16.89J / ESD.352J Space Systems Engineering
Spring 2007

B-Terrestrial Observer Swarm (B-TOS)

16.89 Architecture Review

May 16, 2001

B-TOS Project Participants

- **Students**
  - Mirna Daouk
  - Nathan Diller
  - Qi Dong
  - Carole Joppin
  - Sandra Kassin-Deardorff
  - Scott Kimbrel
  - Daniel Kirk
  - Michelle McVey
  - Brian Peck
  - Adam Ross
  - Brandon Wood
- **Faculty and Staff**
  - Daniel Hastings
  - Hugh McManus
  - Joyce Warmkessel
  - Bill Kaliardos
- **Aggregate Customers**
  - Bill Borer (AFRL)
  - Kevin Ray (AFRL)

Overview

- What is 16.89 Space Systems Engineering?
  - Project-oriented class in which students work as a team to develop system architectures and preliminary design of a space system
- What did the class do?
  - Employed new design process to determine user needs, and explore large design space rather than focusing on point design
  - Applied process to B-TOS space system architecture study
- What is B-TOS?
  - Iteration B of Terrestrial Observer Swarm design
  - For mapping of the Earth's ionosphere using swarms of satellites
  - Sponsored by Air Force Research Laboratory

Presentation Outline

- Introduction (Chapter 2)
- Process Development (Chapter 3, 4, 5)
- Results (Chapter 6)
- B-TOS Requirements (Appendix C)
- Spacecraft Design (Appendix E)
- Review and Concluding Remarks

Accomplishments

- B-TOS mission characterized and defined
- Key attributes of swarm architectures determined
- Thousands of architectures traded
- Optimal architectures identified
- Sensitivities and design studies point to challenges, but basically validate design
- Requirements derived for a potential architecture

Completed process for architecture study

Selected and assessed a potential architecture

Section Outline

- Introduction
  - Motivation
  - Project scope
  - Objective
- Process Development
- Results
- B-TOS Requirements
- Spacecraft Design
- Review and Concluding Remarks

User Driven Motivation and Needs

- Ionosphere disturbs propagation of EM waves
- Need to model ionospheric variations for predictive purposes
- AFRL is primarily interested in two missions:
  1. Studying global behavior of ionosphere
  2. Characterize ionospheric turbulent regions (near equator)
- AFRL model inputs:
  – Vertical Total Electron Content (TEC)
  – Electron Density Profile (EDP)
  – Beacon Angle of Arrival (AOA)

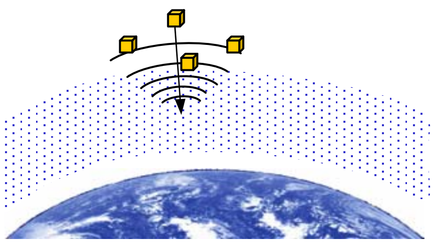
Pedagogy Driven Motivation

- **Measurable Outcomes**
  - ✓ Organize and plan development of integrated space system architecture to meet customers' needs
  - ✓ Establish functional and high level systems requirements
  - ✓ Interactively work in teams to conduct systems engineering trades across multiple system elements

"The process *is* as important as the final product"

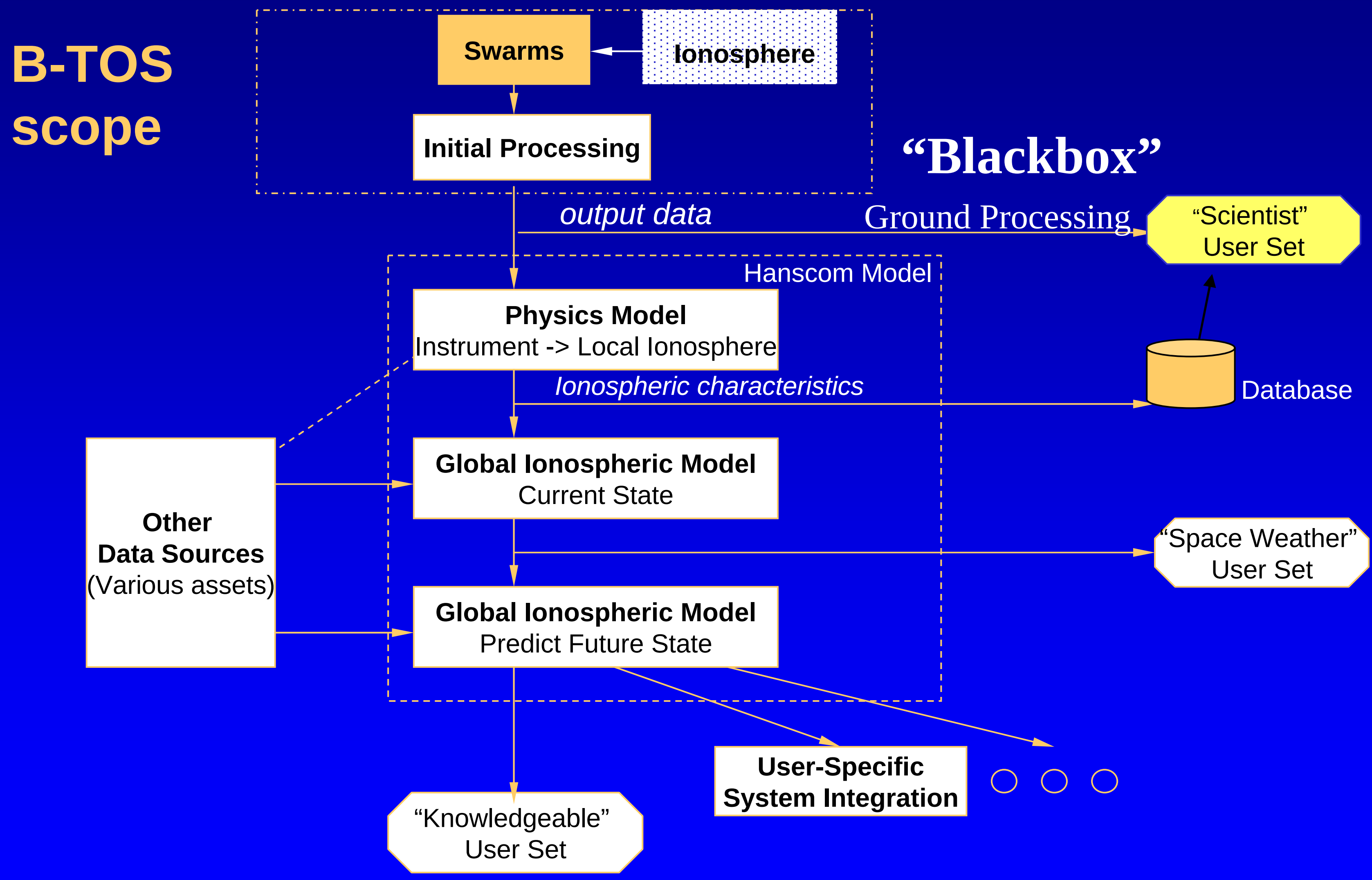
Constraints and Scope

- Space-based data collection platform
- Swarm configuration
- Use top-side sounder
  – Minimum altitude
  – Available instrument capabilities
- Frozen orbit (inclination = 63.4 degrees)
- Mission Life Fixed to 5 Years
- Use TDRSS for communication with the ground
- Bill Borer and Kevin Ray are aggregate customers representing all end users
- Scheduling constraints



Topside Sounding

B-TOS Information Network



Mission Statement

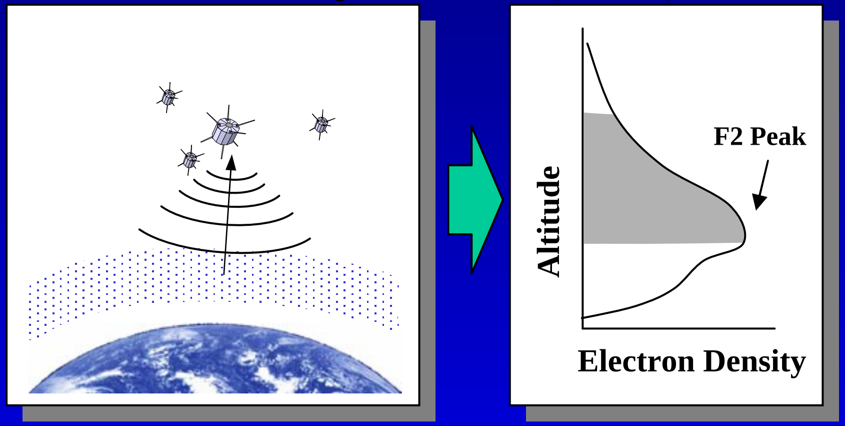
Design a conceptual swarm-based space system to characterize the ionosphere.

Building upon lessons learned from A-TOS, develop a deliverable, by May 16, 2001, with the prospect for further application.

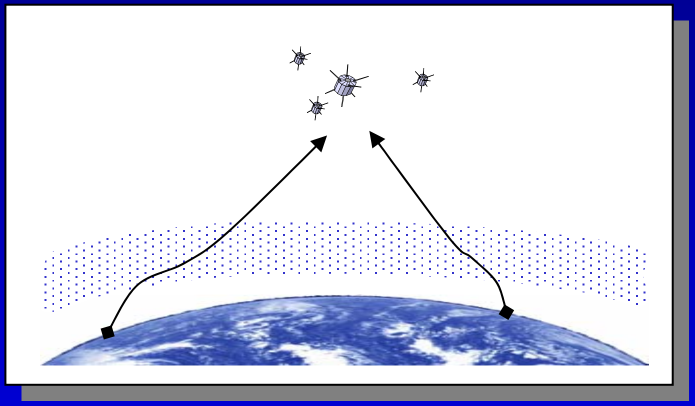
Learn about engineering design process and space systems.

