Systems Integration For Complex Cyber-Physical Systems

Dr. Daniel P. Schrage
Professor and Director
Center of Excellence in Rotorcraft Technology (CERT)
Center for Aerospace Systems Engineering (CASE)
Integrated Product Life-cycle Engineering Laboratory (IPLE)
Georgia Institute of Technology
Presentation Outline

• Brief overview of Modern Systems Engineering and the Risk Management Approach for Product Lifecycle Engineering Implementation
• Brief Description of Systems Integration for Cyber-Physical Systems
• Enabling Systems Integration Technologies for Cyber-Physical Systems
• Horizontal Integration Technology Example
• Complexity and Adaptability Metrics
• Product Lifecycle Management (PLM) Technology Example for Cyber-Physical Systems
• Summary and Conclusions
Product Life-cycle Engineering (PLE) and Product Life-cycle Management (PLM)

Integrated Product/Process Development

- Strategic Planning
- Market Analysis
- Implementation Plans
- Deployment Strategy

Risk Management as Enabler for Modern Systems Engineering

- Stakeholder Interface
- System Safety Management
- Risk Management
- Product Baseline Management

Led by Systems Engineering (SE)

Significant SE role

Deployment of The Product

PRODUCT LIFE-CYCLE ENGINEERING

- System Concept
- System Req.
- Sys Req. Valid
- Operations/ Maint Concept
- System Architecture
- System Safety

- System Integration
- Subsys Reqt Alloc
- Production Planning

- System Verification and validation
- Lean - Six Sigma Process Improvement

- Identification of upgrades
- Capture lessons learned
- MRO

PRODUCT LIFE-CYCLE MANAGEMENT

- Strategic Design
- System Design
- Detailed Design
- Development/Fabrication
- Operations & Life Cycle Support

Dr. D.P. Schrage
Georgia Tech
Modern Systems Engineering

- **Systems engineering** is an **interdisciplinary** field of **engineering**. It focuses on the development and organization of complex artificial **systems**.

- Systems engineering is defined by **INCOSE** as "a branch of engineering whose responsibility is creating and executing an interdisciplinary process to ensure that customer and stakeholder's needs are satisfied in a high quality, trustworthy, cost efficient and schedule compliant manner throughout a system's entire life cycle, from development to operation to disposal."

- This paper will discuss how **System Integration** plays an increasing **important role** in the development and airworthiness qualification of **complex cyber-physical systems**.
The Risk Management Approach has become the Preferred Approach

Risk Identification

What Can Go Wrong?
- Proposed changes
  - Staffing
  - Process
  - Design
  - Supplier
- Transition to production checklists
- Test failures
- Expectation Shortfalls
- Failure To Perform
- Negative trends
- Issues list
- ...And more

Risk Analysis

How Big Is the Risk?
- Likelihood
- Possible consequences
- Categories
  - Cost
  - Schedule
  - Technical
- Identify the risk level from the 5x5 risk grid

Risk Mitigation Planning

How Can You Reduce the Risk?
- Avoid by eliminating the risk cause and/or consequence
- Control the cause or consequence
- Transfer the risk
- Assume the risk level and continue on current plan
- ...And more

Mitigation Plan Implementation

How Are Things Going?
- Communicate risks to all affected parties
- Monitor risk plans
- Review regular status updates

How Can the Mitigation Plan Be Implemented?
- Determine what planning, budget, and requirements changes are needed
- Review with management and the customer
- Incorporate the changes
- Implement the plan

Simple Process Steps Are Common to Different Organizational Risk Management Practices

Dr. D.P. Schrage
Georgia Tech
Modern Systems Engineering Vee Diagram

Technical Management Processes

Initial Design Iteration
Fixed Design Iteration
Verified Design Iteration

Requirements Development
Logical Analysis
Decision Analysis
Technical Planning
Design Solution
Technical Assessment
Requirements Management
System Integration and Risk Assessment
Implementation
Transition From System Development To System Validation
Integration
Technical Processes Validation
Relationship of System Integration with Product/System Life Cycle

- For a Software Product **System Integration** follows the Coding phase in the development life cycle.

