IGT System Engineering

Peter Kazanzides

October 19, 2006

My Background

1989-1990 Postdoctoral research at IBM on ROBODOC

1990-2002 Co-Founder of Integrated Surgical Systems

- Commercial development of ROBODOC® System
- Commercial sales in Europe (CE Mark)
- Clinical trials in U.S. and Japan
- FDA approval for ORTHODOC planning system
- ISO 9001 certification

2002-present Research faculty at JHU CISST ERC

Outline

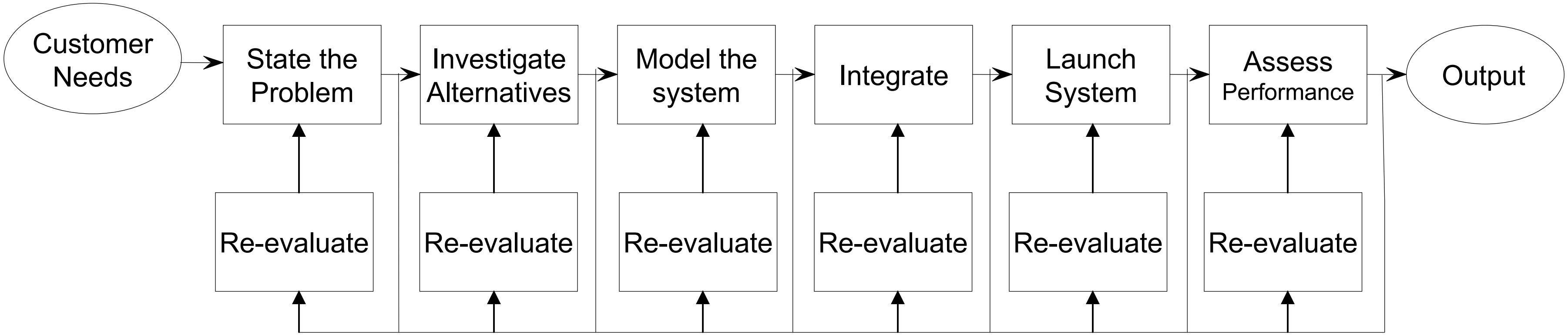
- Elements of System Engineering:
  - Requirements
  - Risk Analysis
  - Architecture
  - Modeling / Simulation
  - Verification and Validation
- Case study: Image-guided robot for rodent research
- Current work: Surgical Assistant Workstation
- Summary and Conclusions
- Three challenges and opportunities for assistance

What is System Engineering?

- Spans the entire development process
- Considers the entire system, including hardware and software (interdisciplinary)
- Key activities include:
  - Requirements
  - Risk analysis
  - System architecture
  - Modeling / simulation
  - Verification and Validation

Systems Engineering: The SIMILAR Process

INCOSE: International Council on Systems Engineering



The Systems Engineering Process from A. T. Bahill and B. Gissing, Re-evaluating systems engineering concepts using systems thinking, *IEEE Transaction on Systems, Man and Cybernetics, Part C: Applications and Reviews*, **28** (4), 516-527, 1998.

How much System Engineering is enough?

- Depends on development scenario:
  1. Prototype for (non-clinical) feasibility study
  2. System for clinical use under IRB (and possibly IDE)
  3. System for clinical use and eventual commercialization
  4. Approved medical device

University/Industry Collaborations

- The typical partnership:
  - University does research, builds prototype system
  - Industry acquires IP, extracts requirements from prototype, re-designs (almost) everything
- Questions:
  - Can system engineering approach improve this?
  - Will regulatory agencies allow this?
  - Can Universities remain agile?

