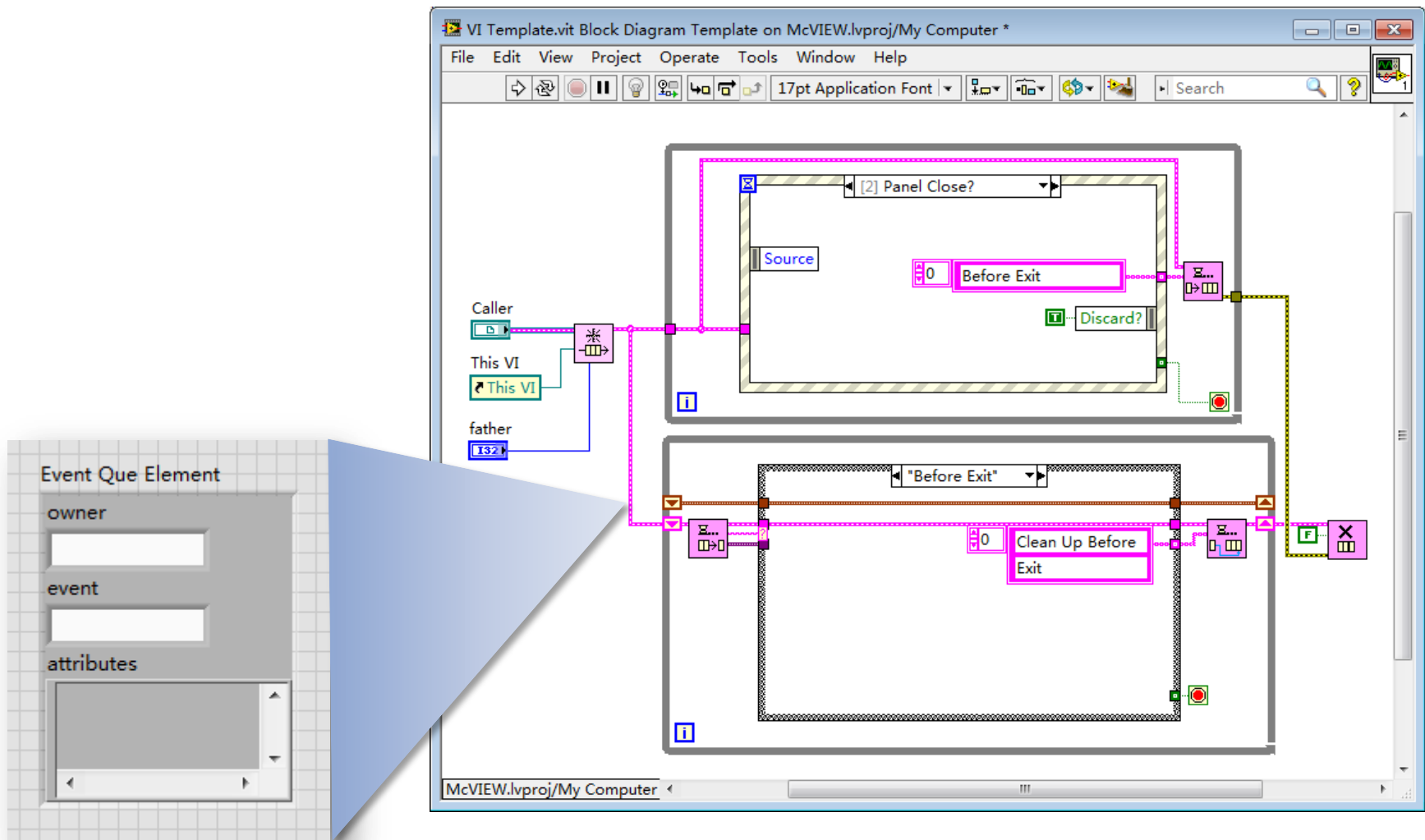


功率谱DV

SignalPad程序基本架构