- While it may sound like the final assembly of the parts of a system, successful system integration involves almost every aspect of the project and reaches from the very beginning into and through the maintenance phase of a system’s life cycle.
What is System Integration?
Some Definitions

• **System Integration** is the bringing together of the components and ensuring that the subsystems function together as a system. **System Integration** is also about value-adding to the system, since additional capabilities are possible because of interactions between subsystems (Wikipedia)

• **System Integration** is the process of assembling hardware and software sub-systems to create a complete system (Georgia Tech Short Course on *Fundamentals of Systems Engineering*, 2007)

• **System integration** is the successful putting together of the various components, assemblies, and subsystems of a system and having them work together to perform what the system was intended to do (Chapter 14, System Integration, Condensed GSAM Handbook, Feb. 2003)

"Like a jigsaw puzzle: you have to make the pieces fit without getting out the scissors." –

Dr. Karl Maurer – On translating Greek sentences
Complete System Integration

- Most systems consist of both hardware and software. These two are sometimes looked at as complete systems in and of themselves, but they cannot function independently of each other. While they may be called the hardware and software systems, in the system level view they should both be considered as elements of the real, complete system.

- Two other system elements illustrated are people and support systems. For a system to be successfully implemented and used, these other elements must be in place and functioning correctly.
Methods of Cyber Integration

• Vertical Integration:
  – integrating subsystems according to their functionality by creating functional entities also referred to as silos

• Star or Spaghetti Integration:
  – each system is interconnected to each of the remaining systems

• Horizontal Integration of Enterprise Service Bus:
  – a specialized subsystem (BUS) is added to the system, dedicated to communicate with other subsystems
Vertical Integration

• The benefit of this method is that the integration is performed fast and with involving only the necessary vendors, therefore, cheaper in the short term.

• However, cost of ownership can be substantially higher than seen in other methods, since in case of new or enhanced functionality, the possible way to implement (scale the system) would be by implementing another silo.

• Reusing subsystems to create another functionality is not possible.
Star Integration or Spaghetti Integration

• When observed from the perspective of the subsystem which is being integrated, this reminds some of a star, but when the overall diagram of the system is presented, the connections look like spaghetti.
• The cost of this method can vary from the interfaces which subsystems are exporting.
• In a case in which the subsystems are exporting vendor-specific interfaces, the integration cost can substantially rise.
• Time and costs needed to integrate the systems is exponentially rising by adding additional subsystems.
• From the perspective of implementing new features, this method is preferable since it provides extreme flexibility to reuse the functionalities from existing subsystems into new systems.
Horizontal Integration or Enterprise service bus

• This allows cutting the number of connections (interfaces) to only one per subsystem which will connect directly to the BUS.
• The BUS is capable to translate the interface to another interface. This allows cutting the costs of integration and provides extreme flexibility.
• With systems integrated with this method, it is possible to completely replace one subsystem with another subsystem which provides similar functionally but exports different interfaces, all this being completely transparent for the rest of the subsystems.
• The only required thing is to implement the new interface between the BUS and the new subsystem.
Example of Horizontal Integration for Real-time Embedded Systems

• An Example of Horizontal Integration, will be provided based on Georgia Tech’s System Integration Role for the Rotary Wing Final Experiments in the DARPA SEC Program.

• The Enterprise service bus in this example case is called an Open Control Platform (OCP), which is a real-time embedded Common Object Request Based Architecture (CORBA) based open system middleware.

• An illustration of Fault Tolerant Control will be provided on how the Collective Control was replaced with Rotor RPM Control to provide similar functionally, e.g., rotor lift, as well as fault tolerant control.
Software Enabled Control
for Intelligent Uninhabited Air Vehicles (UAVs)

Contract Number: # F33615-98-C-1341
Award End Date: 4Q-FY04

Principal Investigators:
Daniel Schrage (AE)          George Vachtsevanos (ECE)
School of Aerospace Engineering School of Electrical and Comp Eng
Georgia Tech               Georgia Tech
Atlanta, GA 30332          Atlanta, GA 30332
daniel.schrage@ae.gatech.edu george.vachtsevanos@ece.gatech.edu
(404) 894-6257             (404) 894-6252

Co-PIs and Key Personnel:
Bonnie Heck (ECE), Eric Johnson (AE), J.V.R. Prasad (AE), Linda Wills (ECE)
controls.ae.gatech.edu/projects/sec
Key Participants in GT SEC Program