Payload Mission Overview

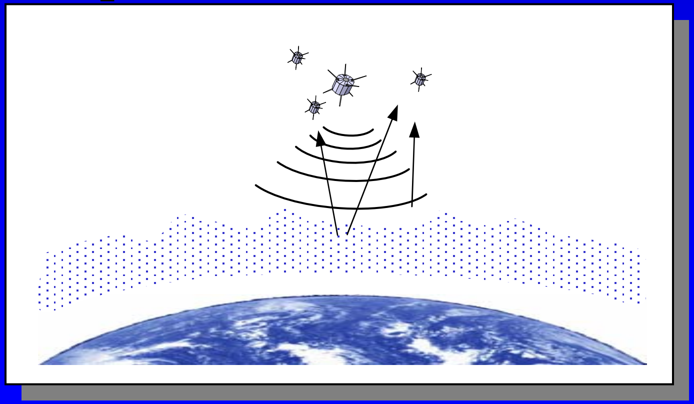
Electron Density Profile (EDP)



Beacon Angle of Arrival (AOA)



Ionosphere Turbulence



Payload "B"

Process Summary

Section Outline

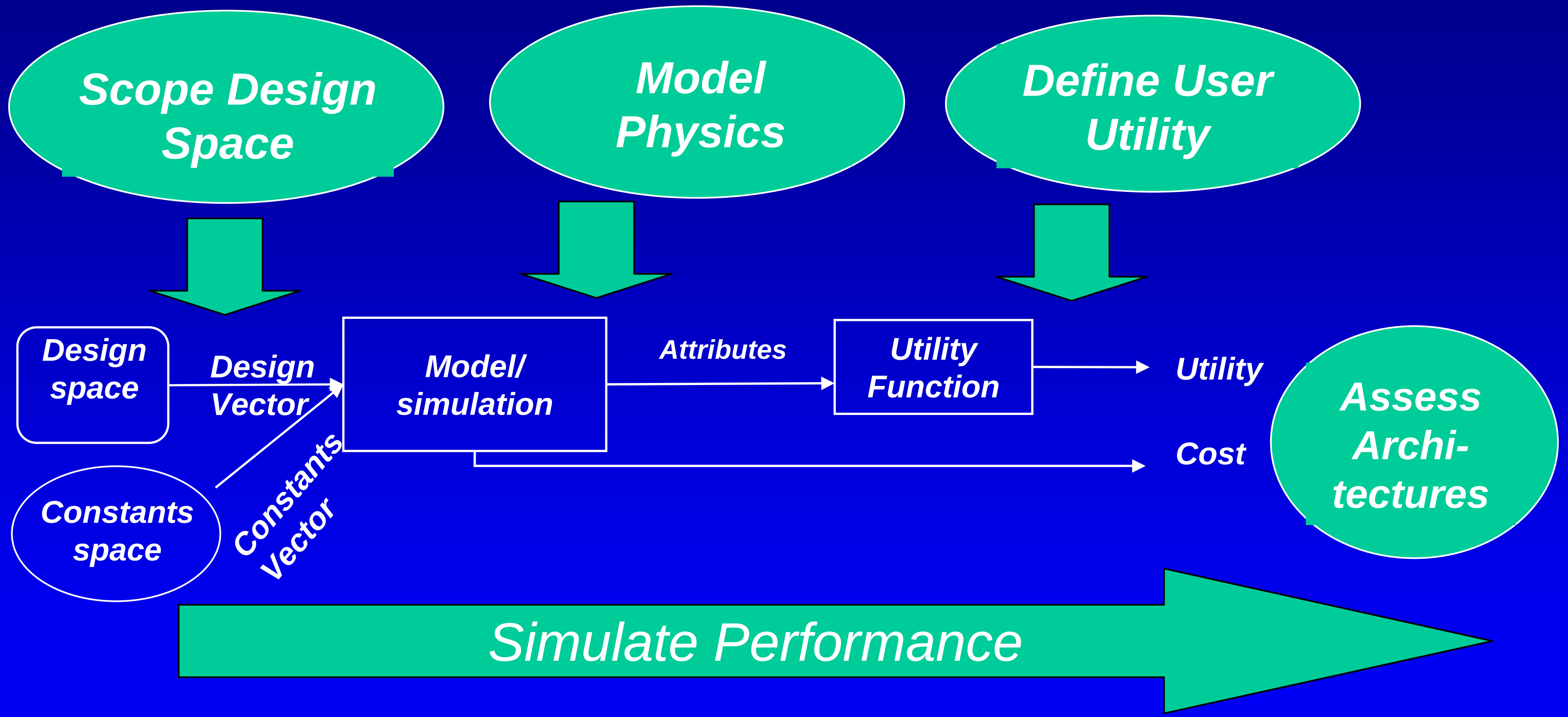
- Introduction
- Process Development
  - Overview of Design Process
  - Utility Attributes
  - Design Vector
  - Module Overview
- Results
- B-TOS Requirements
- Spacecraft Design
- Review and Concluding Remarks

Process/Tools Levels

- **Process:** The whole SSPARC* process, including gather user needs, scoping the problem, Multi-Attribute Utility Analysis (MAUA)
- **Simulation:** Design Vector to Utility and Cost
- **Modeling:** What the Code Does
- **Tools:** Matlab, N-Squared, QFD

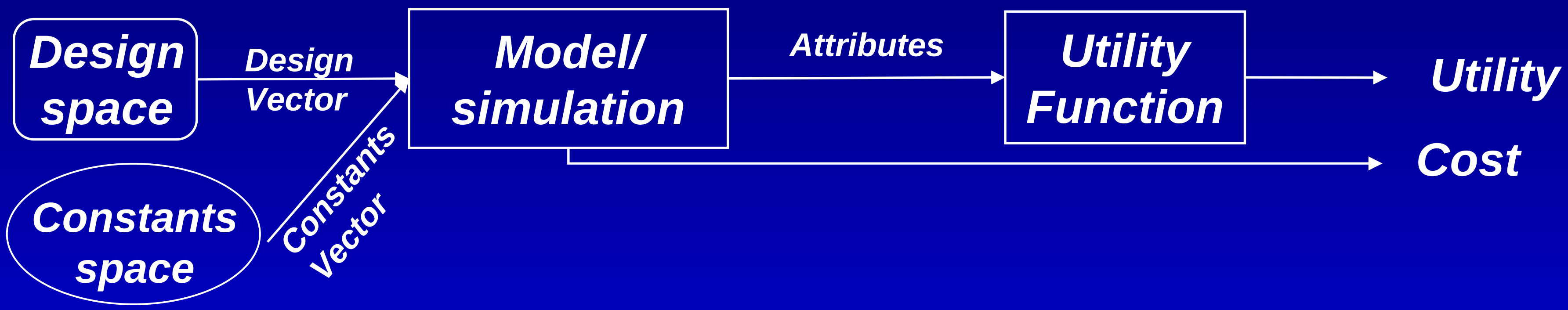
* Space Systems, Policy, and Architecture Research Center

B-TOS Process



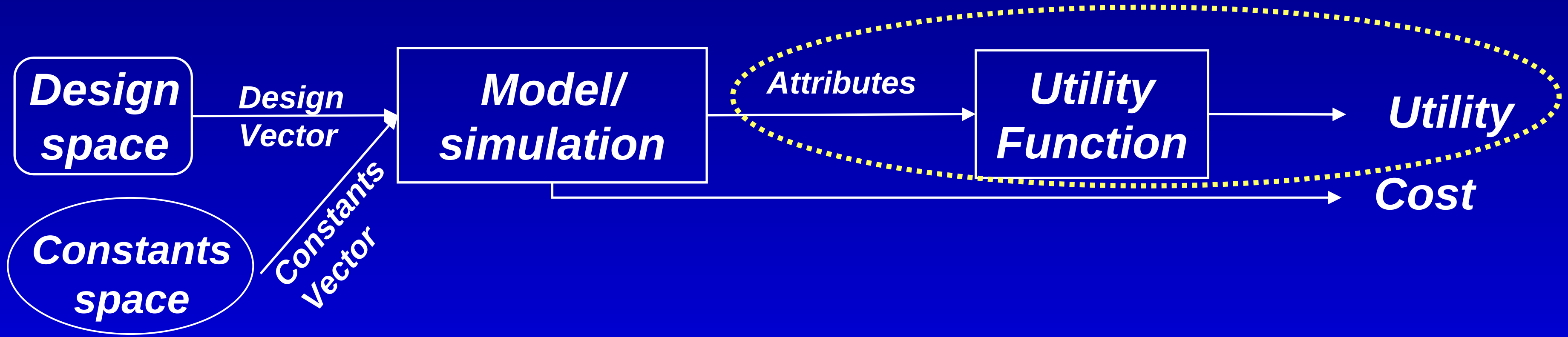
Many tasks done simultaneously with high level of interaction

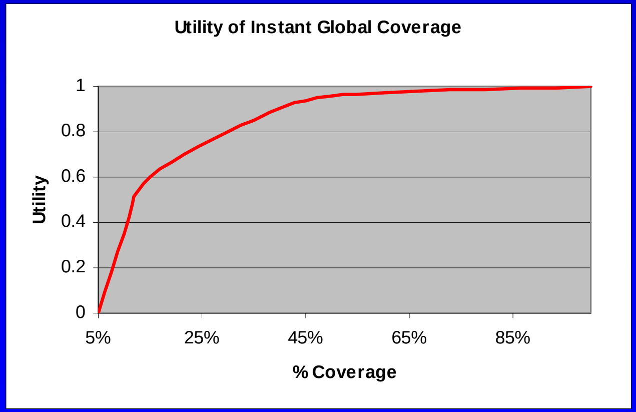
B-TOS Simulation Notional Flow



- **Inputs:**
  - Design vector (architecture)
  - Constants vector (engineering constants, fixed design parameters)
- **Outputs:**
  - Utility
  - Cost
- **Simulation developed using Matlab modules integrated with Satellite Tool Kit (STK)**

Multi-Attribute Utility Analysis





- Process overview
- Definition of attributes
- Customer preferences

Multi-Attribute Utility Analysis (MAUA)

Description

A process to capture complex customer preferences for attributes of architectures.

Application in B-TOS

Assisted generating output metric for architecture comparison (utility).

Strengths

- Defines customer-perceived attributes of architectures
- Captures trade-offs among preferences for different attributes
- Interactive process with customer helps refine customer needs
- Changes hard requirements to more flexible wants, resulting in better overall solutions

Limitations

- Lengthy interview process
- Difficulty thinking probabilistically
- More than 6 attributes at a time is infeasible for single interview

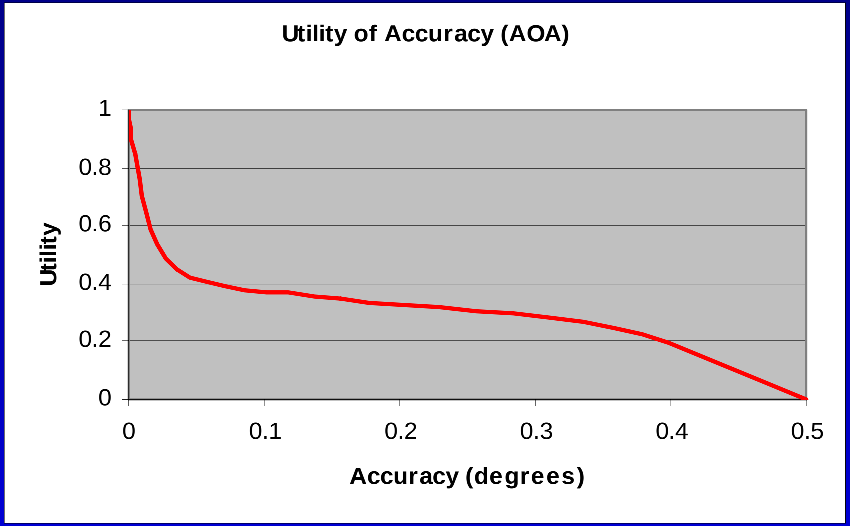
MAUA Process

- ✓ 1. Define attributes
- ✓ 2. Define attribute ranges (worst→best case)
- ✓ 3. Compose utility questionnaire
- ✓ 4. Conduct utility interview with Customer/User
  - Approx. 4 hours
  - Customers = Bill Borer, Kevin Ray (AFRL/VSB)
- ✓ 5. Conduct validation interview
  - Approx. 3 hours