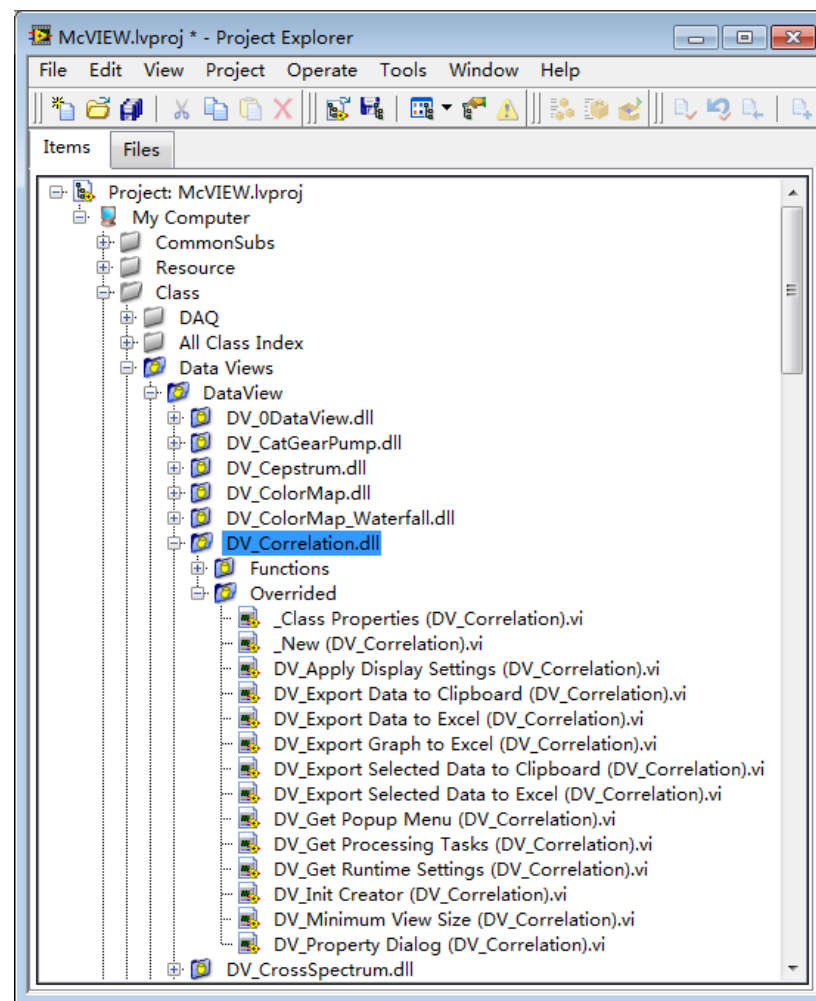
- 生产者消费者
- 插件式架构 (Plug-in)
- 子面板 (Subpanel)

