FROM REQUIREMENTS TO TESTING, VALIDATION AND VERIFICATION

Patricia Derler, National Instruments

CPS V&V I&F Workshop 2017
April 2017
State of the Art

• Systems are becoming more complex, distributed, heterogeneous

• Tighter time-to-market requirements

• “Design by/for me”
  The real design challenge won’t be coming up with a perfect object, but a perfect process: a reliable way of making thousands of variations on a product, quickly and accurately, and giving customers influence over the outcome at a fundamental level.

https://hbr.org/2013/11/welcome-to-the-designed-by-me-era
V&V in Industry

... often refers to testing

At the manufacturing stage, solar panels are inspected and tested to ensure the quality of the panels. Machine vision inspection identifies small manufacturing defects, and compliance testing ensures the solar panels conform to international quality standards and will survive in an outdoor environment for their expected long lifetimes. **Solar validation and verification tests involve mechanical and electrical measurements** such as stress, temperature, voltage, current, and moisture. Test applications include thermal cycle, accelerated life cycle, electrical connector, mechanical impact, fatigue, and nondestructive test techniques such as ultrasonic testing.

**Verification** requires one or more design **documents** or drawings to govern what the system must accomplish. The specification and test methodology for verification must be a thoroughly detailed document with as much information as necessary to create a correct test system.

Designing a **validation** test might seem like more of an art than a science, and although wisdom and experience might seem like the only tools for validation design, remember that gathering requirements can be revealing and useful.

http://www.ni.com/solarverification/

Mechanical system consists of 984 hexagonal mirrors
- 6 sensors per mirror
  - 24-bit precision needed for data acquisition
- 5904 sensors total
- 3 actuators per mirror
  - 2 axes of motion per actuator (one coarse, one fine-grained)
  - 5904 axes of control total
Sampling of sensors occurs at 1kHz loop rate (1ms sampling interval)
Sampling of sensors must be synchronized to 1-10 uS precision

http://sine.ni.com/cs/app/doc/p/id/cs-13414#
Introduction: National Instruments

Academic  Advanced Research  Automotive  Big Physics  Consumer Electronics
Defense/Aerospace  Energy  Life Sciences  Mobile Devices  Semiconductors

Modeling  Graphical Programming  Real-Time Testing
Test Automation  Analysis and Reporting  Enterprise Connectivity

Multisim  LabWindows  Measurement Studio

PARTY SOFTWARE

COMPS AND HARDWARE
The Need for V&V

... from an industry perspective

V&V primarily affects businesses governed by ISO or FDA procedures or good practices that manufacture products such as pharmaceuticals or medical devices, or products for automotive and aeronautical use. Since such products are highly critical to health and safety, these industries are subject to formal oversight, including well-defined V&V processes.

Some companies voluntarily invest in formal V&V processes to reduce costs, or for competitive reasons.

The governing principles of V&V are well-defined for many industries, and are outlined by disciplines like Good Manufacturing Practices (GMP) or by regulation such as ISO9000, FDA's 21 CFR, or IEEE Standards. Each V&V system is similar but uses slightly different terminology to explain the generic requirements of the two processes. Specific requirements are usually not defined.
V&V Best Practices

... from an industry perspective

No written procedures exist to explain what must be verified or validated, or to define how testing must be accomplished. The same is true for reverification or revalidation if changes are made to a test system.

The organization must appoint someone to make recommendations about test procedures and review and approve them.

Although each company must decide and define how to implement design controls and change management in their products and test systems, this document provides some ideas and best practices to help with defining such policies.

NI TestStand

- Test management software for developing, deploying and maintaining test systems
- Graphical sequence editor environment
- Automate tests written in different languages, including LabVIEW, C++, C#, and Microsoft Visual Basic
- Report generation and database integration
NI TestStand Cont.
Sequential or multithreaded execution
Note: for a driver to be used in a multithreaded application, it must be thread-safe and reentrant.

Autoscheduling requires tests to be capable of executing in any order and be independent of prior test results.
From Requirements to Testing
Requirements

• Technical and procedural requirements that guide the product through each engineering phase

• Show trace from original project requirements to executed tests and test results

• However, requirements are still captured in natural language
FROM TIMING REQUIREMENTS TO A TIMING TESTBED
Cyber-Physical Systems

Multiple computers, comprising of sensors and actuators, connected on a network that act and react on events to meet timing constraints.