Utility Attributes: MAUA Results

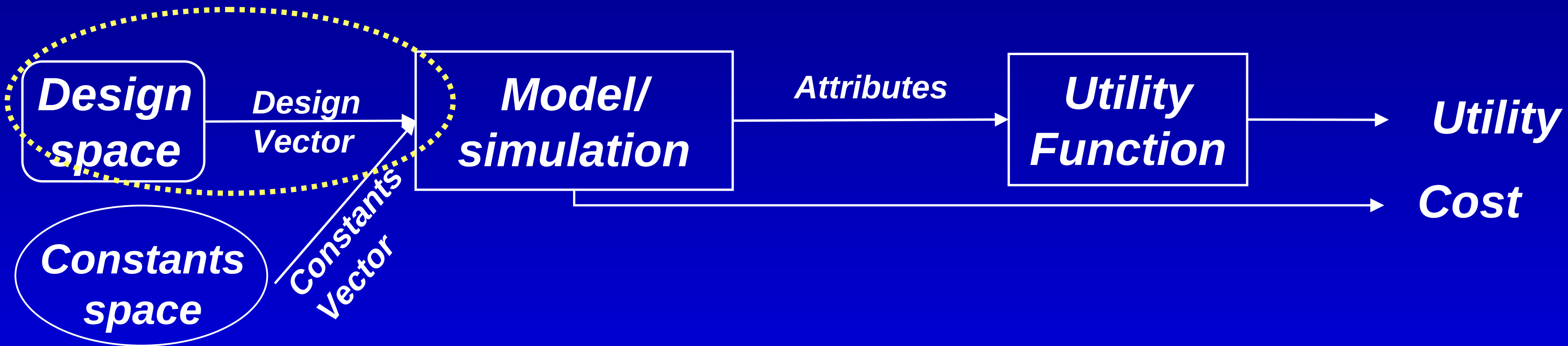
1. **Mission Completeness:** Sub-set of missions performed
2. **Spatial Resolution:** Arc length of Earth between measurements
3. **Revisit Time:** Time between subsequent measurements of the same point above the Earth
4. **Latency:** Time delay from measurement to end user
5. **Accuracy:** Measurement error in angle of arrival data
6. **Instantaneous Global Coverage:** % of Earth's surface in view between subsequent measurements

Converting Attributes to Utility



- Utility interview results in quantified relationship between attribute and usefulness of system to user (utility)
- Different curve for each attribute
- Mathematical combination of attribute values produces system utility

Design Vector



	VARIABLES	Units	CONSTRAINTS	Weight	1 Turbulence Mission Completeness (%)	2 Global Survey Mission Completeness (%)	3 Spatial Resolution (deg., km)	4 Time Resolution (Hz)	5 Latency (t)	6 Accuracy (dB)	7 Instantaneous Global Coverage (%)	TOTAL	8 Lifecycle Cost ($100M)	TOTAL w/ COST
1	Apogee Altitude	km	a > p		9	9	9	0	3	3	1	**34**	1	**35**
2	Perigee Altitude	km	a > p		9	9	9	0	3	3	1	**34**	1	**35**
3	Number of Planes	Integer			3	3	3	?	0	0	9	**18**	9	**27**
4	Swarm per Plane	Integer			3	3	3	?	0	0	9	**18**	9	**27**
5	Satellites per Swarm	Integer			3	3	9	1	0	0	1	**17**	9	**26**
6	Sub-Orbits per Swarm	Integer	concentric orbits									**0**		**0**
7	Size of Swarm	m			3	3	9	0	1	3	9	**28**	0	**28**
8	Sounding, [4]	Y/N			0	0	0	3	3	0	0	**6**	0	**6**
9	Number of Sounding Antennas	Integer	3 or 6		3	3	?	?	0	9	0	**15**	3	**18**
10	Short Range Communications, [4]	Y/N										**0**		**0**
11	Long Range Communications, [4]	Y/N			0	0	0	0	3	3	0	**6**	0	**6**
12	On-Board Processing, [2]	Y/N			0	0	0	0	3	3	0	**6**	0	**6**
13	Autonomy											**0**		**0**
	TOTAL				**33**	**33**	**42**	**4**	**16**	**24**	**30**		**32**	

- **Overview**
- **Tools employed**
- **Design vector**

Design Vector Overview

- Design vector provides fundamental (independent) variables that define architecture trade space
  - Focuses on variables that have significant impact on attributes
  - Design vector excludes model constants
  - Geometric growth of design space motivates curtailed list of design variables

- Provides a means for considering multitude of system architectures

Quality Function Deployment (QFD)

Description

A matrix to capture the relationship between attributes and design vector inputs.
Mechanism to weigh design vector options against each others.

Application in B-TOS

•Assisted generating design vector list. Prioritize technical requirements
•Provide requirement and attribute trace ability and booking keeping
•Provide a simple easy to understand communication mechanism

Strengths

•Expedite correlation of variables with attributes
•Rank order most critical variables and influence on attributes
•Reduce variable list to minimize trade space dimensionality
•Minimize human biases

Limitations

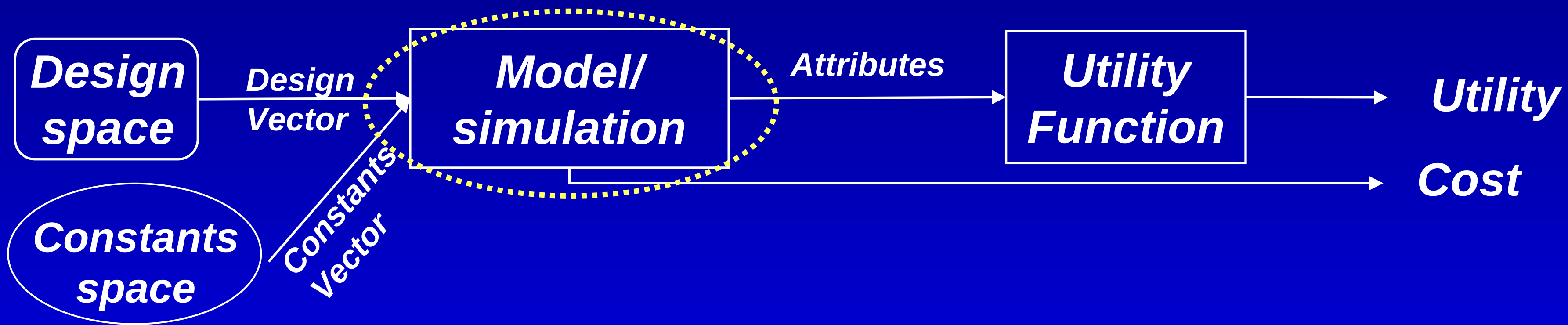
•Requires substantial technical knowledge and iteration to maximize usefulness
•Must be re-scaled when new customer requirements are added

Design Vector Variables

	Variable	Rationale
Large Scale Arch.	Apogee Altitude	Specifies orbit/relationship to ionosphere
	Perigee Altitude	Specifies orbit/relationship to ionosphere
	Number of Planes	Key to meeting global coverage needs
	Swarm per Plane	Key to meeting global coverage needs
Swarm Arch.	Satellites per Swarm	Local coverage resolution
	Size of Swarm	Local coverage resolution
Vehicle Arch.	Number of Sounding Antennas	Captures functionality trade
	Sounding	Captures functionality trade
	Short Range Communications	Captures functionality trade
	Long Range Communications	Captures functionality trade
	On-Board Processing	Captures functionality trade

- **Payload, four choices available:**
  - 0 = none
  - 1 = send
  - 2 = receive
  - 3 = both
- **Communication and processing, two choices available:**
  - 0 = none
  - 1 = yes (all)

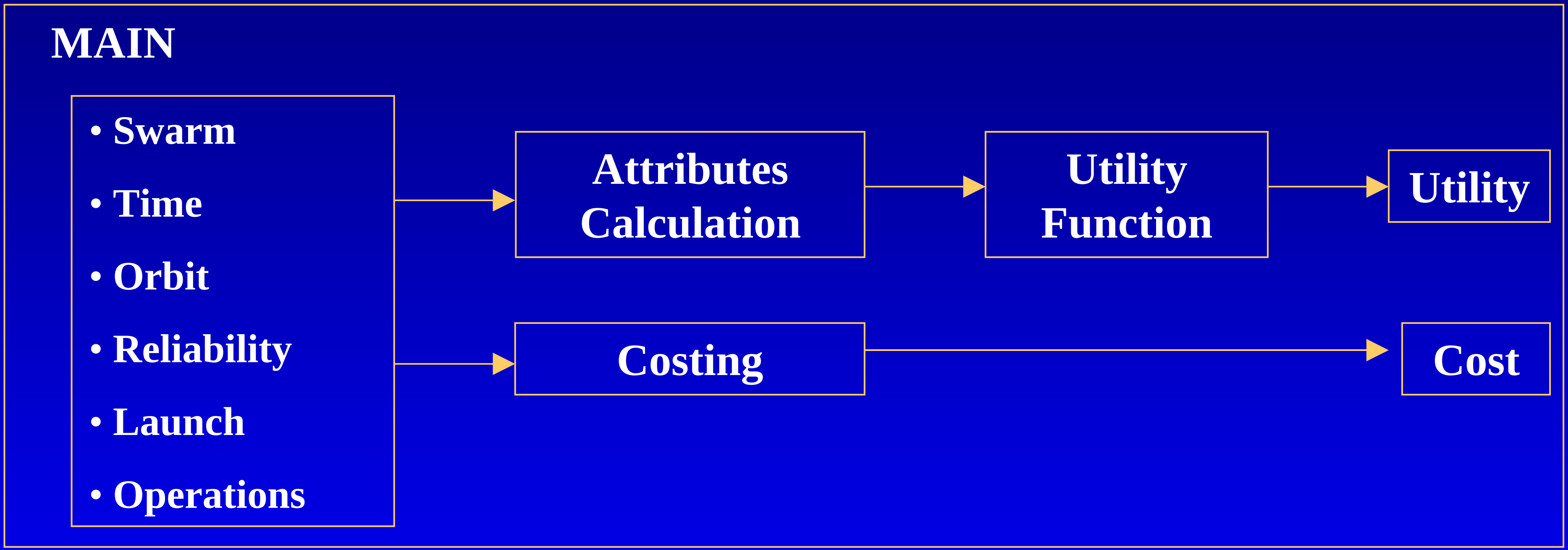
Modeling



- Code overview
- Tools employed
- Module descriptions

		Design	Constants	Swarm	Swarmrel	Reliability	Orbit	Orbitprop	Launch	Operations	Costing	Time	Calculate_Attributes	Spacecraft	Utility Function	output_BTOS
	Module Name	**D**	**C**	**SW**	**SWR**	**R**	**O**	**ORP**	**L**	**OPS**	**Cost**	**T**	**A**	**SC**	**U**	**out**
D	Design															
C	Constants															
SW	Swarm	x	x													
SWR	Swarmrel	x	x	x												
R	Reliability	x	x		x											
O	Orbit	x	x													
ORP	Orbitprop	x	x	x												
L	Launch	x	x	x												
OPS	Operations	x	x	x												
Cost	Costing	x	x	x					x	x						
T	Time	x	x		x											
A	Calculate_Attributes	x	x				x					x				
SC	Spacecraft	x	x													
U	Utility Function	x	x										x			
out	output_BTOS	x	x	x		x	x		x		x		x		x	

