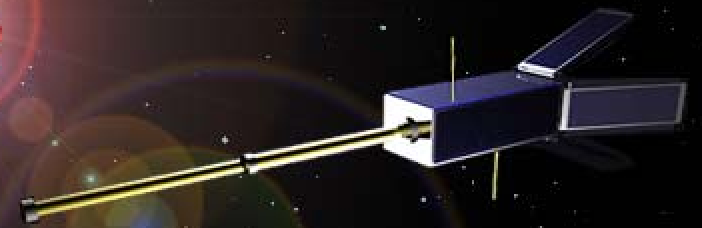
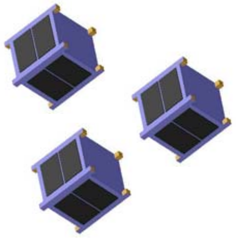
SPACE INTERNET WORKSHOP 4



Reusable, Open Source, and Internet Enabled Software Development Tools for Efficient Small Satellite Design, Mission Planning, and Analysis

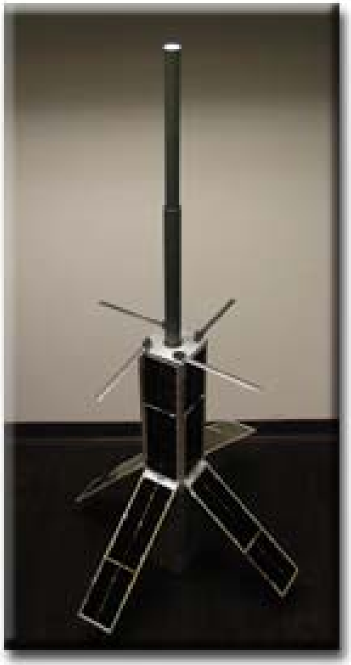
Eric Tapio   Stanford University

The Pursuit for Efficient S/C Design

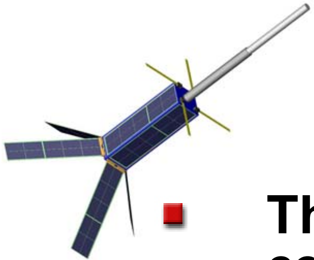


- The Stanford Small Sat Challenge:
  - Learn system engineering processes
  - Design, build, test, and fly a CubeSat project
  - Goal: Initial design to final completion in one year
- Accomplish goal by:
  - Use of COTS parts
  - Implement efficient design & development processes with currently available technologies
- Project Constraints:
  - Limited Resources & Small Teams
  - Low Cost
  - Small Physical Size: Standard CubeSat 10 x 10 x 10 cm, 1kg
- Experienced Risks Resulting in Project Failure:
  - **No.1 Risk:** Ineffective communication and interaction between designers and customer regarding requirements and payload specifications during design and early development phases

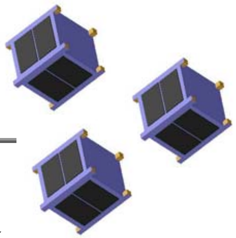
Success Story – The QuakeSat Project



- QuakeSat
  - Stanford's 3rd & lightest (9 lb.) small satellite launched
  - 1.5 years initial design to completion & launch
  - Anticipated 6 month mission
  - Launched June 30th 2003, and still operating…
- Design Challenges & Process Inefficiencies
  - Biggest inefficiencies were in Design phase
    - What's our baseline design, and our expected power?
    - Do we have power margin?
  - Biggest Challenges were related to lack of infrastructure to evaluate design
    - No significant leverage off previously developed analysis tools
    - Only time for one evolving design solution
    - Required significant parallel development of infrastructure
- **Bottom Line:** Better integrated analysis tools are needed upfront
- **Biggest Lesson:** A completed satellite helps, but much more is necessary to achieve mission readiness, and mission success