SignalPad生产者/消费者结构



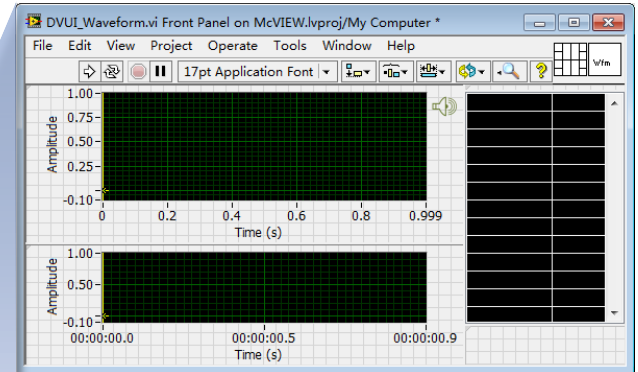
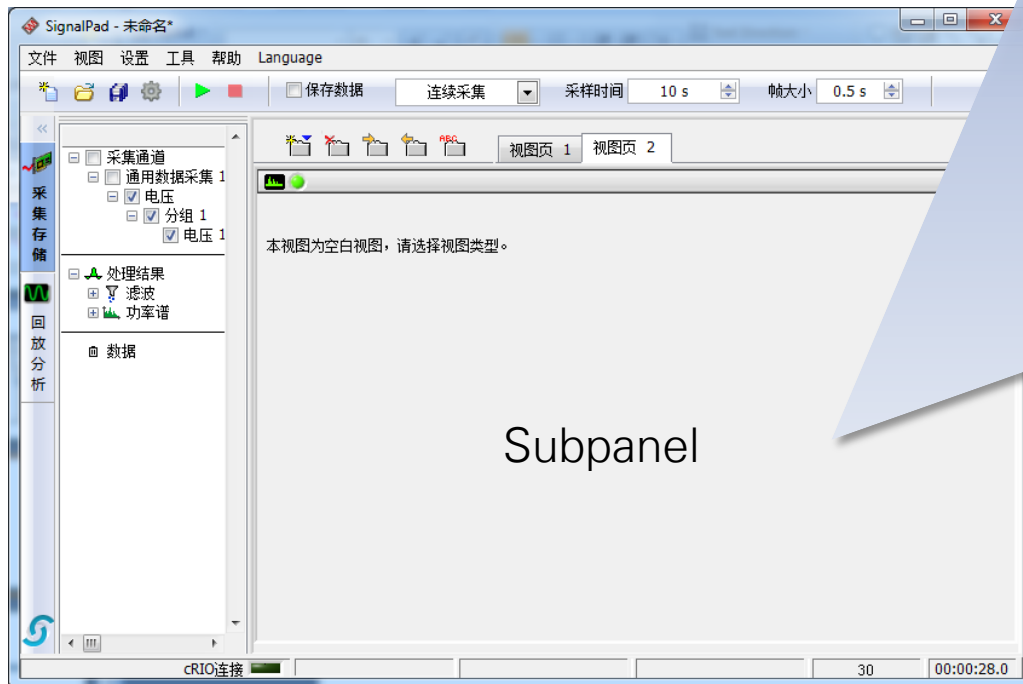
SignalPad插件式结构

- 视图采用插件式结构
- 增加一个视图
 - 增加一组接口固定的VI
 - 增加一个配置文件
 - 无需修改主程序
- 插件式结构实现方法
 - 使用Variant