Organization Rationale



- Attribute calculation separated from space system parameters
  - Attributes are mission specific
  - Enhances code generality and reusability

Teams and Responsibilities

Teams are organized based on software modules

Module	Primary Rep.	Secondary Rep.
Main	Adam Ross	Qi Dong
BTOS Shell	Adam Ross	Qi Dong
Orbit	Scott Kimbrel	Sandra Kassin-Deardorff
Swarm	Nathan Diller	Brandon Wood
Spacecraft	Brian Peck	Nathan Diller
Launch	Dan Kirk	Brian Peck
Operations	Brandon Wood	Nathan Diller
Reliability	Dan Kirk	Michelle McVey
Costing	Michelle McVey	Dan Kirk
Attribute	Carole Joppin	Brandon Wood
Utility	Adam Ross	Carole Joppin
Integration	Qi Dong Adam Ross	

Unified Modeling Language (UML)

Description

A set of software design diagramming tools.

Application in B-TOS

Assisted designing high level software modules and visualizing module interactions.

Strengths

- Stresses the importance of user needs
- Shows the interactions among modules
- Facilitates system architecture design

Limitations

- Difficult to implement at detailed coding level when the programming language is not Object-oriented.

Software I/O Management Workbook

Description

A set of Microsoft Excel spreadsheets to record the interface variables of each software module.

Application in B-TOS

Assisted the interface management of software modules.

Strengths

- Excel is easy to use and to program
- Embedded Visual Basic program automatically checks the consistency of the software I/O.
- Enabled communication among module development teams for integration purpose.

Limitations

- Accuracy depends on how up-to-date the module I/O sheets are.

N^2 Diagram

Description

A square matrix that captures the information flow among system elements.

Application in B-TOS

Assisted the system interface management and integration of the software modules.

Strengths

- Exposes the iteration loops among the modules and facilitates design simplifications.
- Assisted the integration of the modules.

Limitations

- Good system analysis tool, but limited in predicting system interactions.

Module Descriptions

- **Swarm:** Calls on the spacecraft module to define the spacecraft parameters for the entire swarm
- **Reliability:** Determines probability that a particular number of satellites are operational in any swarm at a given time
- **Time:** Determines mission, accuracy, and latency
- **Orbit:** Propagates orbital trajectories from initial conditions
- **Attributes:** Calculates value of 6 attributes at three different times (BOF, mid-mission, EOF) for utility function

Module Descriptions

- **Utility:** Calculates "value" tradeoffs of the attributes (metrics) for a particular architecture
- **Operations:** Calculates operations personnel and facilities costs
- **Launch:** Selects lowest cost launch vehicle(s) that can deploy all satellites in a swarm
- **Costing:** Calculates spacecraft, operations, launch, and program level costs, incorporates learning curve for different spacecraft types

Example Module: Launch

Description

•Selects lowest cost launch vehicle(s) that can deploy all satellites for a single swarm.

•Once a launch vehicle is selected, total cost for initial deployment is computed.

Key Assumptions

•Launch vehicle and cost is function of: number of satellites/swarm, stowed dimensions of satellite, orbital altitude, launch vehicle mass capacity, and launch vehicle payload dimensions

•Assumes 100% launch success rate

Fidelity Assessment

•First iteration makes use of average satellite mass

•Considers 6 different launch vehicle possibilities and 14 altitudes

Verification

•Tested over range of satellite numbers, satellite masses and swarm sizes

•Fully integrated into B-TOS master design code

Section Summary

- Review
  - Motivation
  - Project scope
  - Objective
- Process Development
  - Overview of Design Process
  - Utility Attributes
  - Design Vector
  - Module Overview

Results


Design space
Design Vector
Constants space
Constants Vector
Model/ simulation
Attributes
Utility Function
Utility
Cost

Section Outline

- Introduction
- Process Development
- Results
  - Architecture Survey Results
  - Sensitivity Analysis
- B-TOS Requirements
- Spacecraft Design
- Review and Concluding Remarks

B-TOS Model Analytical Capability

- Variation of orbital geometries
- Multiple swarm size and density options
- Satellites have individually varying functionality

Model currently produces a focused tradespace, not a single-point architecture

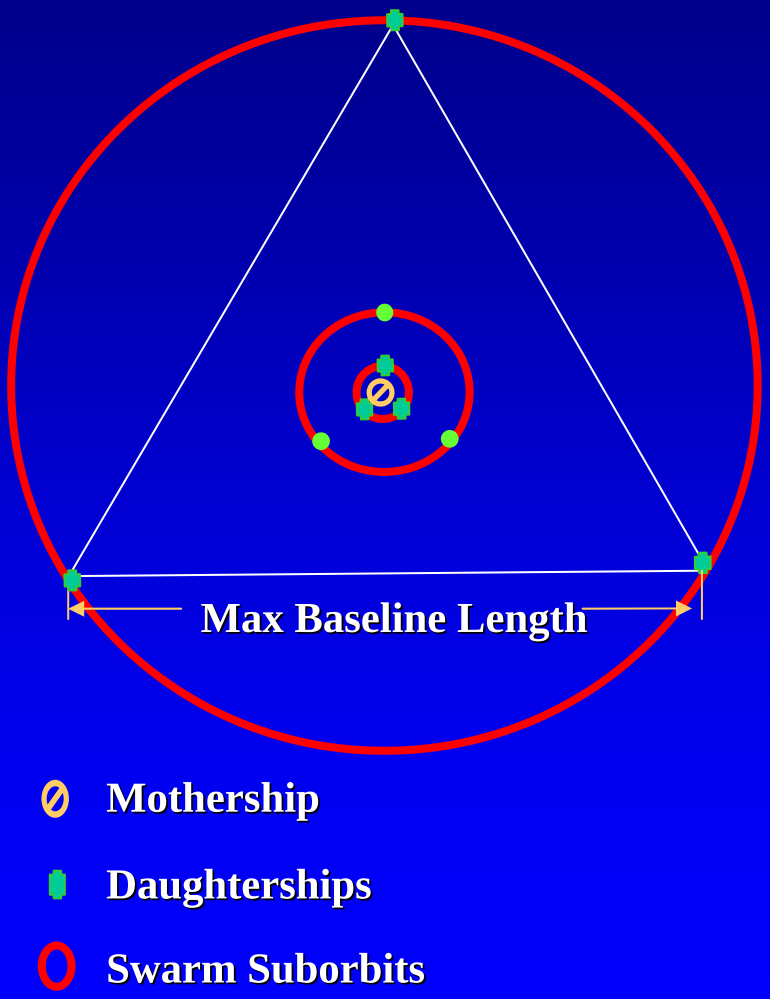
Tradespace Enumeration

• Circular orbit altitude (km)	1100, 1300
• Number of Planes	1, 2, 3, 4, 5
• Number of Swarms/Plane	1, 2, 3, 4, 5
• Number of Satellites/Swarm	4, 7, 10, 13
• Radius of Swarm (km)	0.18, 1.5, 8.75, 50
• 5 Configuration Studies	Trades payload, communication, and processing capability

4,033 Architectures

73 Hrs total computation time with 8 Pentium IIIs

Swarm Geometry



- Max baseline length is defined by desired angle of arrival accuracy (.0005 degrees)
- Minimum baseline length limited by beacon frequency (100 MHz)
- Swarm suborbit spacing is a factor of 5.7 and defined by:
  - Phase error of the swarm
  - Frequency of the beacons
- Filling the baselines ensures no ambiguity in the angle of arrival measurement