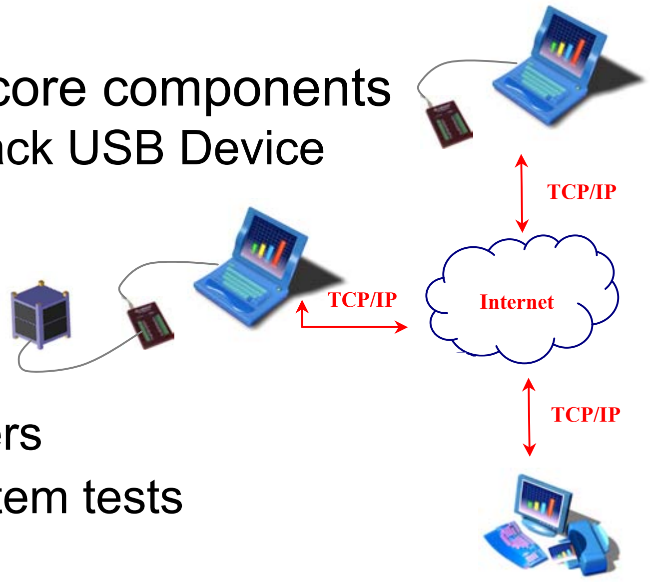
An Improved Paradigm



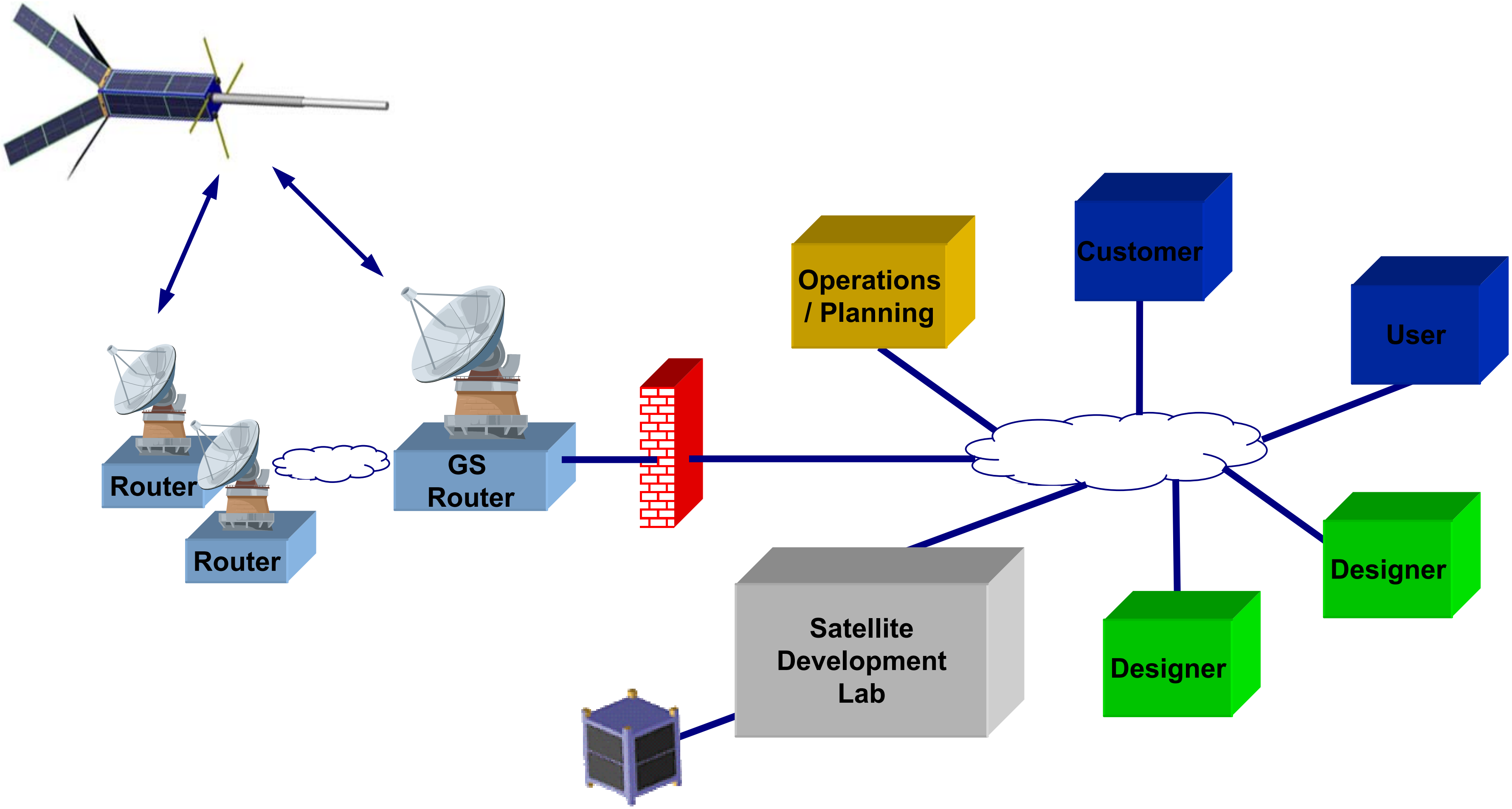
- **The Desired Goal:** Enable transparent end-to-end network communications among space mission resources
  - A necessity after spacecraft deployment
  - But also critical throughout the S/C design & development lifecycle
- Start with a mission-centric architecture starting from the design stage with networked TCP/IP solutions
  - Enable communication between design team, developers, customer, operators, mission planners
  - During 90% of the design and development phase, QuakeSat team members worked independently from separate geographical locations
- Target Lifecycle Reusable Tools and Incremental Development
- Build a satellite, co-develop mission essential tools
- Use IP enabled ground stations (Stanford Mercury GS, J. Cutler)
- Develop mission tools that work seamlessly with GS for mission data, information visualization, and data dissemination