Dr. Dan Schrage
Cc-PI

Dr. George Vachtsevanos
Co-PI

Dr. D.P. Schrage
Georgia Tech

Dr. Eric Johnson
UAVRF Director
DARPA
Software-Enabled Control (SEC) Program

John Bay
DARPA/IXO

November 6, 2002
**SEC Objectives**

**Control Systems for Vehicle and Mission Management:**
- Design control systems for innovative vehicles
  - UAVs, OAVs, rotorcraft, fighters
- Increase automation for extreme maneuvers
  - Assured stability for flight mode transition
- Improve disturbance rejection and fault tolerance
  - Automatic control reconfiguration
  - Redundancy management
- Provide reusable middleware for coordinated, embedded software control on multiple aircraft types
  - Modernize flight control with adaptive, distributed computing

**Multiple levels of control:**
- **Vehicle** management (including flight-critical systems)
- **Mission** management (including route following)

**SEC provides innovative interoperable flight controllers for UAVs and manned vehicles, both fixed- and rotary-winged.**
SEC Technology Organization

Active State Models
- Honeywell
- U Washington
- Vanderbilt
- OGI
- U Minnesota

Multi-Modal Control
- Northrop Grumman
- Draper Labs
- Vanderbilt
- Cal Tech
- U Minnesota

On-Line Control Customization
- UC Berkeley
- Northrop Grumman
- Stanford
- Ga Tech

High-Confidence Systems
- Boeing
- SRI
- Honeywell
- MIT
- Cal Tech
- Rockwell
- Northrop Grumman

COTS Computing Technology

Real-Time Control Services

Bold Stroke

OPEN CONTROL PLATFORM

Prediction
Switching
Adaptation
Confidence

IMPLEMENTATION

New Control Technologies
- API for switching svcs.
- Predictive models oper.
- Hybrid multi-model svcs
- Integrated model svcs.
- Mode triggering defs.
- CLF and LPV control
- Hybrid stability, single sys.
- Customization svcs.
- Hybrid run-time svcs
- High-level multi-mode API
- Multi-mode run-time svcs.
- Multi-vehicle hyb. control
- Hybrid model checking
- Formal specification lang.
- Integrated fault mgt.
- Sensor/act reconfig.

Dr. D.P. Schrage
Georgia Tech
SEC Final Demonstrations

• Rotary Wing Demo: Low-level flight control – Led by Georgia Tech as **System Integrator**
  – Launch small vehicle from helicopter
  – Identify entry point in building (marked window)
  – Image specified object inside building
  – Rapid ingress and egress
  – Adaptive control and real-time hybrid mode switching capabilities

**Measures of Performance:**
- Stability of mode switch
- Robustness to unmodeled dynamics
- Aggressiveness of maneuvers
- Loop rate
- Trajectory tracking in standard maneuvers
- ADS-33 flying qualities

• Fixed-Wing Demo: Mission-level control – Lead by Boeing Phantom Works as **System Integrator**
  – One F-15E1 and one “real” UCAV (T-33, UCAV surrogate)
  – Additional virtual UCAVs
  – Real-time plan updates to UAVs from WSO in F-15
    - Trajectories to T-33 and virtual UCAVs computed with open control platform (OCP)

**Measures of Performance:**
- Formation stability
- Resource utilization/adaptation
- Quality of “anytime” tasks
- Rate of trajectory generation
- Detection of simulated failures
History of Georgia Tech UAS Program

• Began in early 1990’s with establishment of the *International Aerial Robotics Competition*, held for the first five years at Georgia Tech, then at Disney World and elsewhere

• Georgia Tech *won the Competition in 1993, 2001, 2002, 2003, 2008*, demonstrating autonomous helicopter flight for the first time and accomplishing Level 1 through 3 in final competition

• Army Autonomous Scout Rotorcraft Testbed (ASRT) Program *initiated UAV Research Program* at Georgia Tech from 1994-1997

• Flight Controls Research using VTOL UAVs in the Army/NASA Center of Excellence in Rotorcraft Technology (CERT) from 1997 to Present

• DARPA/AFRL SEC Program in *Software Enabled Control for Intelligent UAVs* 1998 - 2004