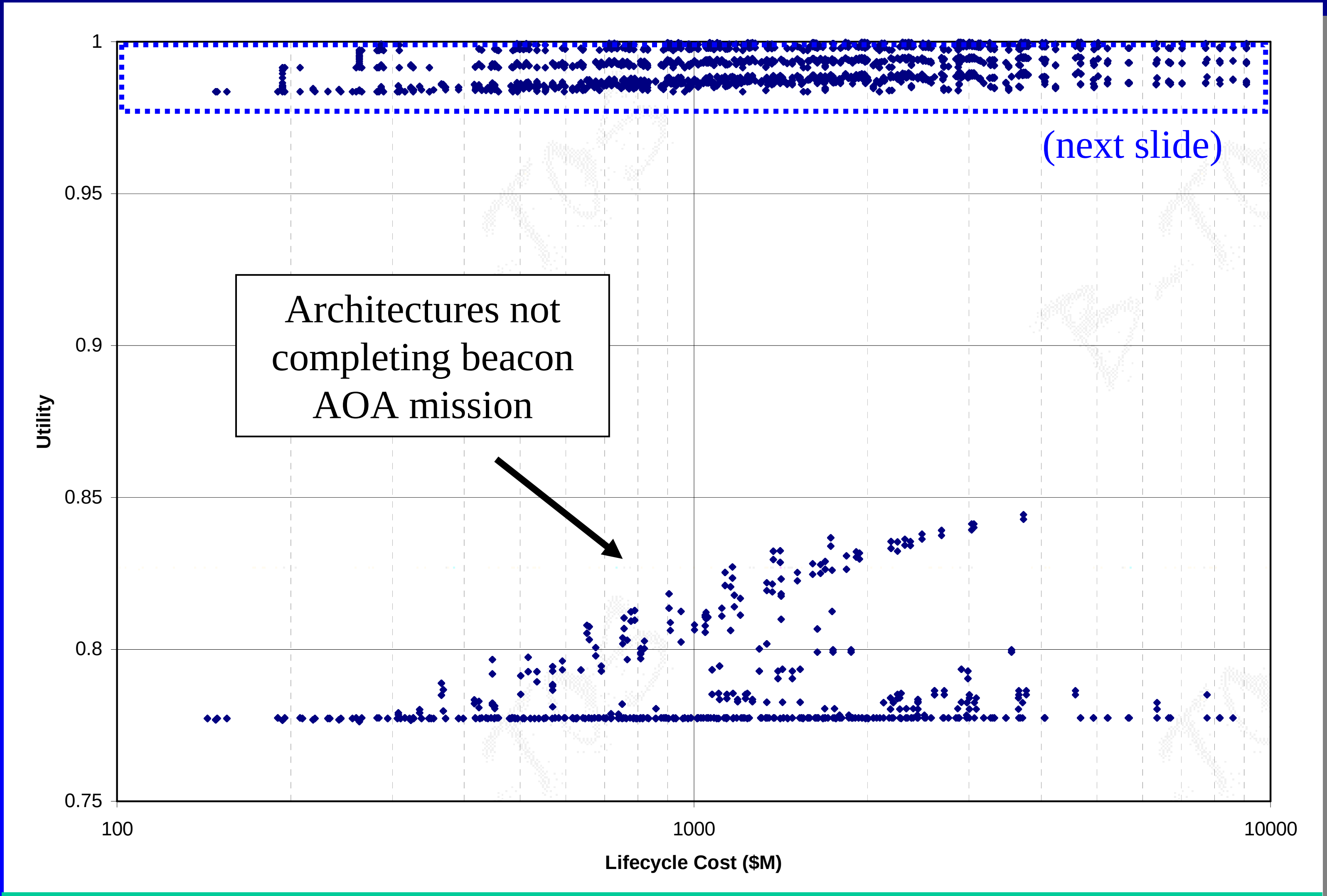
Configuration Studies

Study	1		2		3		4		5	
Type	M	D	M	D	M	D	M	D	M	D
Number	**4+**	**0**	1	3+	1	3+	1	3+	1	3+
Payload (Tx)	Yes	n/a	Yes	Yes	Yes	Yes	**No**	Yes	**Yes**	**No**
Payload (Rx)	Yes	n/a	Yes	Yes	Yes	Yes	**No**	Yes	**Yes**	Yes
Processing	Yes	n/a	Yes	**No**	Yes	**Yes**	Yes	No	Yes	No
TDRSS Link	Yes	n/a	Yes	**No**	Yes	No	Yes	No	Yes	No
Intra-Swarm Link	**No**	n/a	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes

M = Mothership **D = Daughter**

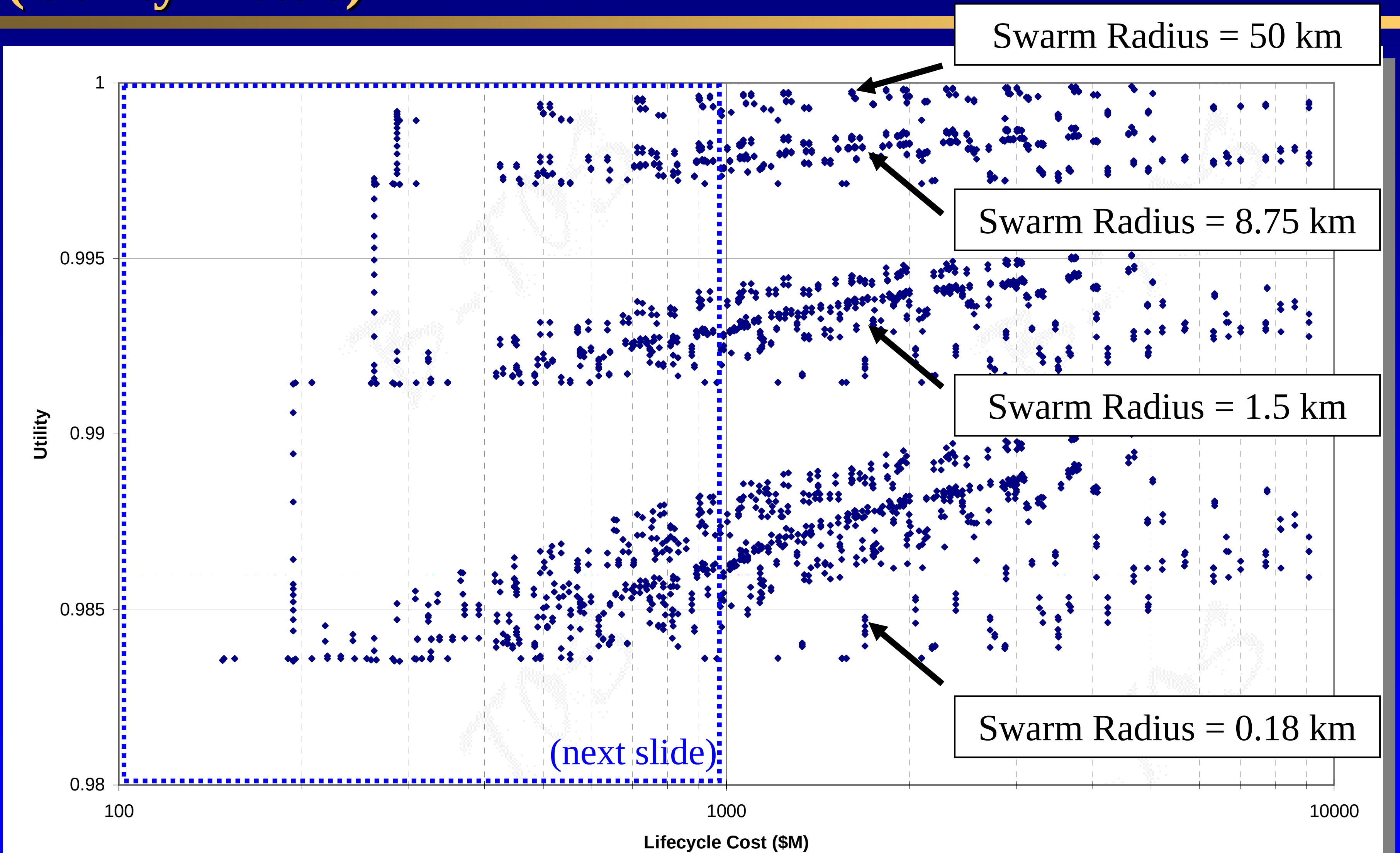
- **Study 1: All spacecraft are independent**
- **Study 2: Mothership processes and downlinks**
- **Study 3: Distributed processing**
- **Study 4: Mothership dedicated to processing and downlink (no payload)**
- **Study 5: Mothership processes, downlinks, and has payload transmitter**

Lifecycle Costs vs. Utility
(Entire Tradespace: 4,033 Architectures)



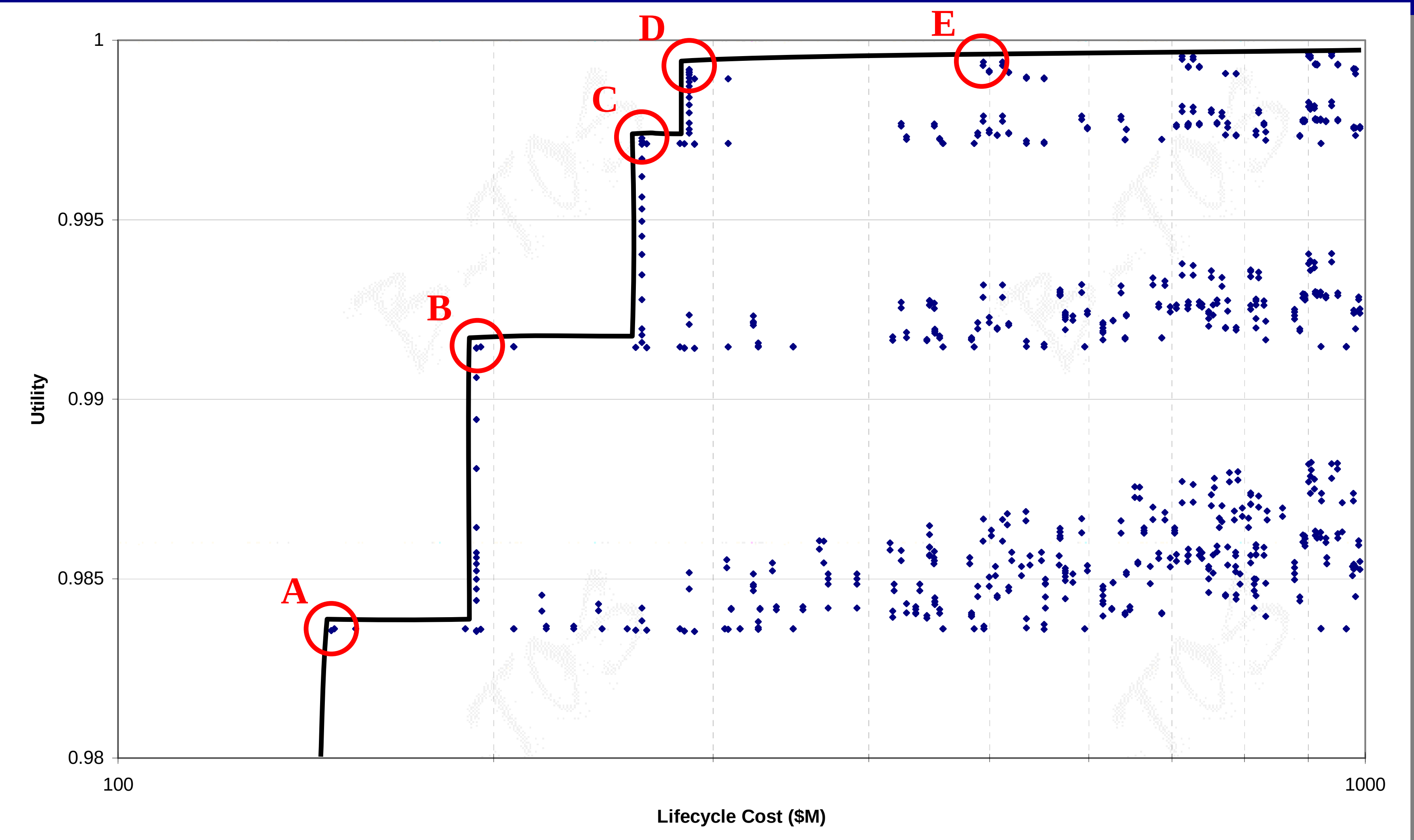
Completing AOA mission is main driver for utility

Lifecycle Costs vs. Utility
(Utility > 0.98)



Radius of the swarm is the main differentiator between architectures of high utility

Lifecycle Costs vs. Utility
(Frontier Architectures)

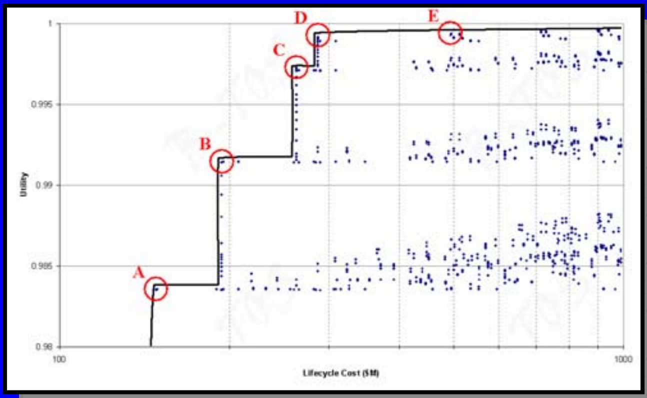


Frontier architectures are the most desirable

Frontier Architectures

Point	A	B	C	D	E
Altitude (km)	<-- 1100 -->				
Num of Planes	<-- 1 -->				
Swarms/Plane	1	1	1	1	2
Satellites/Swarm	4	7	10	13	13
Swarm Radius (km)	0.18	1.5	8.75	50	50
Functionality Study	<-- #5 -->				