Requirements

- There are always requirements!
- Informal (undocumented) is fine for early prototypes
- Documented requirements necessary for:
  - Any system for clinical use
  - Any development involving multiple/distributed parties (e.g., university researchers, industry, clinicians)
- Requirements may not be necessary for toolkits

Risk Analysis

- Risk analysis should be performed by cross-functional team, including application expert
- Two common methods:
  - Fault Tree Analysis (FTA)
    - Top down analysis: trace each system failure down to components
    - Most useful for after-the-fact analysis
  - Failure Mode Effects (and Criticality) Analysis (FMEA/FMECA)
    - Bottom up analysis: for each component failure, determine (potential) system failure
    - Most useful in design phase (proactive)

Risk Analysis: FMEA / FMECA

- Typically presented in tabular format:
  - Failure Mode
  - Effect on System
  - Cause of Failure
  - Methods of Control
  - (FMECA) Risk Priority Number (RPN), product of:
    - Severity (S) – seriousness of effect of failure
    - Occurrence (O) – likelihood of failure
    - Detection (D) – ability to detect failure

Risk Analysis: FMECA

- Risk assessment should be an iterative process:
  - Determine RPN for initial system design
  - Add methods of control where necessary
  - Determine RPN for system design including methods of control
- FMECA table can include both risk assessments or just the final one

Architecture

- Most often refers to software, but can include hardware elements
- Simple definition: "How the pieces work together"
- From Software Engineering Institute (SEI) at CMU:

  Software architecture is the set of design decisions which, if made incorrectly, may cause your project to be cancelled – Eoin Wood

