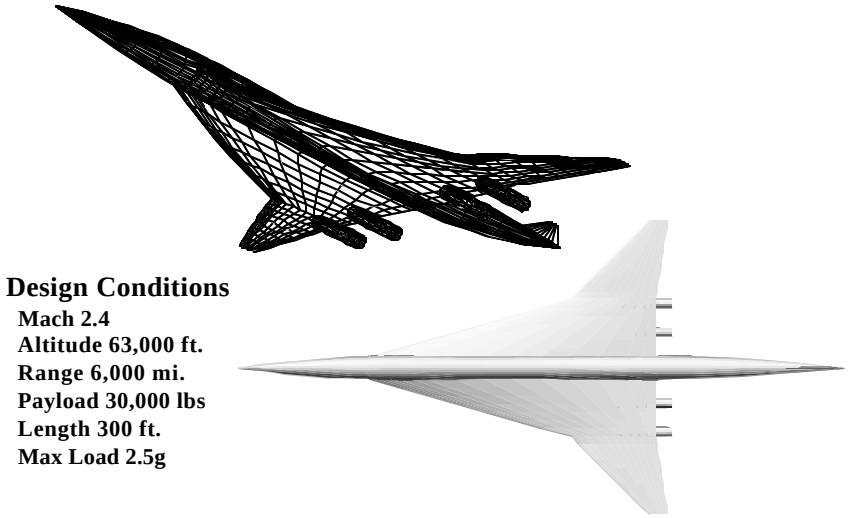
# CONFIGURATION MANAGEMENT OF AN OPTIMIZATION APPLICATION IN A RESEARCH ENVIRONMENT

James C. Townsend, Andrea O. Salas, M. Patricia Schuler

**NASA Langley Research Center (LaRC)**




Optimization in Industry II, Banff, Canada, June 6-11, 1999

# HSCT4.0 Complexity

| **Application** | **HSCT2.1** (1994) | **HSCT3.5** (1997) | **HSCT4.0** (1999) |
|---|---|---|---|
| **# Design Variables** | 5 | 7 | 271 |
| **Constraints** | 6 | 6 | 31868 |
| **Major Codes** | | | |
| Aerodynamics | WingDes | ISAAC | CFL3D, USSAERO |
| Structures | ELAPS | COMET | GENESIS |
| Performance | Range eq'n | Range eq'n | FLOPS |
| Propulsion | Engine deck | Engine deck | ENG10 |
| **Analysis Processes** (without looping) | 10 | 20 | 70 |
| **Analysis Control** | | | |
| Major Loops | Wgt Conv., Trim | Wght Conv., Aeroelastic, Trim | Wgt Conv., Aeroelastic, Trim |
| Load conditions | 2 | 2 | 7 |
| Mission conditions | 1 | 1 | 10 |
| Process (with loops) | O(10) | O(100) | O(1000) |
| Total time (serial) | O(minutes) | O(1 hour) | O(11 hours) |
| **Optimization Cycle** | | | |
| (ndv+1)*#anal. procs | O(100) | O(1000) | O(100,000) |
| Total time/cyle | O(10 min) | O(3 hours) | O(3 days) |

# Software Configuration Management (SCM)

- **Purpose**:
  - Increase the reliability and quality of software
  - Identify and control software during its development and use
  - Provide defined procedures and tools for
    - **entering new software into SCM system**
    - **checking out / checking in software for modifications**
    - **reviewing / prioritizing proposed modifications**
- **Activities**:
  - Establish baselines
    - **fixed versions for delivery or further development**
  - Review, control, and track changes, using
    - **version control tools**
    - **forms**
    - **reports**

# Software Configuration Management Tools
## Used by HSCT4.0

- Version control:
  - Consultants evaluated commercially available SCM tools
  - Chose TRUEchange™**,*** recommended because
    - **version control** associates sets of **related files**
    - works on **all required computers**
- Electronic forms:
  - **Forms** and associated **database** available at LaRC
    - **adapted** to the needs of the HSCT4.0 project
  - **Manage changes** to the software, keep change-control **metrics**
    - **trouble reports, change requests, promotion notifications**
- SCM Plan:
  - Documents procedures (updated as necessary)

*Use of trademarks or manufacturer names does not imply an official endorsement by NASA

# Expectations from SCM for HSCT4.0

- **Keep track** of the many pieces of the project:
  - Code
    - **avoid confusion among researchers' different versions**
  - Documentation
    - **current version immediately available to all**
  - Input, output, and intermediate data
    - **both test and research results**
- **Know** how and when versions were created
  - What **versions** of codes were used to **generate data**
  - What **versions** were **delivered**
  - What were **earlier** versions
    - **ensure ability to go back**

My answers to the five questions are in terms of applying SCM to projects like HSCT4.0 at LaRC

# Q1: *What is the State-of-Art reflected in the literature with respect to using SCM for optimization development (e.g. HSCT4.0) at LaRC?*

- Software configuration management is well established for typical applications
  - Software that has **clear requirements** and a well-defined life cycle
    - business applications
    - control systems
    - engineering applications (e.g., CAD system development)
    - software **supporting** research (e.g., data collection, spacecraft control, but **not** research itself)
  - Rarely used where software ***is*** the research
    - **HSCT4.0**

# Q2: *What are road-blocks / hurdles in implementing SCM for projects like HSCT4.0?*

- Biggest hurdle: Getting **team members** to **accept** SCM
  - Overcoming **inertia**
    - “It’s not the way we’ve always done it!”
    - “It’ll interfere with research creativity!”
    - “Do we really need this added complication?”
  - Understanding its **value** and potential **benefits**
  - Applying SCM **actively** and **consistently**
    - **not** just placing final products into an SCM repository
    - **willingly**, rather than by coercion
- Providing **Time** and **effort** required to plan and implement SCM
- **Training** team members in SCM
  - Not part of standard engineering curriculum
- Determining when MDO **research** codes are ready for SCM
  - Never will be generic "industrial-strength" codes