• *UAV Controls Research led to the establishment of the UAVRF in 2000*

• AFOSR MURI in Vision Based Control of Agile UAVs *expanded research effort* in 2002 – 2008

• Expansion to Multi-UAS Autonomous Control: 2008 - Present
Georgia Tech UAV Research Facility

http://controls.ae.gatech.edu/labs/uavrf
http://controls.ae.gatech.edu/labs/gtar

Dr. Eric Johnson, GT UAV RF Director
Georgia Tech UAVRF has a Fleet of Different Types of UASs

- **GT Buzz Aeromechanics Testbed**: 180 lbs, 10 ft rotor
- **GTMax**: 160 pounds, 10 ft rotor
- **1/3 scale Yak trainer**: 9 ft span
- **1/3 scale Edge 540T**: 9 ft span
- **GT Spy**: 5 pounds, 11 inch duct
- **D6**: Electric airplane
GTMax SEC Research Testbed

- Yamaha R-Max,
  - 66kg
  - 3m Rotor
  - 40kg max payload
- Flights Began March 2002
- Instrumented as a Research VTOL UAV
- Platform for DARPA Software Enabled Control program final experiments
Georgia Tech GTMax Systems Integration Approach

- Flight control reconfiguration
- Limit detection & avoidance
- Adaptive trajectory-following flight control
  - Envelope expanded: –50 to +85 ft/sec speed
  - Automatic Takeoff and landing
  - Automatic Aggressive maneuvers

controls.ae.gatech.edu/uavrf
Baseline Control/Navigation System Flight Testing

- Envelope expanded: –50 to +80 ft/sec speed
- Have experienced up to 40 knot gusts (estimated)
- First automatic takeoff and landing
- First automatic aggressive maneuvers
Small Autopilot Development

- 2 or 3 board design – processor & sensor boards
- 60mm x 90mm x 32mm, 120 grams
- Processor board
  - DSP is optimized for fast floating point operations (1.3Gflops)
  - FPGA enables fast, parallel and flexible IO interface
- Sensor board
  - 3 MEMS technology rate gyros
  - 4 MEMS +/-10g accelerometers
  - Air data
- GPS
- Installed on All GT UASs where applicable
SEC Rotary Wing Final Experiments

- **Unmanned supply/sustainment**
  - Mode transitioning
  - Fault tolerance
  - Rapid ingress and egress

- **Urban Reconnaissance**
  - Vision aided inertial navigation
  - Moving target tracking
  - Agile maneuvering
  - Trajectory generation
  - Fault tolerance
McKenna MOU Site
Final experiments Flight Demo

**Scenarios:**

7 Collaborators, ≥2 Flights

**Flight 1 (Reconnaissance):**
- Identify Structure & Portals (GT)
- Cooperative Reconnaissance (GT)
- Surveillance of a Moving Target in Urban Terrain (GT)
- Extreme Maneuvers - Envelope Protection (GT)
- Fault Tolerance - Envelope Re-Shaping (GT)
- Trajectory Generation (GT)

**Flight 2 (Unmanned Supply Sustainment):**
- Trajectory Generation (Draper, SSCI, GT)
- Mode Transitioning (GT)
- Fault Tolerance / Low Level Control (SSCI, UV, OGI, GT)
- Extreme Maneuvers (Draper, GT)

“The GTMax in a lab and not in the field is criminal.”

-Mike Barnes

Chief, Robotics Div. U.S. Army
Dismounted Battlespace Battle Lab.
Flight 1 (Reconnaissance):

- Identify Structure & Portals (GT)
- Cooperative Reconnaissance (GT)
- Surveil of a Moving Target in Urban Terrain (GT)
- Extreme Maneuvers - Envelope Protection (GT)
- Fault Tolerance - Envelope Re-Shaping (GT)
- Trajectory Generation (GT)
Flight 2 (Unmanned Supply Sustainment):
• Trajectory Generation (GT, Draper, SSCI)
• Mode Transitioning (GT)
• Fault Tolerance / Low Level Control (GT, SSCI, UV, OGI)
• Extreme Maneuvers (GT, Draper)
GT System Integration Approach as Rotary Wing Integrator for DARPA SEC Final Experiments