Recall:



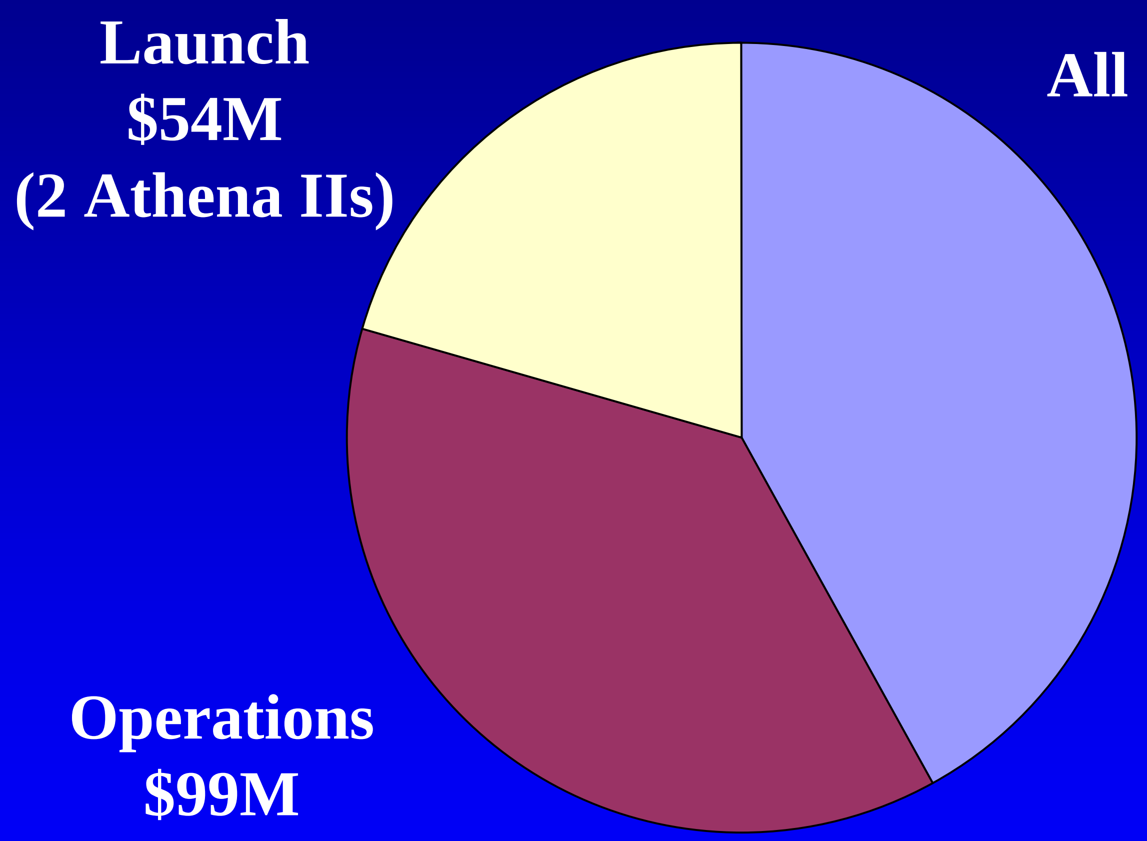
Study	5	
Type	M	D
Number	1	3+
Payload (Tx)	Yes	No
Payload (Rx)	Yes	Yes
Processing	Yes	No
TDRSS Link	Yes	No
Intra-Swarm Link	Yes	Yes

Frontier Attributes, Utility, & Cost

Point	A	B	C	D	E
Spatial Resolution (deg)	4.36	5.25	7.34	9.44	9.44
Revisit Time (min)	805	708	508	352	195
Latency (min)	3.40	3.69	4.36	5.04	5.04
Accuracy (deg)	0.15	0.018	0.0031	0.00054	0.00054
Inst. Global Coverage	0.29%	0.29%	1.15%	2.28%	4.55%
Utility	0.9835	0.9914	0.9973	0.9992	0.9994
IOC Cost ($M)	90	119	174	191	347
Lifecycle Cost ($M)	148	194	263	287	494

Frontier architectures can be evaluated using attributes in place of nondimensional utility values

Cost Breakdown: Point C



Total Lifecycle: $263M

Architecture Analysis Summary

- Architecture must collect beacon angle of arrival data to be in best part of tradespace
- Swarm radii become key differentiator between optimum architectures
- Most promising trades revolve around
  - Simple orbit geometry
  - Single swarm missions
  - Consolidating functionality on mothership
    - Complicated mothership (payload processing, payload transmitter, and TDRSS link)
    - Simple daughterships

Sensitivity Analysis Rationale

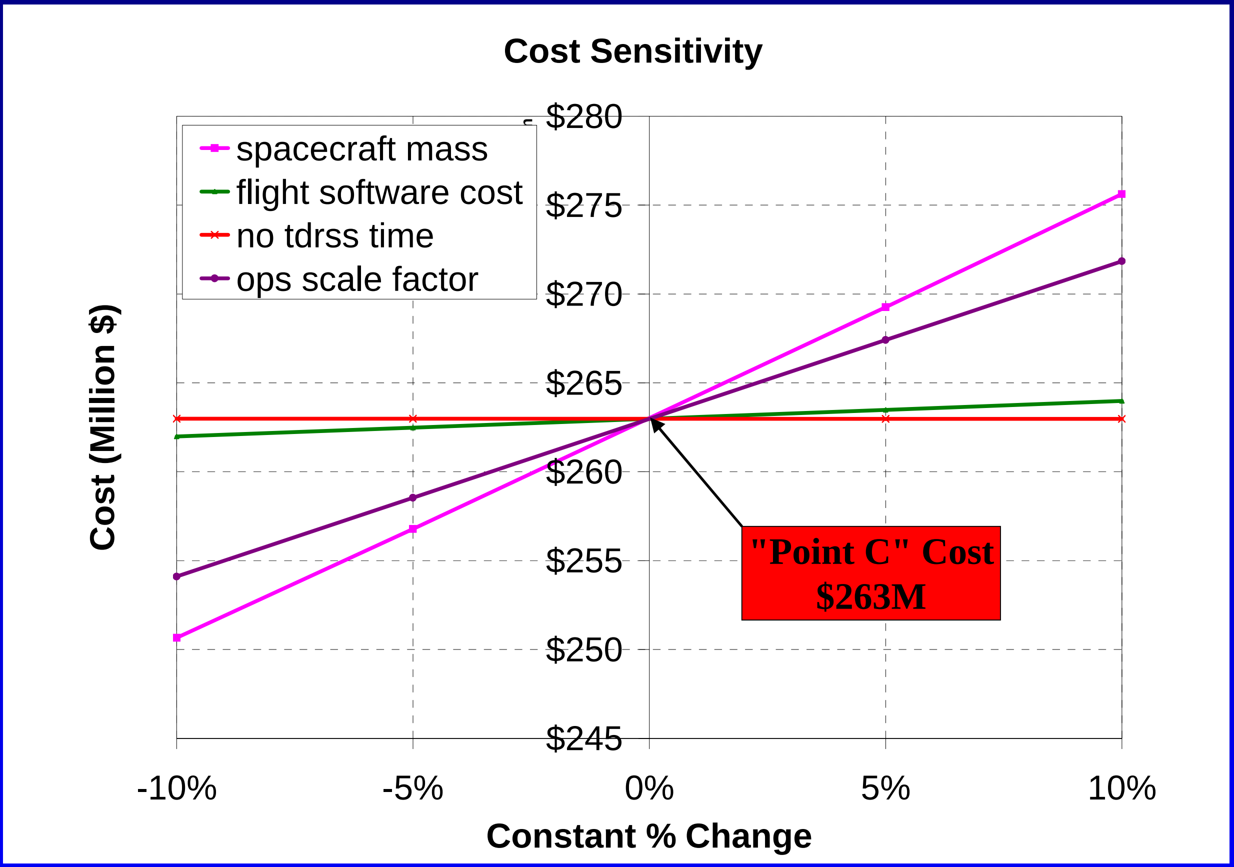
- Study dependence of cost and utility on the main constants
- Test sensitivity to the main assumptions used in the code