# Q3: *What research needs to be undertaken that will promote the use of SCM for projects like HSCT4.0?*

- **Pilot projects** applying SCM where the software development ***is*** the research
  - HSCT4.0 is the first one at LaRC
  - Now needed for ISO 9001 certification
- Methods and tools for efficient SCM of **formatted** documents
  - e.g., MS Word, Adobe FrameMaker
- **Improvements** to the SCM procedures
  - Provide **flexibility** to accommodate research environment
- Development of **automated** tools to **convert legacy codes** into objects
  - this is more OO development than SCM research
- Training of grad students and sending them into industry

# Q4: What are the elements of an SCM system necessary to support distributed design with High Speed Computing at its core (e.g., HSCT4.0)?

- Both SCM Plan and its implementers need to **be specific**:
  - Define the **procedures**
  - Define the **responsibilities**
  - Provide **templates** or **examples** of what is expected
- Important to allow **adequate time** in the schedule
  - To **introduce** SCM to the team
  - To **perform** SCM installation
- **Explicitly** address the use of SCM
  - In the project plan and schedule
  - In all subcontractor tasks

## Q5: What new developments in CM methodology are needed to accommodate the full potential of the new hardware/software architectures for High Speed Computing that offer massively parallel processing in a heterogeneous, distributed computing environment?

- Development of automated software change-control tools for a **heterogeneous**, **distributed** environment:
  - Different file systems (i.e., not NFS mounted)
  - Different directory structures
    - **not common paths to communications libraries, etc.**
  - Different vendors' communication products
    - **e.g., CORBA differences, Java RMI, Globus - LAWE**
- Development of secure, yet efficient, methodologies
  - Storage / retrieval
  - Communications

The following are the five original Optimization in Industry II questions as answered by members of the Multidisciplinary Optimization Branch at LaRC

# Q1: *What is the State-of-Art reflected in the optimization / information sciences literature with respect to using your work in industry?*

- References
  - **Giesing & Barthelemy: "A Summary of Industry MDO Applications and Needs"**. Presented at the 7th MA&O Symposium Sept 1998, AIAA-98-4737
    - a summary of MDO from the industry perspective
    - two focused sessions, ~12 papers from industry on MDO at their shops
  - **Sobieski & Haftka: "Multidisciplinary Aerospace Design Optimization: A Survey of Developments"**. Presented at the Aerospace Sciences Meeting, January 1996, AIAA-96-0711
- Multilevel methods and approximation management work published
  - however, both areas are **new to industry**
  - still working on **proofs of concepts**
- All-at-once MDO: either
  - **medium-fidelity** methods with **2-3 disciplines**, or
  - **low-fidelity** methods with **~10 disciplines**
- Multilevel MDO: only **low**-fidelity methods with **3** disciplines

# Q2: *What are road-blocks / hurdles in implementing your work in industry?*

- Overwhelming roadblock: **time** and **effort** required to assemble any practical, significant multidisciplinary application
- Approximations methods require only sufficient testing
  - will be easily implemented in industry as an easy-to-use toolbox
- Multilevel methods will be more difficult to implement in industry
  - assume the availability of problem information that may not be present
  - multilevel method really needs "grey" boxes, not just a set of black boxes
- MDO research produces prototype codes
  - not generic "industrial-strength" codes
- Current optimization tools require practice, skill, and math/CS background
  - e.g., even iSIGHT requires experience to understand what is happening and to formulate problem in the best way
- Others:
  - Lack of sensitivity analysis capability in commercial CAD systems
  - Lack of training
    - optimization problem not recognized by most engineers when they see it
  - Difficult to define meaningful, physically significant optimization problem
    - identify appropriate objective function, constraints, design variables

# Q3: *What research needs to be undertaken that will promote the use of your work in industry?*

- Develop **automated** tools to **convert legacy codes** into objects
  - this is more development than research
- Conceive and develop **problem-definition** tools
- Extensively **test** and **tune** algorithms already proposed on progressively realistic problems
- Undertake **joint agreements** with industry
  - no money changes hands; we learn about their specific application and they learn about tools (e.g., ADIFOR and CONMIN)
  - LaRC had a very successful one with Boeing Helicopters
- Train grad students and then send them into industry
  - LaRC had good success with this, too

# Q4: *What are the elements of a design theory necessary to support distributed design with High Speed Computing at its core?*

There are two elements:

- Fine-grained distributed computing
  - a matter for disciplinary specialists
- Coarse-grained distributed computing
  - involves problem decomposition along disciplinary lines
    - **decomposition is inherent in multidisciplinary optimization**
  - currently lacks robust discipline integration
  - applies to both gradient-based optimization and genetic algorithms
    - **requires high-speed processors with lots of local memory**

# Q5: *What new developments in analysis and optimization methods are needed to exploit full potential of the new hardware/software architectures for High Speed Computing that offer massively parallel processing in a heterogeneous, distributed computing environment?*

- Special purpose codes for a specific application can be quite successful
  - e.g., structural optimization
- **But**, optimization methods that work well for 100 design variables have big problems when the number of design variables grows to 1000s
  - LASSO is one example (CONMIN with Sequential Linear Programming)
- Among very little work in this area is current Vanderplaats SBIR
  - look at optimization methods for large constrained nonlinear problems
- Genuinely parallel optimization algorithms
- Methods for managing a suite of optimization algorithms
  - apply to a given problem for comparison of efficiency
- Software components written as modules with well-defined interfaces
  - computation, physics, and introspection services
  - problem solving environment (e.g., LAWE) on distributed computing service