• Developed jointly with Boeing, Honeywell and UC Berkley a specialized subsystem (BUS middleware) called **Open Control Platform (OCP)** for **Horizontal Integration** demonstration

• Used a GT developed **robust adaptive neural network flight controller** to support a variety of guidance, navigation & control (GNC) algorithms developed by our academic (both at GT & elsewhere) & industry partners

• Created a **GT Unified Simulation and Test (GUST)** environment so partners and GT in-housers could evaluate the new GNC algorithms, first through a **Software In The Loop (SITL)** simulation; followed by a **Hardware In The Loop (HITL)** simulation for safety and flight readiness evaluation; and finalized by **flight demonstration** for verification and validation

• Resulted in an **extremely capable time-saving system integration approach**, as opposed to traditional trial and error approaches
Evolution Path of the OCP

Dr. D.P. Schrage
Georgia Tech

Avionics BoldStroke
(Boeing)

Control domain requirements & communication patterns
High-performance services (e.g., replication)

α-OCP
Controls
(Georgia Tech)

On-line customization and reconfiguration of mid-level controls;
*Demonstrated 100 Hz update rates through OCP

β-OCP
Real-time, dynamic reconfiguration of UAV software
(Boeing, Georgia Tech, Honeywell, OGI, UCBerkeley)
*Used in SEC RW Final Experiments

CORBA
Middleware for distributed, component-based software

TAO
Real-time CORBA (from Washington University)

Design Environments
Simulation
Run-Time Support

β-OCP

α-OCP

Design Environments
Simulation
Run-Time Support

β-OCP
Real-time, dynamic reconfiguration of UAV software (Boeing, Georgia Tech, Honeywell, OGI, UCBerkeley)
*Used in SEC RW Final Experiments

Evolution Path of the OCP
DARPA SEC GTMax Software Architecture

Sensor Drivers -> Navigation Filter -> Flight Controller -> Actuator Driver

Sensor Raw Data -> Command Vector

Primary Flight Computer

OCP + Mid-Level Control Components

Second Computer

Navigation Data Trajectory Commands

Desktop Computer

Ground Control Station
Software-In-The-Loop (SITL)
(Image Processing Static Images Example)
Hardware-In-The-Loop (HITL)
(Pictured in the GT UAV Research Facility)
Hardware-In-The-Loop (HITL) (Image Processing Movie Playback Example)
Hardware-In-The-Loop (HITL)
(Image Processing Sim Example)
GTMax Auto Aggressive Maneuver

180 Degree Velocity Change in a congested environment

>60 Degrees Max Pitch Angle

Keep nose aligned with velocity (zero sideslip) throughout

Decelerate 2/3 G

Go ~high to avoid saturating collective

Start and Finish at 30 Knots

~60 ft
Flight Envelope Protection Demonstrated

Design the flight control system incorporating envelope protection

Modular Design of Flight Control System and Envelope Protection System

“...operational responsibility for tasks must migrate from the ground station to the air vehicles, the air vehicles gaining greater autonomy and authority,...”

UAV Roadmap, DoD, Dec. 2002
SEC Technology - Fault Tolerant Control

“In improving UAV reliability is the single most immediate and long reaching need to ensure their success.”

- OSD UAV Roadmap 2002-2027

Collective Pitch Saturation

Loss of Tail Rotor Effectiveness

Stuck Swashplate Actuator

Conventional Control Algorithms

CRASH!
The Solution:
Hierarchical Architecture For Fault Tolerant Control
Reconfiguration for FTC

Active Control:
- Longitudinal Cyclic
- Lateral Cyclic
- Collective Pitch
- Tail Rotor Pitch

Optical RPM sensor:

Feedback Linearization:
- Reference Model
- Inverted Model
- Adaptive Neural Network
- Nominal controller

Model error

Control inputs

UAV State

Dr. D.P. Schrage
Georgia Tech
Reconfiguration / Envelope Re-Shaping

Active Control:
- Longitudinal Cyclic
- Lateral Cyclic
- Collective Pitch
- Tail Rotor Pitch

Reconfigurable Flight Controller

Left swashplate actuator
Right swashplate actuator
Aft swashplate actuator

Baseline Controller → Envelope Re-Shaping → Re-map Controls → RPM Controller
FTC OCP Implementation