Sensitivity analysis validates results and optimum architectures

Parameters Studied

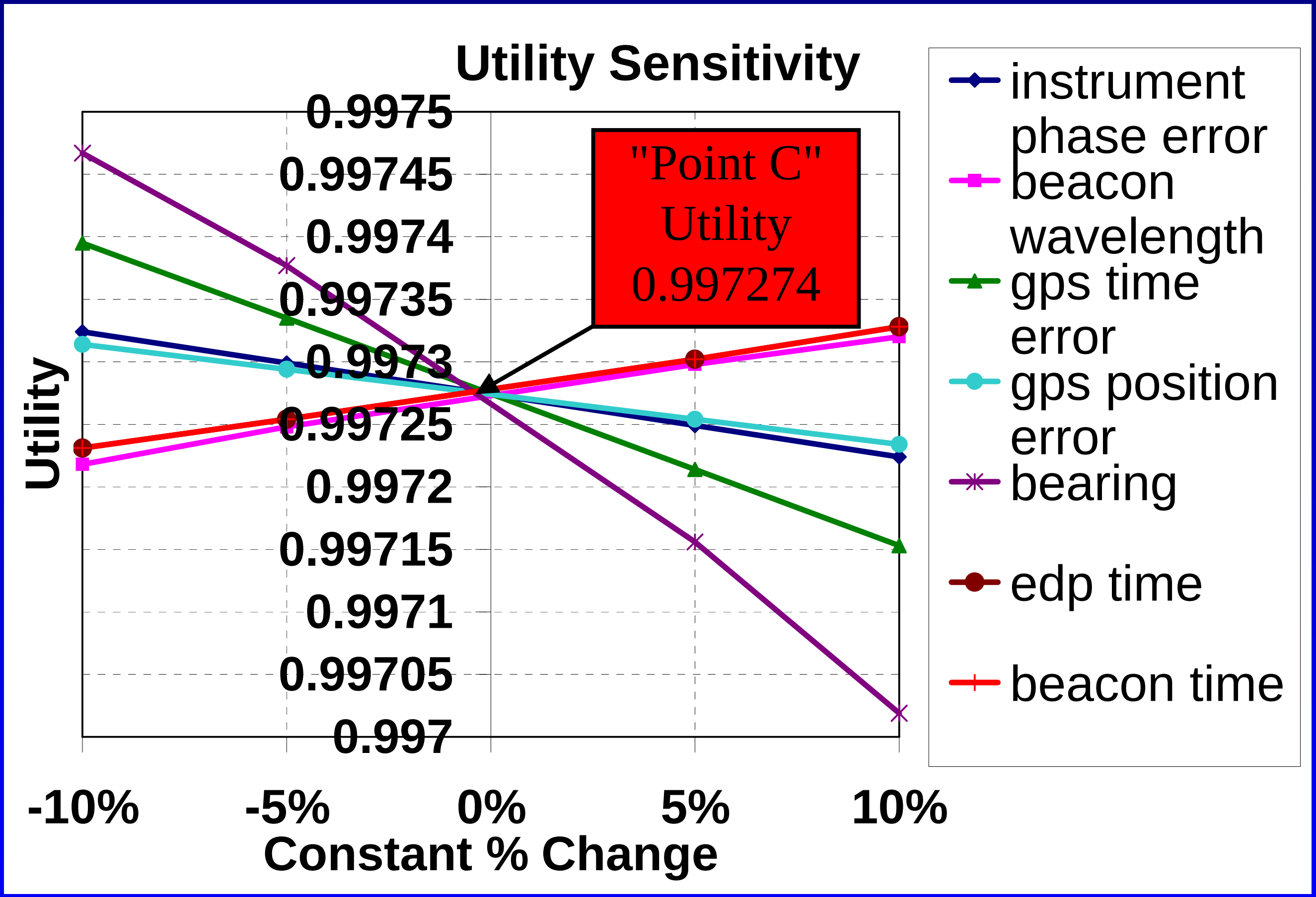
- Focused analysis of Multi-Attribute Utility function
- Varied 12 previously constant parameters by ±5% and ±10%
  - Spacecraft mass
  - Instrument phase error
  - Beacon wavelength
  - GPS time error
  - GPS position error
  - Assumed bearing
  - Flight software cost
  - EDP data collection time
  - Beacon data collection time
  - Maintenance time factor
  - No TDRSS time factor
  - Ops scale factor
- Varied 4 previously constant parameters
  - MTTF
  - Mission life
  - Dataset delay
  - Turbulence data collection time
- Swarm geometry and Delta V implications

Cost Sensitivity



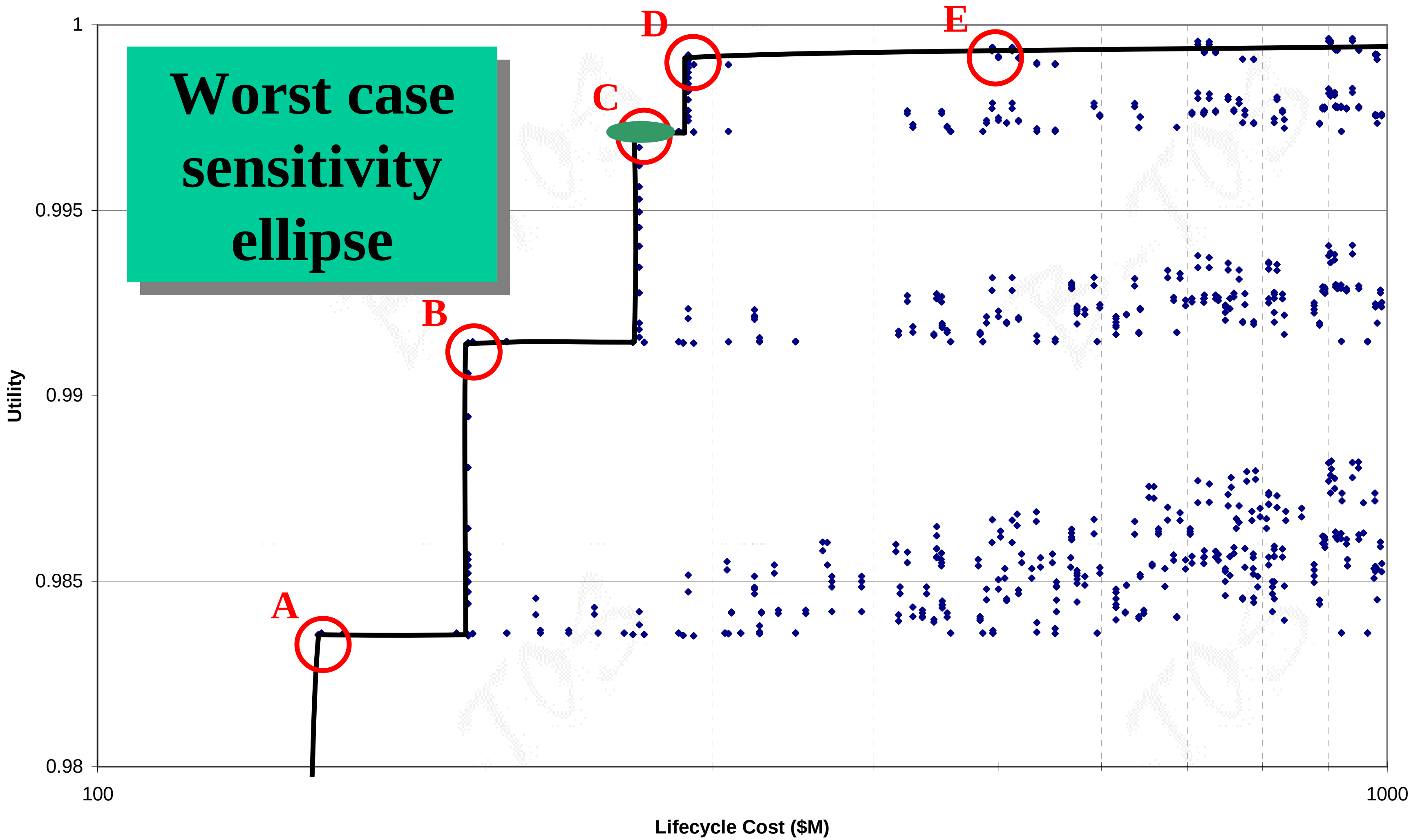
- Cost is most sensitive to S/C mass
  - –10 % change in S/C mass results in $15 M shift
- Cost is less sensitive to ops scale factor

Utility Sensitivity



Even largest error maintains same architecture choice

Frontier Architecture Sensitivity Summary



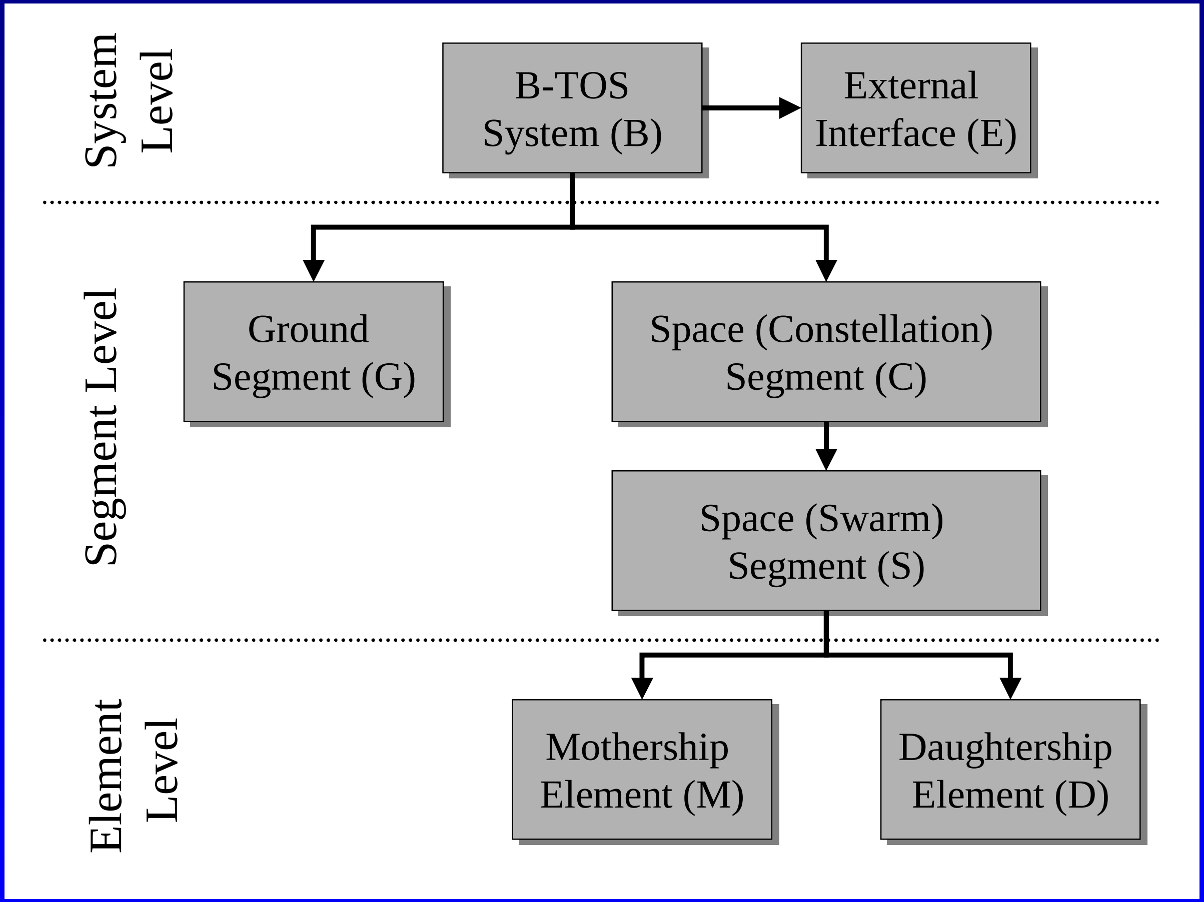
Even largest error maintains same architecture choice

System Level Requirements

Presentation Outline

- Introduction
- Process Development
- Results
- B-TOS Requirements
  - System Level
  - Segment Level
  - Element Level
- Spacecraft Design
- Review and Concluding Remarks

Requirements Hierarchy



Refer to appendix C of design document for complete B-TOS requirement document

System Level Requirements

- B-TOS system level
  - Contents: Mission characterization, payload B, US launch vehicle, lifetime, TDRSS
  - Example: The B-TOS system shall have the capability to collect data from the topside of the ionosphere below 1100 km to produce an Electron Density Profile (EDP).
- External requirements
  - Contents: Constrained to interface with external systems (TDRSS, beacons, US launch vehicle) and compatibility with the AFRL model
  - Example: The B-TOS space system will be capable of communicating with TDRSS

Segment Level Requirements

- Constellation
  - Contents: Constellation orbital parameters, revisit time and global coverage
  - Example: The constellation shall have one plane.
- Swarm
  - Contents: Swarm configuration and geometry, communication, accuracy and spatial resolution
  - Example: Each swarm shall have ten satellites consisting of 1 mothership and 9 daughterships.
- Ground
  - Contents: Scheduling, communication, telemetry, payload data processing, command and control
  - Example: The operations center shall perform mission scheduling.

