



NORTHROP GRUMMAN

DEFINING THE FUTURE

March 1st, 2005

GSAW 2005 Emerging Directions

Northrop Grumman Next Generation Ground System Architecture Study (NG²GAS)

Next Generation Ground: Looking into the Future

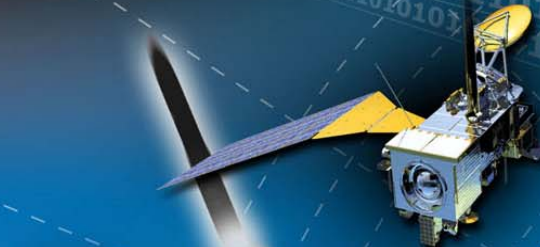
Emmet C. Eckman III

Sr. Engineer & Technology Evangelist

Surveillance & Reconnaissance Division

Data Systems Operations

FROM UNDERSEA TO OUTER SPACE TO CYBERSPACE



Agenda

- Problem Description
- NG/MS Strategy

Challenges, Decisions & Lessons Learned

- Architectural Development
- Transition Planning
- Cost Model
- Summary & Questions

Problem Description

- 2 contracts to independently study architecture approaches and options
 - CPFF contracts, 9 months (start 10/03), 4 formal TEMs
 - Unconstrained Requirements Document (URD)
 - 100s of reference documents
- Describe the next ground system architecture
 - next: start in 2010, thru 2020 maintenance thru 2030
 - ground system: reception of data from signal reception at the ground thru reporting
 - architecture: DoDAF Operational (OV) & System (SV) views
- Cooperative COTR
 - OK to go outside of the box
 - CAIV and traditional funding boundaries
 - Unconstrained Communication boundaries

Team Strategy

- **Six geographically distributed teammates (NJ, VA, CO, CA)**
 - Broad Mission Understanding
 - Technology Forecasting
- **Removed proprietary corporate boundaries**
- **“War room”**
 - Write it down and put it on the wall
 - Weekly team TELECONs
 - Microsoft Sharepoint Portal ®
- **Bi-weekly reporting**
 - Time
 - Products
- **Develop architectural alternatives**
 - One champion per architecture
 - Several based upon leveraging from current systems
 - One “Unconstrained” Architecture
 - Compare and contrast
 - Select (merge) the best that survived

Architectural Development

Crystal Ball

■ Challenge

- How to envision requirements, CONOP, and the *system*
- *Go back to the day you graduated college, and think would you have been able to predict today's technology?*



■ Decision

Focus on System "behavior"

■ Lesson Learned

- Easy to leverage from existing systems or systems being built or acquired (5 years)
- Harder to get beyond "today" (10+ years)
- Have lots of white board space
- "what" v. "how"

Architectural Development

Gary Larson -- "What you *say* and what Spot *hears*"

■ Challenge

Almost every technical term and acronym is overloaded

■ Decision

Write it down

■ Lesson Learned

➤ Don't assume the audience knows what you mean

➤ Standardization is a very slow process

So what really is:

Architecture

Service-Based (SBA)

System-of-System (SoS)

Architectural Development

“Everything I need to know I learned in kindergarten”

■ Challenge

- Engineers tend to tightly couple their self worth with their products
- Finding closure

■ Decision

- Unbiased arbitrator
- Cat herder

■ Lesson Learned

- Play well with others & Check your ego at the door
- Get gratification from other sources



Architectural Development

Cultural Differences

- **Challenge**

NG/MS DSO (f.k.a. TRW) have a culture of lively discussion & debate

- **Lesson Learned**

- *Recognize* the cultural differences
- *Appreciate* the cultural differences
- Lively debate does not always equal disrespect

Architectural Development

Niche Expertise

- **Challenge**

 - Integrate niche domain experience with mission experience

- **Decision**

 - Augment core team (with lots of mission experience) with niche expertise

- **Lesson Learned**

 - Niche expertise without domain knowledge was not as helpful as we predicted
 - Manage cultural differences

Architectural Development

Architectural Evaluation

- **Challenge**

 - How to compare architectures?

- **Decision**

 - SEI Quality Assessment Workshop (QAW) and ATAM (Architecture Tradeoff Analysis Method) process

 - http://www.sei.cmu.edu/ata/ata_method.html

- **Lesson Learned**

 - QAW and ATAM worked well, but required significant investment from customer and project engineers

 - Government participation

Architectural Development

A Few Important Architectural Tenants

- **Challenge**

 - What architectural tenants are more important?

- **Decision**

 - Keep them small

 - Try to balance

- **Lesson Learned**

 - Encapsulation, Abstraction and Data hiding

 - Separate system behavior from implementation

 - What we want the system to do

 - How the system will perform the desired activity

 - Design architecture that was benign to signal source

 - Traditional emphasis on signal source

 - Overabundance of ground resources

Architectural Development

As an interesting aside ...

Two teams, two approaches



Interestingly, both teams came up with remarkable similar high level architectures

- Differences in the starting approach (OV vs SV)
- Differences in nuances

the “undiscovered truth” remains undiscovered

Transition Planning

Transition Planning

■ Challenge

- Where to start, where to end
- Define the SoS boundaries
- Define needed infrastructure & assumptions
- Legacy data migration

■ Decision

Define the strategy for transition

- What to keep from existing systems
- New system

■ Lesson Learned

- Didn't have the strategy soon enough in the analysis process
- "Opted" out



Cost Model

What is the future going to cost?

- Challenge

Cost the architecture and transition

- Decision

- Excel Cost Model

- Started with legacy models

- IDEF activity model

- Input, output, processing & control
- "black box" approach

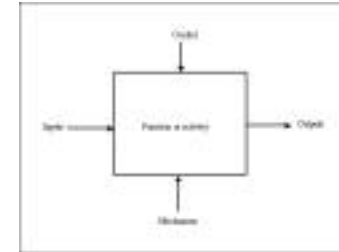
- Peel-the-onion



- Lesson Learned

Don't let an engineer (*unconstrained*) build a cost model

- 14 Worksheets
- 256+ variables to describe the input and environment
- 1200+ variables to describe the architecture

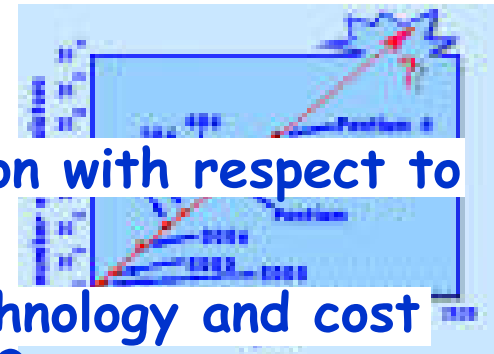


Cost Model

Moore's Law

■ Challenge

- Apply Moore's law and technology evolution with respect to cost and capability or performance
- Could you estimate today's computer technology and cost from the first two computers you bought?



■ Decision

No consensus on the application of technology evolution

■ Lesson Learned

Built the model to be able to apply technology forecasting on/off and different rates

IF HISTORY IS ANY GUIDE
MOORE'S LAW WILL
TRANSCEND CMOS
SILICON AND JUMP TO A
DIFFERENT SUBSTRATE. IT
HAS DONE SO FIVE TIMES
IN THE PAST.

-
- Thank you
 - Questions?