Architecture: 4+1 View

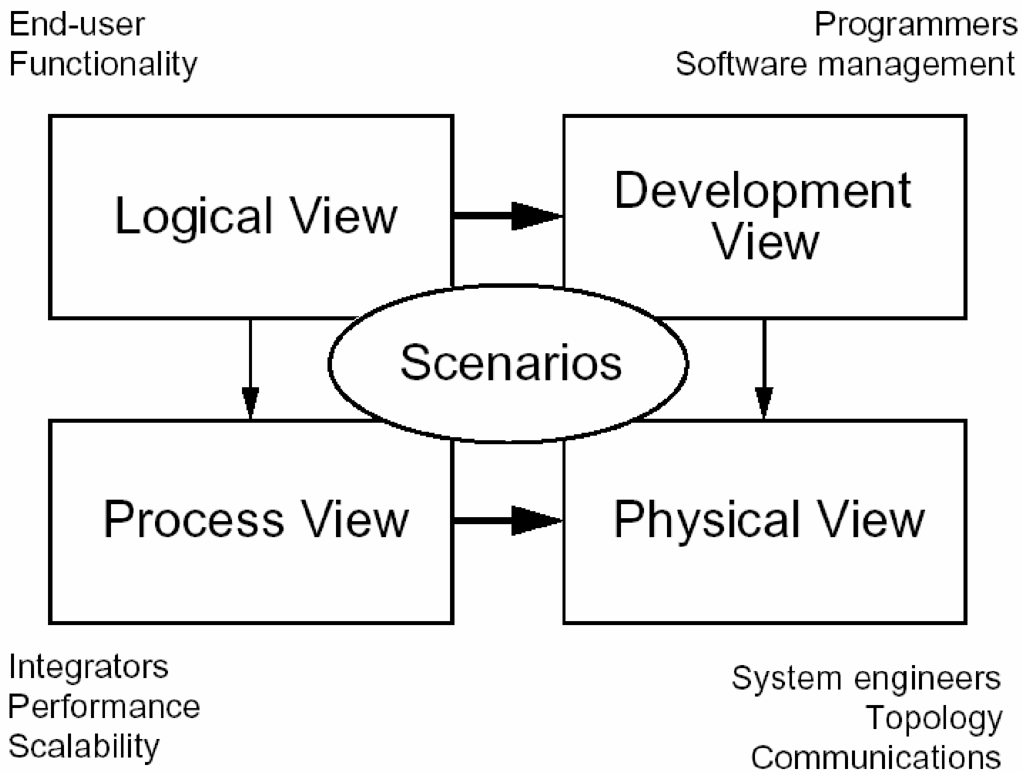
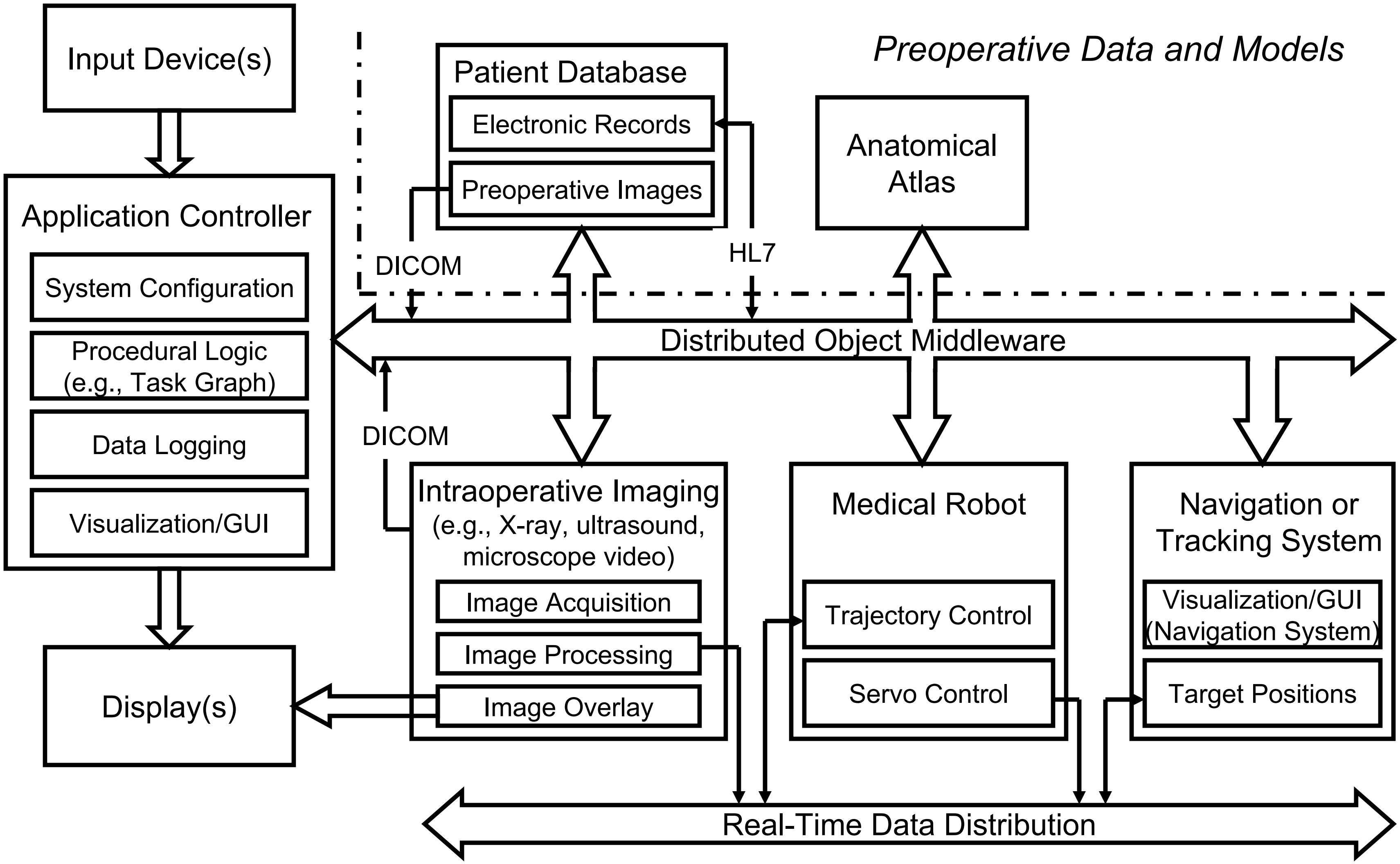


Figure 1 — The “4+1” view model

From: P. Kruchten, “The 4+1 View Model of Architecture”, *IEEE Software*, 12(6), Nov 1995.

Other approaches include RM-ODP, Zachman framework

Image-Guided Intervention System Architecture



Modeling and Simulation

- Create models of the system to guide the development
- Model Driven Architecture (MDA) ®*
  - Model is enduring asset
  - Perform simulation/testing with model
  - Generate code from model
  - Is the technology (tool set) there yet?