Low Level Flight Controller, Navigator

Primary Flight Computer

Ethernet UDP Comm

50 Hz

“Datalink” Component

Fault Detection Identification Component

Reconfigurable Flight Control Component

Secondary Flight Computer – OCP/QNX

UAVState

Fault flag

Re-config Control

50 Hz
GT Flight Demo of Horizontal Integration (BUS) & FTC through the Open Control Platform (OCP) (Reconfiguration to RPM Control with Stuck Collective Actuator)
SEC Participants in SEC RW Final Experiments

- **Other Partners:** Boeing, Draper, GST, Lockheed Martin, Honeywell, McKenna MOUT, SSCI, OGI, Virginia Tech, MIT, UCLA
- **Sponsors:** DARPA, AFRL, AFOSR, NSF, NASA, Lockheed Martin
- **Georgia Tech Participants:** Anthony Calise, Henrik Christophersen, Graham Drozeski, Luis Gutierrez, Jincheol Ha, Bonnie Heck, Jeong Hur, Eric Johnson, Suresh Kannan, Adrian Koller, Sumit Mishra, Alex Moodie, Wayne Pickell, Seung-Min Oh, J.V.R. Prasad, Alison Proctor, Nimrod Rooz, Daniel Schrage, Suraj Unnikrishnan, Allen Tannenbaum, Shannon Twigg, George Vachtsevanos, Yoko Watanabe, Linda Wills, Ilkay Yavrucuk, and many others…
Georgia Tech Multi-vehicle Autonomous Control
Proposed Approach for a Cost-Effective System Integration Cyber-Physical Testbed

- Baseline the development time for a traditional VTOL UAS Development (Use System Integration Checklist from Systems Engineering)
- Identify Complexity and Adaptability Metrics for Evaluation
- Identify and apply the Enabling Technologies for major cycle time reduction and optimization of the Complexity and Adaptability Metrics
- Verify and Validate on the GT Buzz Testbed over a several year program the cycle time and enhanced robustness achieved
System Integration Checklist
(GSAM Handbook, Chapter 11)

Before Starting
1. Have you implemented systems engineering as an integrated life cycle effort?
2. Do your test plans include and support integration efforts?
3. Does your development plan allocate adequate time and resources for system integration efforts, including rework time?
4. Are the interfaces between components, assemblies, subsystems, and systems defined in adequate detail?
5. Will hardware be available for testing software during integration?
6. Is there a contingency plan if the schedule slips if and the integration schedule is compressed?
7. Are all elements of the system included in the integration plan?
8. Is all documentation current and available for reference?
System Integration Checklist

During Integration

9. Is there an efficient rework cycle in place to fix problems found during integration testing?

10. Are “fixed” modules or components integrated and retested at all levels of integration up to the level where the problem was found?

11. Is the people element (operators, maintainers, logisticians, trainers, etc.) being prepared to work with the system when it is deployed?

12. Is the support systems element (logistics, maintenance, training, etc.) being prepared to support the new system when it is deployed?

13. Are you following an iterative, progressive integration process?

14. Are experienced integrators involved with the integration?
System Integration Checklist

During Integration (continued)

15. Are area/subject matter experts involved with the integration?

16. Is adequate time being allowed for integration, testing, rework, reintegration, and retesting?

17. Are all necessary resources being made available for integration?

18. Is adequate testing being performed on integrated units (assemblies, subsystems, elements, system) to ensure that there are no surprises during acceptance testing?

19. Are you updating documentation during rework?

20. Are integration and system test errors being traced back to requirements and design? And if so, are the requirements and design being updated?
Converting the “Vee” to an “I” Acquisition Model
(Alex Boydston & Dr. William Lewis, AMRDEC for AHS Symposium 15 Oct 2009)
Complexity and Adaptability Metrics

• Complexity Metric based on Ontonix Definition and Approach
• Adaptability Metric based on time to Reconfigure for Fault Tolerant Control in flight for Improved Reliability and Product Lifecycle Management (PLM) for Development Time Reduction
• Verify and Validate the Complexity and Adaptability Metrics on the GT Aeromechanics Testbed
Defining Complexity