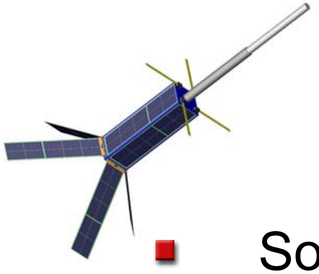
Better SW Tools - One Approach

- Utilize and Interface Software COTS programs
- Leverage off of existing COTS capabilities
  - STK Scenario and Orbit Propagation Tools
  - Matlab scientific computing and hardware interfacing capability
  - National Instruments Data Socket Technology
  - The Internet, web servers, HTML, XML
- Enable Spacecraft design with simple core components
  - Laptop/ PC, Internet Connection, LabJack USB Device
- Enables end users to:
  - Run scenario simulations
  - Perform mission utility analysis
  - Evaluate design performance parameters
  - Affect hardware for system and subsystem tests
  - Hardware interaction across the Web
  - Data & resource sharing

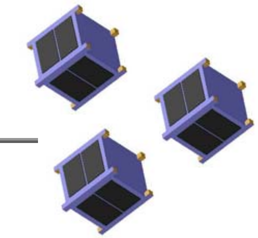


Team Architecture



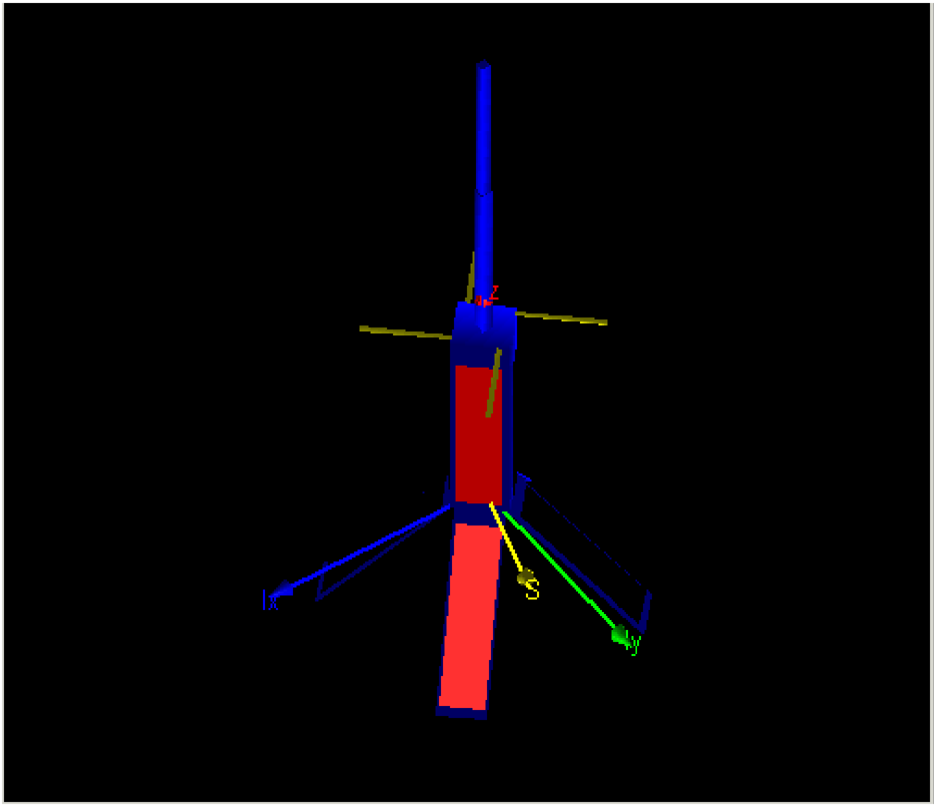


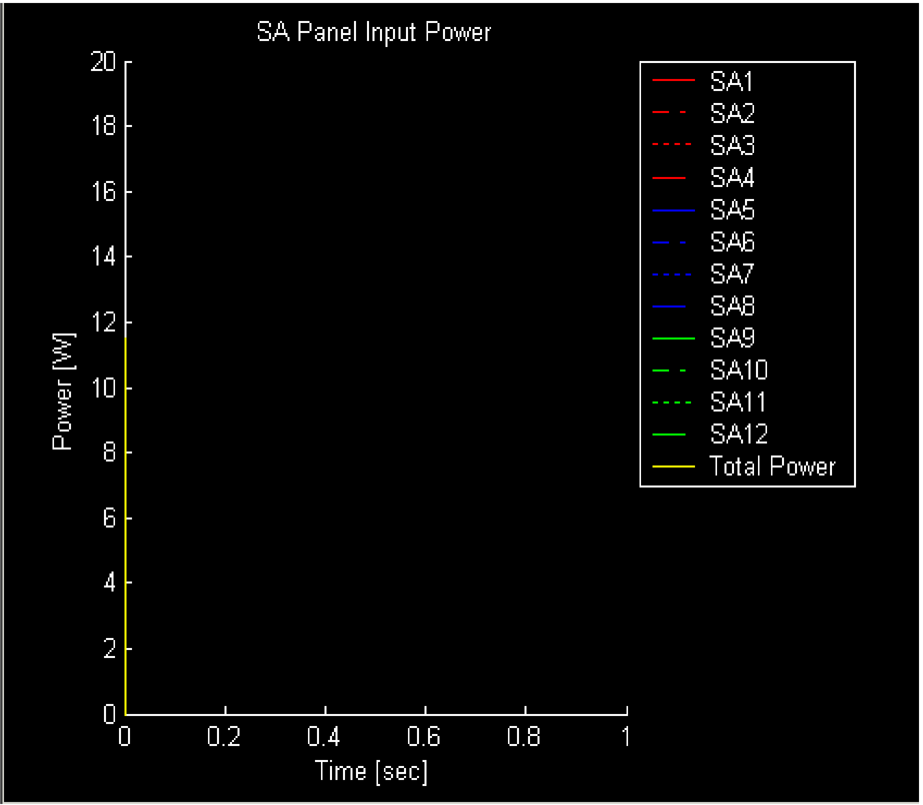
S/C Solar Power Performance



- Solar Power Performance Parameters tool
  - For small satellites in particular, power is a driving design constraint
  - Utilize reusable tool, run different design models in an STK scenario
  - Output model power analysis
  - Incorporate 3D-visualization, helpful for designs w/ complex geometry
  - Drive power subsystem hardware in baseline scenario simulation
  - Enable tool so that satellite design teams have access via Internet
- QuakeSat Power Performance Design & Testing
  - Took 1.5 months to modify an Matlab simulation with our design specifics, and it did not easily accommodate alternative designs
  - Testing power subsystem HW in early development phase, invaluable
  - **Change of requirements:** Near end of development, change in orbit, equatorial to sun-synchronous, and change in attitude profile. What's the expected impact on power?