"Processes, Methodologies, and Tools used for the Development of a Model Driven Architecture Based Open Software Framework for Distributed Medical Devices", Amen Ra Mashariki, Ph.D. proposal, Morgan State University

*Registered trademark of OMG

Verification and Validation (V&V)

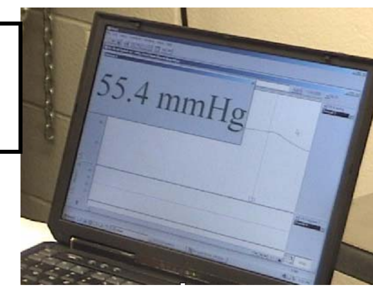
- NASA SATC: "*differences between verification and validation are unimportant except to the theorist*"
- Two primary V&V activities:
  - Reviews (inspections, walkthroughs)
  - Testing (e.g., against requirements)
- Many software toolkits (e.g., VTK, ITK) can be automatically tested, but this is more challenging for IGT systems

Case Study:
Image-Guided Robot for Rodent Research

Initial application: correlate pO_2 measurements with PET values to validate non-invasive method for locating hypoxic tumor regions

PET pixel value

pO_2 probe reading



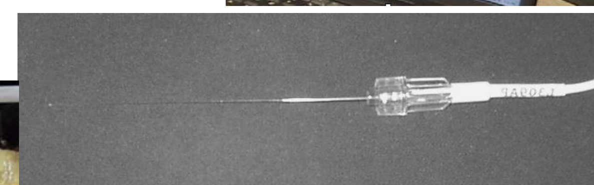


Image-Guided Robot for Rodent Research: Requirements

- Distributed team:
  - Developers at JHU (Baltimore)
  - Customers (users) at Memorial Sloan Kettering Cancer Center (New York City)
- Requirements were critical:
  - First meeting: Sept 2003
  - Three major revisions
  - Final version approved: March 2004
  - System installed at MSKCC: Jan 2005

Image-Guided Robot for Rodent Research: Risk Analysis

- System not for human clinical use
- No formal risk analysis performed
- Requirements document included safety requirements
  - Emergency stop button

Image-Guided Robot for Rodent Research Architecture Views

- Physical View – see following
- Development View – see following
- Process View – not needed
- Logical View – not used
- Scenarios (Use Cases)
  - Discussed during development of requirements specification
  - Should have documented this!