- The Complexity & reliability of complex systems is not fully understood which raises the following issues:
  - How do we accurately assess operating risk, performance, reliability of complex systems based on limited testing and analysis?
  - How do we know when system design is good enough?
  - Latent defects occur in supposedly well-tested, mature systems
- Ontonix, Inc has developed a tool and demonstrated how Fuzzy Cognitive Maps (FCMs) are automatically generated from complex data sets to generate a “Knowledge Map.”
Defining and Understanding Complexity
(Complexity Management for Decision Making, Tutorial by Gene Allen, Ontonix, 2005)

- The Complexity metric developed by Ontonix is provided below.

\[
\text{Complexity} \times \text{Uncertainty} = \text{Fragility}
\]

\[
C_{\text{design}} \times (U_{\text{manufacturing}} + U_{\text{environment}}) = F_{\text{product}}
\]

- A highly sophisticated design will result in a fragile product if:

  The manufacturing process is of poor quality
  The environment is very “turbulent”

  Hence, a more robust product requires:
  A high-quality manufacturing process, or
  A less severe environment in which to function, or
  A less “ambitious” initial design
Critical Complexity and Robustness

(Complexity Management for Decision Making, Tutorial by Gene Allen, Ontonix, 2005)

Robustness is proportional to the Margin $C_{\text{critical}} - C$. This measure is known as topological robustness and quantifies the system’s ability to preserve its functionality.
Tradeoffs using Critical Complexity and Robustness Metric

(Complexity Management for Decision Making, Tutorial by Gene Allen, Ontonix, 2005)
Adaptability Metrics Technologies

• For autonomous VTOL UASs real time embedded systems use Horizontal Integration and Integrated Software and Hardware-in-the-loop for Verification and limited Flight Testing for Validation to obtain reconfigurability and robustness and reduced fragility

• For Development Cycle Time Reduction use Product Lifecycle Engineering (PLE) through Product Lifecycle Management (PLM)
Product Lifecycle Management (PLM)

- The PLM environment that aerospace and automotive companies are using to support their PLE activities is beginning to be based on an integrated set of PLM tools.
- While CAD, CAE, CAM and PDM tools are being taught individually, IDM Tradeoffs and verification and validation of results need to be taught using IPPD along with an integrated set of PLM Tools by Integrated Product Teams (IPTs).
PLM Evolution

Using the 3D Media for:
- Social Design
- Global Collaborative Innovation
- Intellectual Property for PLM (Modeling & Simulation)

1. Digital Mock-Up
2. Product Life Cycle Management
3. 3D
4. Collaborative Business Process
5. Realistic Simulation
6. Lifelike Experience

Progress of Virtual Knowledge

Dr. D.P. Schrage
Georgia Tech
Typical System Life Cycle Cost

- Life Cycle Cost Effectively Rendered Unchangeable for a Given Design
- Life Cycle Cost Actually Expended

Cumulative Percent of LCC:
- 100%
- 75%
- 50%
- 25%
- 0%

Stages:
- Con Exp
- PD & RR
- E&MD
- Production, Deployment, Operations and Support
EFFECT OF VIRTUAL MANUFACTURING

- High Change Rate
- High Rework
- High Regeneration of Technical Data
- Poor Communication
- High Level of Liens

- Efficiency of build 5 in build 1
  - First Pass Success
  - Reduction of cycle time
  - Eliminate cost of poor quality

Recurring Product Cost Trend

DIGITAL LEARNING CURVE
PLM Impact on Systems Engineering

Integrated SE Processes

Knowledge Capture and Management

Product Life-Cycle Modeling

Integrated Virtual and Real Design, Test, Production, and Operations

Manual SE Integration

Design & Manufacturing

2D

3D

3D Collaboration Tools

Knowledge Inside


Dr. D.P. Schrage
Georgia Tech

Integrated Virtual and Real Design, Test, Production, and Operations

Integrated SE Processes

Knowledge Capture and Management

Product Life-Cycle Modeling

Manual SE Integration

Design & Manufacturing

2D

3D

3D Collaboration Tools

Knowledge Inside


Dr. D.P. Schrage
Georgia Tech

Integrated Virtual and Real Design, Test, Production, and Operations

Integrated SE Processes

Knowledge Capture and Management

Product Life-Cycle Modeling

Manual SE Integration

Design & Manufacturing

2D

3D

3D Collaboration Tools

Knowledge Inside


Dr. D.P. Schrage
Georgia Tech
IPPD Trade Studies Approach for IDM through PLM
Product and Process Models Required for IPPD and IDM Trade Studies

MULTI LEVEL LCC MODEL
Process Decomposition

ENGINEERING MODELS
Product Decomposition

Top-Down Aircraft LCC Model
Cost model
Required inputs

Bottom-up wing cost estimate

Cost metrics

Performance metrics

Re-Design Decision?