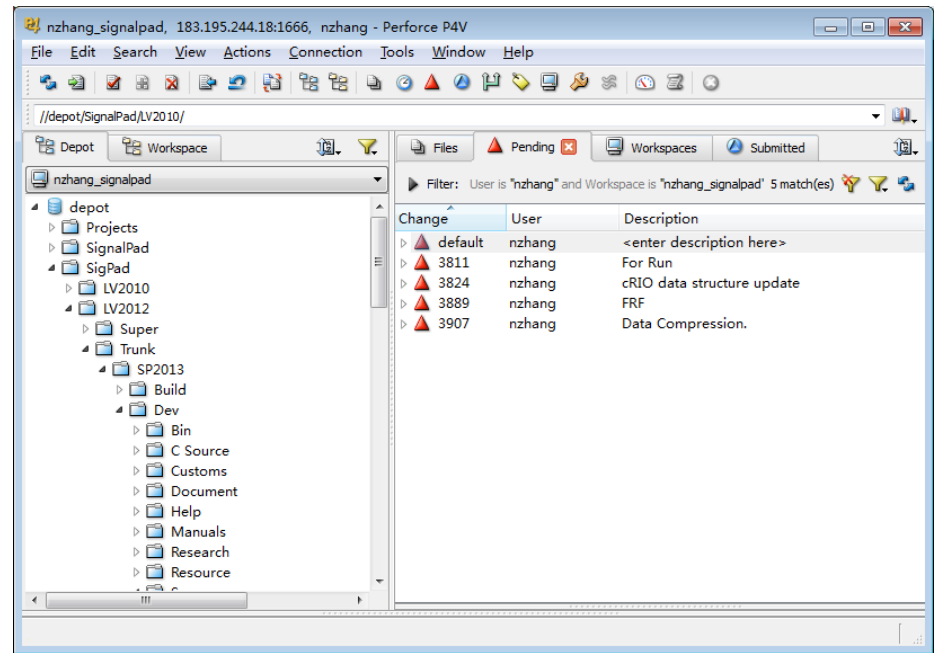
子面板Subpanel

- SignalPad大量使用Subpanel
 - 执行效率高
 - 实现动态UI
 - 重用UI （替代XControl）



SignalPad源代码管理

- Perforce
 - 20个workspace以下免费
 - LabVIEW内嵌支持
- 源代码管理工具的作用
 - 团队协调工作
 - 版本管理
 - 代码备份
 - 代码回溯



SignalPad缺陷跟踪

- Redmine
 - 免费
 - 跟踪bug
 - 任务管理 (Gannt)

The top screenshot shows the Redmine 'My page' view. It displays a list of issues assigned to the user and a list of reported issues.

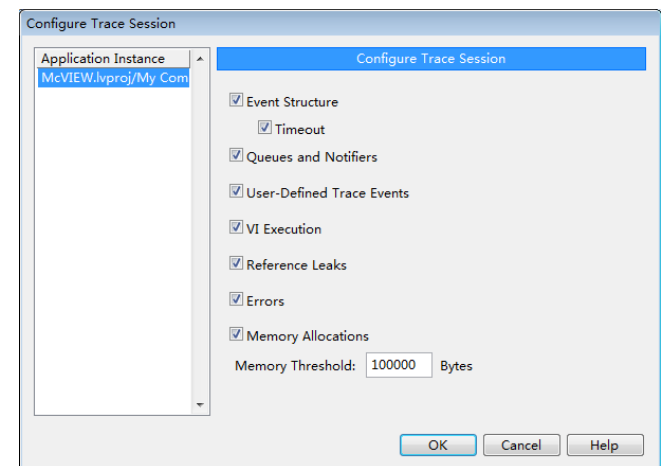
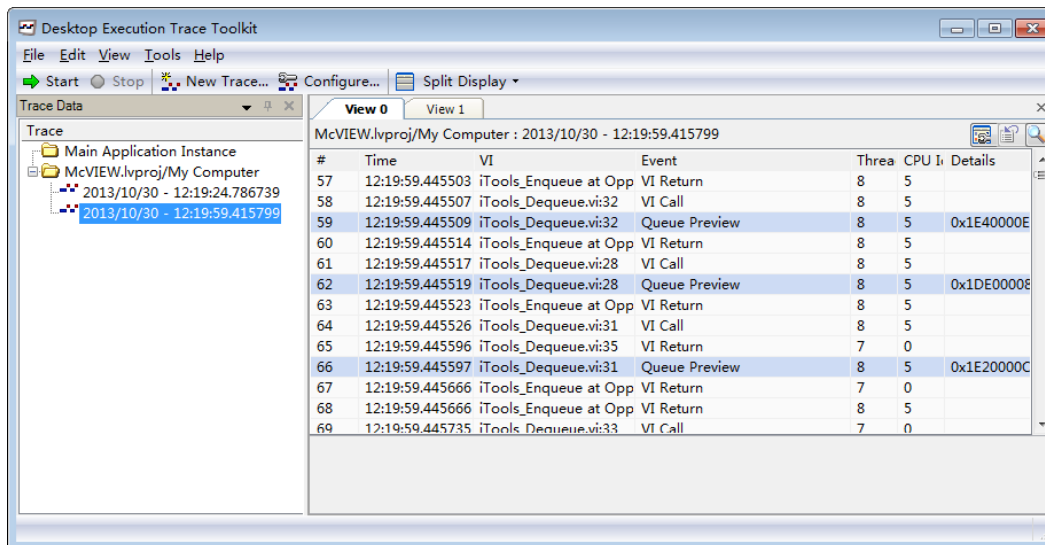
#	Project	Tracker	Subject
102	视图 (DataView)	Bug	Resampling DV returns "memory size mismatched" error when ... (New)
93	数据采集 (DAQ)	Bug	位于两个机箱中的两块9205同时采集出错 (Resolved)
143	Sensor Settings(传感器设置)	Bug	Can not add new sensor (New)
222	Menu(菜单)	Bug	no help doc shows up in English Version (New)
186	杂项 (Utilities)	Bug	MCM特征参数表格不随信号选择变化 (New)
168	Correlation (相关分析)	Bug	删除通道的颜色没有清除 (Accepted)
177	Numeric	Bug	configtree_Get_All_Measurement.v输出只包含采集通道的名字, 不包含其他DVUI处... (New)
176	架构 (Framework)	Bug	Mix Table在使用colorbox的时候, colorbox控件上会显示color字符 (New)
174	架构 (Framework)	Bug	Mix Table的对应行已经删除, 但是table仍然保留color列的颜色 (New)