**Cyber**: software, hardware, networks

- **Sensor**: Sense events in the environment at specified rates or when events occur
- **Plant**: Physical Processes with timing characteristics
- **Actuator**: Actuate at the right time to optimally control the plant/physics

**Timing Requirements** specify when the cyber needs to interact with the physical:
- Latency
- Simultaneity
- Chronological
- Frequency
- Phase
- Sporadic
- Burst
Challenges in Programming with Time

- Time representation
- Precision
- Phase alignment
- Jitter
- Hardware clock
- Distributed systems
- Clock edge, clock domain, clock rate
- Multiple timescales, relation to global/TAI time
- Clock synchronization
- Execution time, WCET
- Response time, WCRT
- Communication time
- Timing tolerances

All these concerns make programming with time difficult. We need the right abstractions.
**Time** in the Software Lifecycle

**Requirements definition**
- Specify **timing** requirements, capture them in natural language/spreadsheets
- e.g. It should take exactly 100ms between sensing x and actuating y, with an acceptable tolerance of 2ms

**Design**
- Model the system with **timing** requirements in mind

**Implementation**
- Implement the system with **timing** requirements in mind

**Testing**
- Does the implementation satisfy the **timing** requirements?
Traditional Development

Design: Functional model

Implementation: Software implemented on specific hardware, tweaked and tuned to achieve correct timing behavior

Platform independent, no timing information

Platform dependent, timing depends on hardware: execution time, communication time, scheduling overhead, network latency, jitter

Brittle Designs
Instead ...

Design:
Functional model with timing specifications

Implementation:
Model implemented on specific hardware

Platform independent functional and timing application requirements

A correct implementation must satisfy both, the functional and the timing specifications
Enabling a New Paradigm

Correct-by-Construction Design

- Model system requirements in an abstract, mathematical model
- **Analyze** the model for correctness
- Verified tool chain to **generate** the implementation (automatically)

Global notion of time

- At design time, assume a global notion of time
- Abstract away details of imperfect clocks
- Made possible by modern clock synchronization techniques
Capturing Timing Requirements

Traditionally: Natural Language
- In form of text documents or spreadsheets
- Ambiguous, cannot be interpreted by computer

Formal, mathematical unambiguous description
- **Temporal logic** to formally specify patterns that timed behaviors of systems should (not) satisfy
- LTL, CTL, TCTL, MTL, TILCO-X, STL, ...
- Signal Temporal Logic (STL)\(^1\): properties related to the order of discrete events and the temporal distance between them

\(\varphi := G_{[2,6]} (|x[t]| < 2)\)

\(^1\)Alexandre Donzé, On Signal Temporal Logic, UC Berkeley, Lecture EECS294-98 Spring, 2014
Timing in the Model

- Traditional functional models: no timing specifications
- Synchronous languages: zero execution time abstraction
- Giotto: execution time bounds and IO timing on tasks
- Ptides: bounds on causal paths


Dataflow with Timing

Timing Specifications on IO nodes in Synchronous Dataflow (SDF)\(^1\)

Synchronous Dataflow (SDF):
- nodes consume and produce fixed amount of tokens, communicate via FIFO channels, can have initial tokens/delays on channels

\(^1\)Patricia Derler, Kaushik Ravindran, and Rhishikesh Limaye, Specification of Precise Timing in Dataflow Models, Memocode 2016
SDF with Timing

- **IO node**: the exact time of the interaction with the physics is called side effect. Side effects need timing specifications.
- **non IO nodes**: do not have side effects, do not need timing specifications.

---

Patricia Derler, Kaushik Ravindran, and Rhishikesh Limaye, Specification of Precise Timing in Dataflow Models, Memocode 2016
SDF with Timing

periodically read inputs

consume inputs from incoming FIFO and produce tokens on outgoing FIFO fast enough such that A always has a new value to actuate

periodically write outputs

1Patricia Derler, Kaushik Ravindran, and Rhishikesh Limaye, Specification of Precise Timing in Dataflow Models, Memocode 2016
Testing Timing Behavior

Application Requirements: functional, timing, ...

Application Model: algorithms, timing specifications

Platform Model: characterization of platform timing specifications

Design and Implementation

Test framework: Does the implementation satisfy the application requirements?

Platform: CPS Nodes (consisting of computing elements, sensors, actuators, network accessors), network components

Aviral Shrivastava, Patricia Derler, Ya-Shian Li Baboud, Kevin Stanton, Mohammad Khayatian, Hugo A. Andrade, Marc Weiss, John Eidson, Sundeep Chandhoke, Time in Cyber-Physical Systems, CODES-ISSS '16, Invited Paper, Pittsburgh, USA, October 2016.
Timing Testbed

$\varphi := G_{[2,6]} \ (|x(t)| < 2)$

Hugo A. Andrade, Patricia Derler, John C. Eidson, Ya-Shian Li-Baboud, Aviral Shrivastava, Kevin Stanton and Marc Weiss. Towards a Reconfigurable Distributed Testbed to Enable Advanced Research and Development of Timing and Synchronization in Cyber-Physical Systems, 2015 International Conference on ReConFigurable Computing and FPGAs, December 7-9, 2015.
Experiments

Two motors are controlled by two Arduino Mega 2560 boards that are synchronized, and the phase constraint is tested by two distributed NI-cRIO (9067 and 9035). The testing accuracy is checked by an oscilloscope.
Challenges

• Capturing requirements in an **unambiguous** manner, ideally a mathematical model

• Define the path from requirements to modeling, verification, validation and testing
  • Generating code from requirements
  • Generating tests from requirements

• V&V for test software - LabVIEW code

• Convincing industry of the importance of **formal** V&V