Aircraft Synthesis (Sizing)
Cost requirements
Performance requirements

Finite Element Analysis
Wing planform geometry

Materials
Air loads

Component Cost Modeling
Learning curves

Labor rates

Process metrics

Product metrics

IDM

KBS Process Modeling
Structural concepts
Alternative processes

Labor hours
Material costs

Finite element analysis

Dr. D.P. Schrage
Georgia Tech
Example Product and Process Models for IPPD and IDM Trade Studies

Dr. D.P. Schrage
Georgia Tech
Professional Education Certificate Program for Boeing Integrated Design & Manufacturing (IDM) Trade Study

Dr. D.P. Schrage
Georgia Tech
Product and Process Model Integration

- OML/Subsystems/Structures Integration
- Virtual Product Data Management (ENOVIA)
- Cost Analysis (LCCA)
- Vehicle Assembly Processes (DELMIA)
- Manufacturing Processes (DELMIA)
- Design Evaluation (Cost/Producibility)

- Vehicle Sizing & Performance (FLOPS)
- OML Design (CATIA)
- Structural Layout (CATIA)
- CFD Analysis (GT-NASCARD, FLUENT)
- FEM Analysis (ABAQUS)
- Aero Corrections
- Mass Properties Corrections

- Derivative Aircraft Definition
- Design Update
- Part Models
- Loads
- Grids

- FLOPS: FLight OPtimization System
- LCCA: Life Cycle Cost Analysis

Dr. D.P. Schrage
Georgia Tech
Proposed Systems Integration Education and Research on a Autonomous VTOL UAS Testbed

- Improved Systems Integration Education and Research is required for reduction in cost and schedule, and enhanced robustness in complex cyber-physical systems.
- Few universities have the capability and understanding to address Systems Integration Education and Research.
- Georgia Tech successfully demonstrated its Systems Integration capabilities on one of the most successful cyber-physical technology systems demonstrated to date – the DARPA IXO SEC Program.
- A low cost University Systems Integration Testbed can help to educate the needed interdisciplinary engineers and to eliminate risk for future military and civilian complex cyber-physical systems.
Rationale for Use of Off-the-Shelf Systems for Proposed GT VTOL UAS Systems Integration Testbed

• Proven approach for real-time and autonomous systems horizontal integration demonstrated by GT on the SEC Program is very relevant and state-of-the art

• While the Yamaha RMax served as a more than adequate air vehicle testbed for the SEC Program it was not a fully engineered and documented vehicle system. Furthermore, purchase of new vehicles and replacement parts are greatly restricted and may not be possible

• The GT Rotor Buzz, based on the SWE/UAVRL remotely piloted vehicle, is a well engineered and fully documented system, to include complete set of CATIA V5 models and is being purchased as the GT Rotor Buzz air vehicle
GT Rotor Buzz Actual and CAD MODEL

Engine Area

Picture of Actual Model

CAD Drawing
GT Rotor Buzz Actual and CAD Rotor Model

Rotor Head Area

Picture of Actual Model

CAD Drawing
System Integration Approach for a Cyber-Physical System VTOL UAS Testbed

[Diagram showing integration approach with various components and processes]

Dr. D.P. Schrage
Georgia Tech
Summary and Conclusions

• **System Integration** is becoming more relevant as complex systems and system of systems become more cyber-physical systems providing more capabilities, but at a higher risk

• **System Integration** capability is and will become even more critical for successful demonstration of military and commercial aerospace systems and also as a discriminator for gaining world market share

• The presentation has attempted to provide the fundamentals on **System Integration**, as well as to demonstrate on a real program that world class system integration can be found in many places, including universities

• **Education and Research** in Systems Integration of Complex Cyber-Physical Systems is required and a relevant testbed must be available for interdisciplinary teams