Image-Guided Robot for Rodent Research
Physical Architecture

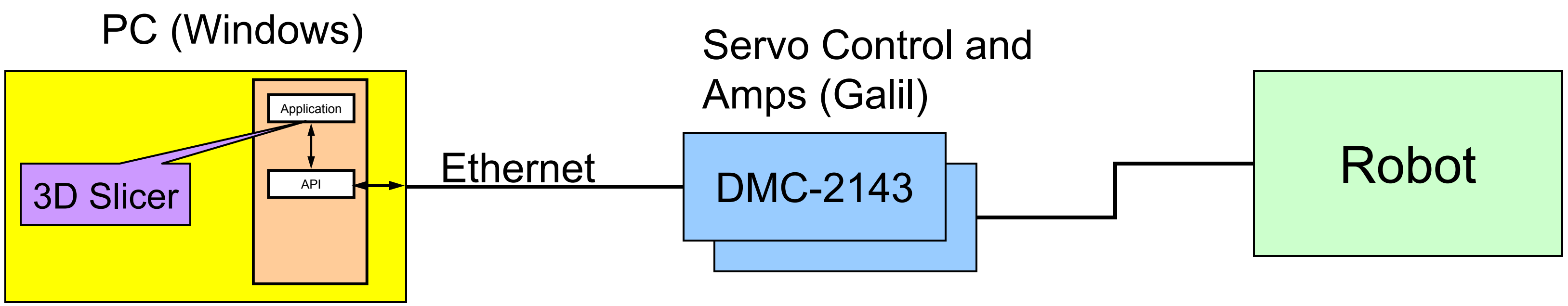
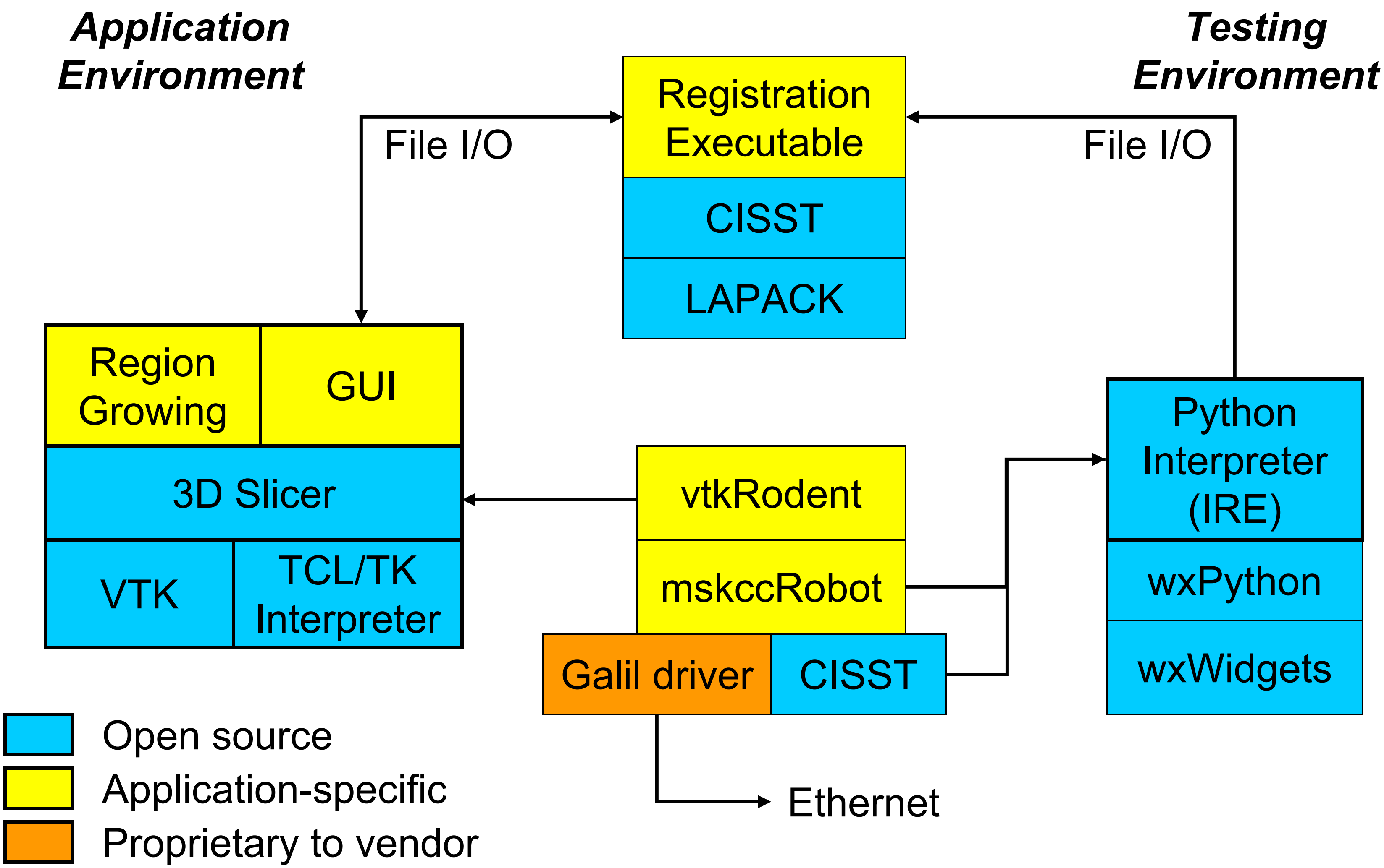