The bottom screenshot shows the Redmine 'Gantt' view for a specific project. It displays a Gantt chart with tasks and their progress.

Task	2013-10	2013-11	2013-12
SignalPad			
Feature #777: Wrong project			
Menu(菜单)			
Bug #220: no help doc shows up in English Version			
定制功能 (Customized)			
Windturbine Report Generator			
Feature #19: 增加增加功能允许客户指定公司标志			
Bug #88: 查询信息系统标题与内容不符			
Bug #90: 查询信息文件log信息是7-12			
Bug #100: 增加增加Correlation Coefficient			
Bug #136: 报表应该能够输出到Excel			
Bug #215: 分析程序应该能够输出log file			
Feature #216: 增加增加数据列表功能			
Bug #217: 查询分析功能			
Feature #218: 增加增加数据列表功能			
数据采集 (DAQ)			
Bug #93: 位于两个机箱中的两块9205同时采集出错			
Feature #222: General IEPE Channel Needed			
Feature #223: General Channel associated for order...			
杂项 (Utilities)			
Bug #45: Error Report for Channel Configuration is E...			

内存占用跟踪

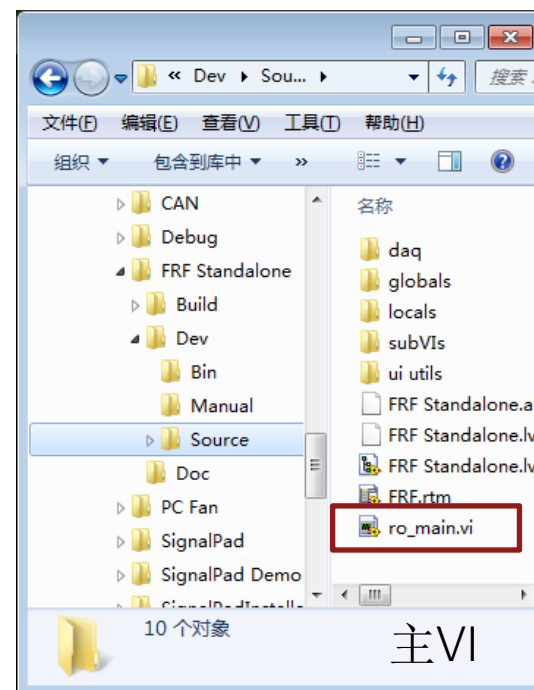
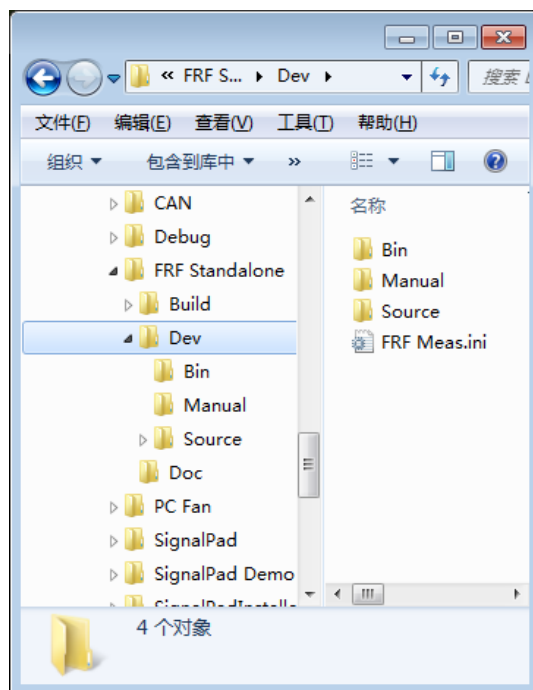
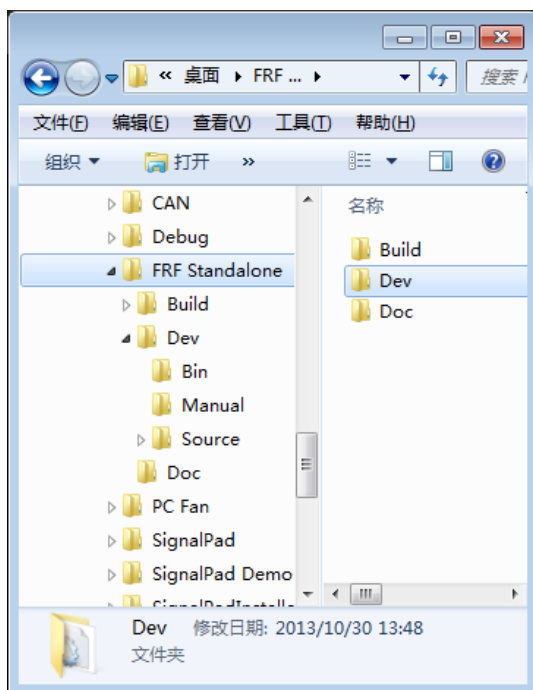
- LabVIEW Desktop Execution Trace Toolkit
 - 定位内存消耗过大的VI或内存泄露
 - 查找文件或VI指引（reference）泄露