  - Change scenario definition in STK, and rerun

S/C Solar Power Performance Demo

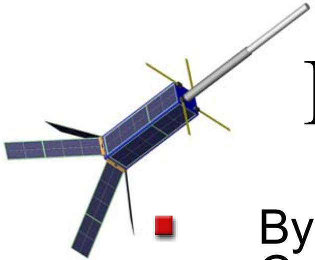




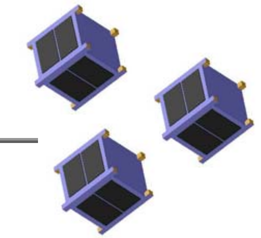
-All quaternion and vector data generated by STK
-QuakeSat model is given:
  - 'Spinning' constraint
  - 1.1°/sec about Z- Inertial

-Sun Vector, shown in Yellow

- Based on the panel-sun geometry, expected power for each solar panel is plotted
- The net input power is calculated
- Power conversion model outputs a voltage to drive HW



Findings, Conclusions & Trends



- By utilizing COTS parts and available technologies, small sats & CubeSats are a cost efficient platform for conducting short scientific missions in space
- With development of better integrated design tools, small satellite design and development can be a more efficient process
- With appreciable time savings in using reusable software design tools, goal of making small satellite design to flight time in one year a consistent and repeatable process is obtainable
- The development of reusable, open source, S/C design and development tools are crucial infrastructure needed, and provides a helpful starting point for new teams
- TCP/IP enabled design tools that enable end-to-end communication may be effective in mitigating No. 1 risk preventing project completion
- Small satellite projects mimic all the complexities that their larger counterpart projects face. So leverage off the low cost of implementing new ideas on small satellites, and scale up to improve current processes used for design on larger projects

For More Information:

http://ssdl.stanford.edu/quakesat