Image-Guided Robot for Rodent Research Development Architecture



CISST Software Package

Foundation libraries

- cisstCommon
- cisstVector
- cisstNumerical
- cisstInteractive

↓

Real Time Support

- cisstOSAbstraction
- cisstDeviceInterface
- cisstRealtime

↓

Interventional Devices

- cisstTracker
- cisstRobot
- …

Open Source
www.cisst.org/cisst

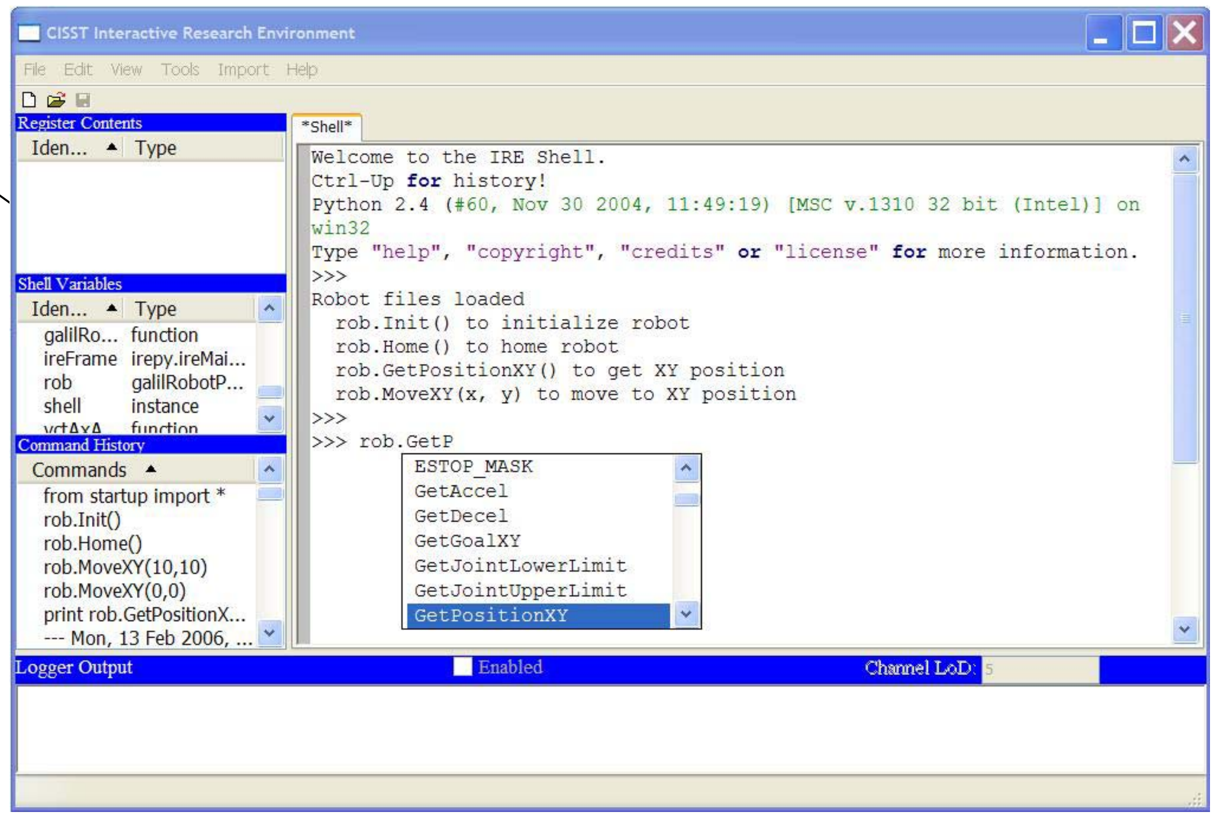


Image-Guided Robot for Rodent Research: Modeling / Simulation

- Not needed

Verification & Validation

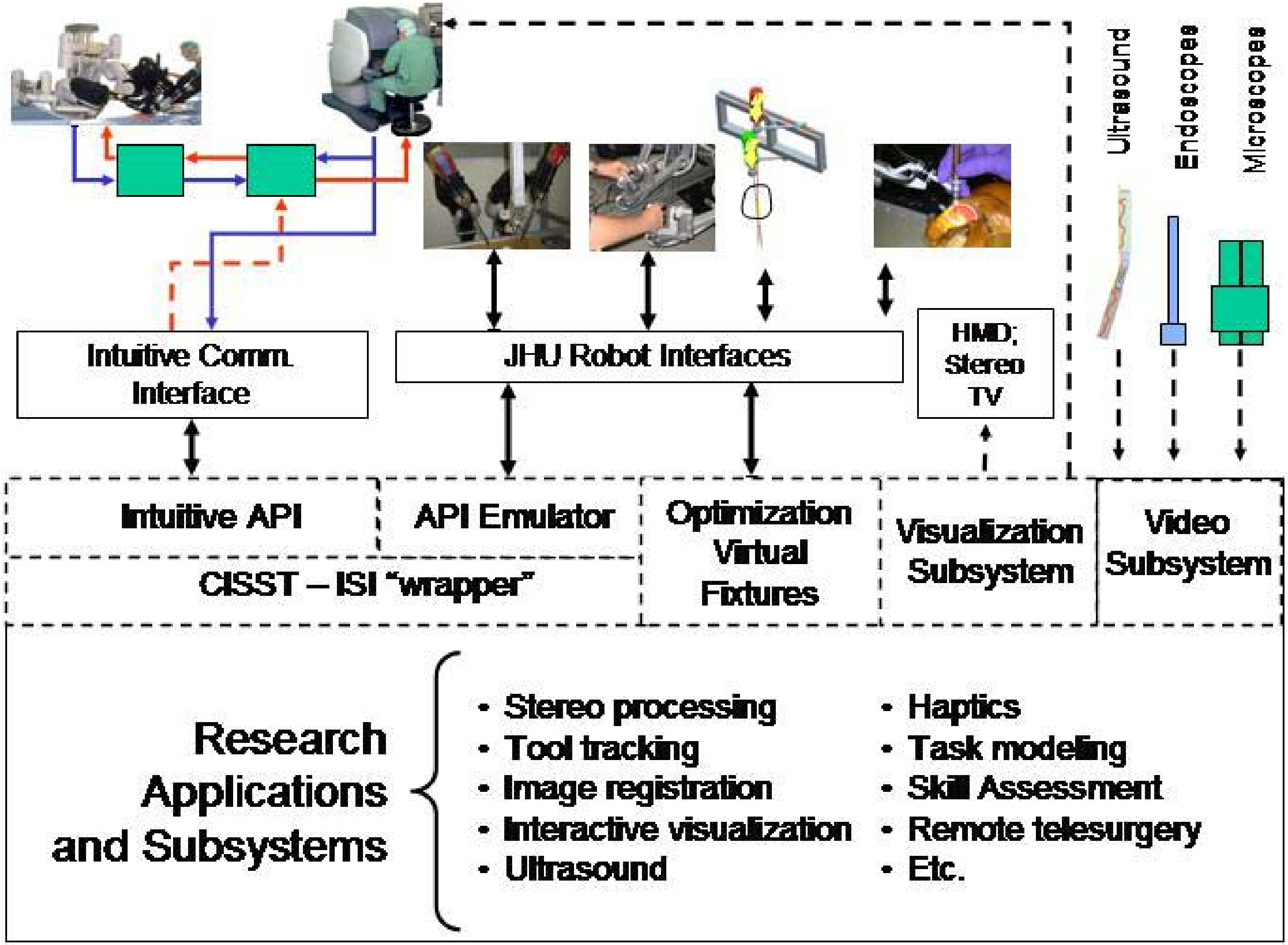
- Customer acceptance testing
- System accuracy tests with phantom:
  - PET Fiducial Localization Error: 0.26 mm
  - Robot Fiducial Localization Error: 0.18 mm
  - Target Registration Error: 0.29 mm

"Design and Validation of an Image-Guided Robot for Small Animal Research," *MICCAI*, Copenhagen, Denmark, Oct 2006.

Current Work: Surgical Assistant Workstation

- NSF supplement to CISST ERC
  - Started Sept 2006
- Collaborative effort between JHU and Intuitive Surgical
  - Integrate daVinci research API with CISST Software
  - Emulate daVinci API for research robots
- Currently working on system architecture
  - Plan to use 4+1 views

Surgical Assistant Workstation



Summary and Conclusions

- System engineering integrates multiple disciplines over the development life cycle
- Key activities include requirements, modeling, architecture, verification & validation
  - What’s needed depends on development scenario
- Image-guided robot for rodent research required good system engineering documentation due to separation of developers and users
- Surgical assistant workstation will require even better system engineering due to distributed development and increased complexity.

Three Challenges

1. Matching level of system engineering to development scenario
   – Too much: time & schedule cost
   – Too little: poor outcome or difficulty proceeding to next stage → time & schedule cost
2. Insufficient tools for modeling, simulation, and automated testing of IGT systems
3. Building an interdisciplinary team and integrating its output

How can NCIGT help?

1. Provide guidance documents for performing system engineering
2. Develop realistic modeling and simulation environments for system evaluation and testing
3. Provide forum for researchers in different disciplines to collaborate on common problems