LabVIEW编程技巧

• 推荐目录结构

- VI和文件在发布EXE时相对路径应保持不变
- VI中使用相对路径

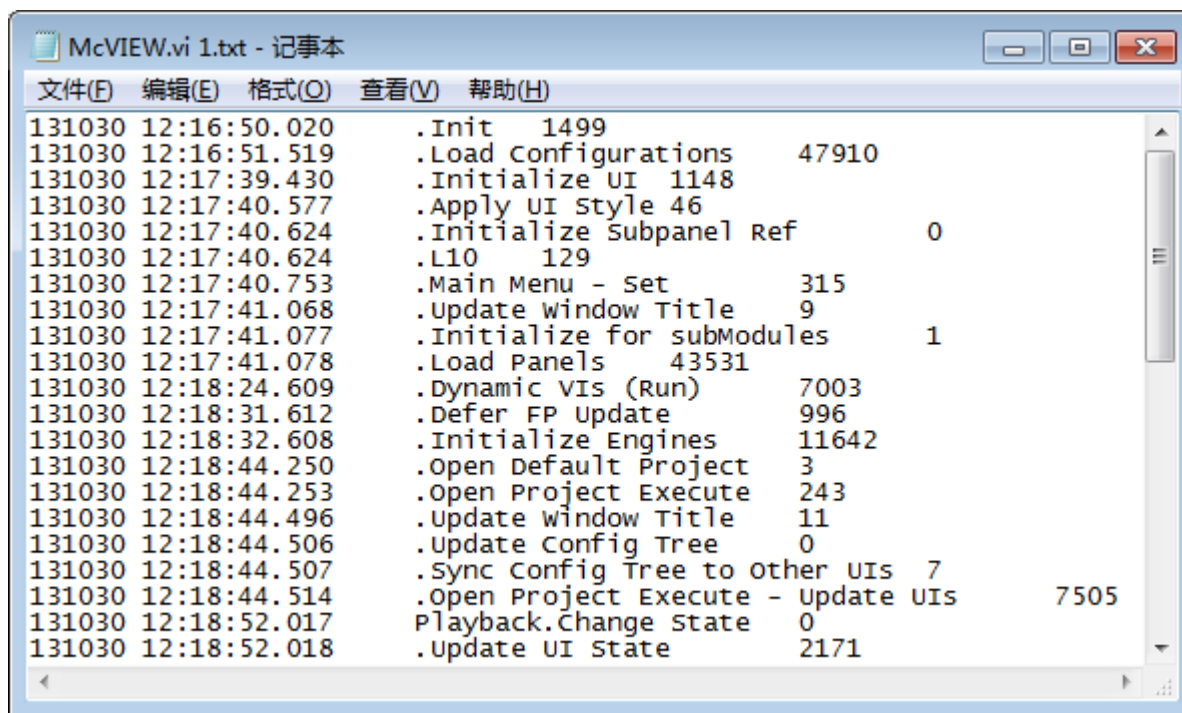


LabVIEW编程技巧

- 配置信息存储
 - 程序启动时自动加载上次关闭时的设置
 - 常见方法 – ini配置文件
 - 优点：直观，可直接修改
 - 缺点：效率稍低，配置内容太多时速度明显变慢
 - 可选方法 – Variant
 - 优点：效率很高；可存储较为复杂配置信息
- Variant使用技巧
 - 存储版本信息
 - 利用Variant的属性存储信息

LabVIEW编程技巧

- 如何调试使用生产者/消费者的VI
 - 记录每个消费者循环的执行顺序和执行时间，帮助了解执行顺序以及费时的分支



```
McVIEW.vi 1.txt - 记事本
文件(F) 编辑(E) 格式(O) 查看(V) 帮助(H)
131030 12:16:50.020 .Init 1499
131030 12:16:51.519 .Load Configurations 47910
131030 12:17:39.430 .Initialize UI 1148
131030 12:17:40.577 .Apply UI style 46
131030 12:17:40.624 .Initialize Subpanel Ref 0
131030 12:17:40.624 .L10 129
131030 12:17:40.753 .Main Menu - Set 315
131030 12:17:41.068 .Update window Title 9
131030 12:17:41.077 .Initialize for subModules 1
131030 12:17:41.078 .Load Panels 43531
131030 12:18:24.609 .Dynamic VIs (Run) 7003
131030 12:18:31.612 .Defer FP update 996
131030 12:18:32.608 .Initialize Engines 11642
131030 12:18:44.250 .Open Default Project 3
131030 12:18:44.253 .Open Project Execute 243
131030 12:18:44.496 .Update window Title 11
131030 12:18:44.506 .Update Config Tree 0
131030 12:18:44.507 .Sync Config Tree to other UIs 7
131030 12:18:44.514 .Open Project Execute - update UIs 7505
131030 12:18:52.017 Playback.Change State 0
131030 12:18:52.018 .Update UI State 2171
```

总结

- LabVIEW可以用来开发大型应用程序
 - 使用一致的程序编程模式（如生产者/消费者）
 - 插件式结构有利于团队成员并行开发
 - 使用正确的工具
 - Perforce
 - Redmine
 - VMWare
 - LabVIEW Desktop Execution Trace Toolkit

Thanks