Element Level Requirements

- **Mothership**
  - **Contents: Mission capability, functionalities, communication, data compression**
  - **Example: The mothership shall have a communication subsystem capable of sending data at 5 Mbps and receiving data at 100 kbps with the ground via TDRSS' S-band single access antennas at 10^{-6} bit error rate.**
- **Daughtership**
  - **Contents: Mission capability, functionalities, communication**
  - **Example: The daughtership shall have a communication subsystem capable of sending data at 1.2 Mbps and receiving data at 10 kbps with the mothership.**

Spacecraft Design

Spacecraft Design Overview

Designed mothership for architecture "C"

- **B-TOS Spacecraft Design Process**
- **Preliminary Mothership Design Results**

B-TOS Mothership Design Process

- Utilized simplified Integrated Concurrent Engineering (ICE) design method
  - ICE method developed at Cal Tech
- Each individual assigned a subsystem
- Excel spreadsheets created to spec each subsystem using SMAD equations
- N^2 diagram created to determine subsystem dependencies and flow of calculations
- One iteration completed and a first-order mothership design specified

Subsystem Breakup and Descriptions

Sub-system	Requirement	Approach	Who
Power	Full ops at end of life, peak and avg	Size battery and solar cell	Carole
Thermal	Acceptable temp range at eol, temp range	Energy balance	Adam
Payload	List from customer	Set requirements for other systems	
Comm	Comm through TDRSS and with all daughters	Link budget	Scott, Brandon
Attitude	Set by payload	Select and size sensors, wheels, and motors	Nathan
Structure	Not fail or resonate	15% mass fraction budget	Hugh
C.D.H	Support operations, survive environment	Recall ops scenarios, develop link budget inputs, select and size computers and recorders	Scott, Brandon
Propulsion	Provide deltaV and max impulse to support ops scenarios	Select and size motors, possibly combined with attitude, consider drag, deorbit, margin, NOT differentials	Brian, Hugh
Configuration	Fit in launch vehicle and config in 3D	Sketch or CAD	Sandra
Mass	Launchable	Sum up systems' masses	Hugh
Reliability	No single-point failures of vulnerable systems	Check batteries, computers, sensors, thrusters, thermal	Dan
Cost	Not exceed reasonable cost	SMAD cost estimating relationships	Michelle

N^2 Diagram – Subsystem Info Flow

	Payload	Attitude	C.D.H.	Comm.	Therm.	Prop.	Config.	Power	Mass	Struct.	Reliab.	Cost
Payload	X											
Attitude	R	X		R				f	f		f	f
C.D.H.	R		X	R				f	f		f	
Comm.	R	I	I	X			I	f	f		f	f
Therm.		I	R	R	X	R	I	I	f			f
Prop.		I				X	I	I	f		f	f
Config.	R	I		I	I	I	X	B				
Power	R	B	B	B	B	B		X	f		f	f
Mass	R	B	B	B	B	B		B	X			
Struct.									I	X		
Reliab.		I	I	I		I		I			X	
Cost	I	I	I	I	I	I		I	I	I		X

I = Input **R = Input from Requirements**

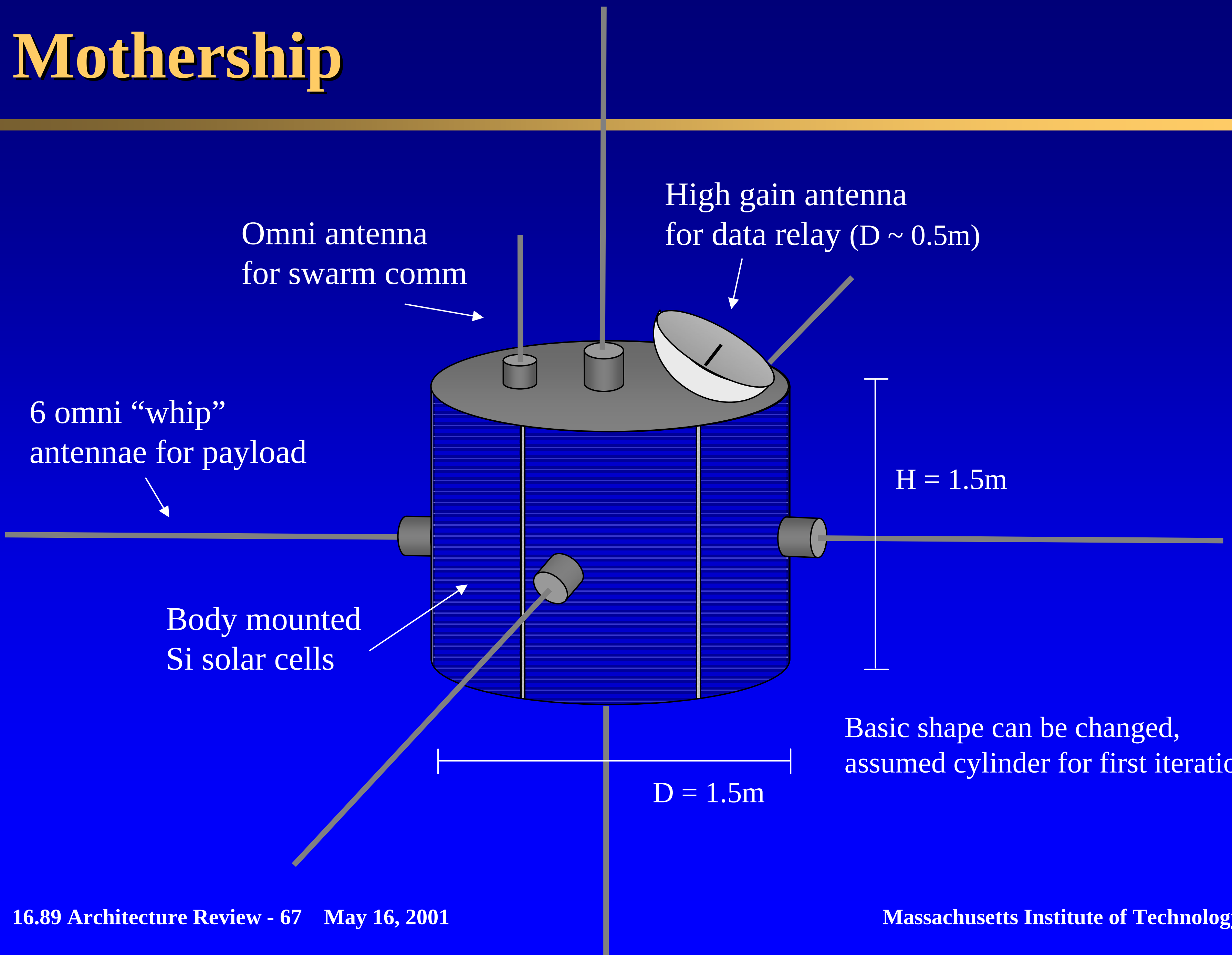
B = Budget **f = Possible feedback**

Info flows from columns Into rows

Preliminary Mothership Design Results

Sub-system	Spec	Power	Mass	Cost
Payload	6 omni antenna plus transceivers	64W	36kg	N/A
Attitude	3-axis momentum wheels	20W	7kg	\$9.8M (±4.4)
C.D.H.	Computers plus data storage	14W	5kg	\$6M (±2.4)
Comm	0.5m diameter antenna	10W	20kg	\$3M (±0.6)
Thermal	0.32m^2 radiator plus radiative paint	1.3W	4.5% dry mass	\$8M (±1.4)
Propulsion	12 PPT thrusters	40W	20kg dry plus 7.30kg fuel	\$6.5M (±1.5)
Configuration	Cylinder (D=H=1.5m)	N/A	27kg (structure plus thermal)	\$1.6 (±1)
Power	2.5m^2 Si body mounted solar arrays 4 NiCd batteries	Total Power Req: 150W	33.5kg	\$16.7M (±7.1)
Mass	Sum of all systems	N/A	Totals: 185kg dry 193kg w/ fuel 208kg boosted	N/A
Reliability	N/A	N/A	N/A	N/A
Cost	SMAD cost est. relationships (CERs)	N/A	N/A	S/C Total: \$45M (±19)

Mothership



Basic shape can be changed, assumed cylinder for first iteration

Preliminary Mothership Design Results

- Spacecraft for architecture "C" appears to be feasible.
- Mass was up 17%, and power down 21%, from estimates made as part of the architecture study
- Mothership cost (~$45M) is a significant fraction of the total spacecraft budget (from the architecture study, ~$101M)
- Comm. requirements were severe for TDRSS relay (~10Mbps) and would compete with ISS and Shuttle
- Body mounted solar cell area approaching limit for power needs (~150W)

Summary

Design space
Design Vector
Constants space
Constants Vector
Model/ simulation
Attributes
Utility Function
Utility
Cost

Strategy and Process (1)

1. Collect stakeholder value propositions
   - ✓ Professors
   - ✓ Customer
   - ✓ Students
2. Develop mission statement
3. Develop utility function
   - ✓ Create list of system attributes
   - ✓ Conduct utility function interview
   - ✓ Create utility function based upon customer responses

Strategy and Process (2)

4. **Define design space**
   - ✓ Create list of design variables (design vector)
   - ✓ Map design variables to system attributes using QFD to determine which variables will be important
   - ✓ Eliminate extraneous variables to make a design vector of manageable size
   - ✓ Define design space by determining appropriate ranges for design vector variables, using available technologies, physical and system constraints

Strategy and Process (3)

5. Develop model of the system
   - ✓ Define metrics to be evaluated
   - ✓ Partition the problem into modules that calculates system attributes based upon design vector inputs
   - ✓ Integrate modules into a single model
6. Evaluate all possible meaningful architectures with respect to the utility function
   - ✓ Use model to iterate across design space and evaluate utility of all architectures
   - ✓ Select architecture(s) that best fit customer needs
7. Design space system based upon selected architecture

Accomplishments

- B-TOS mission characterized and defined
- Key attributes of swarm architectures determined
- Thousands of architectures traded
  - Captured "goodness" of architecture through utility analysis
- Code is robust and modular:
  - Easy to upgrade
  - Can accommodate distinct satellite types with different functionality combinations
- Optimal architectures identified
- Narrowed tradespace facilitates future analysis and direction
- Sensitivities and design studies point to challenges, but basically validate design
- Requirements derived for a potential architecture

Lessons Learned

- Process validated
  - Helps to surface issues early
  - Forces solution with traceable decision rationale
- Communication was key!
  - Iteration with customer was vital because of mission complexity—learning process for us and AFRL
  - Facilitated by web-based tools and early emphasis on integration of code
  - Hindered by lack of suitable lexicon and evolving definitions

Consistent and clear communication proved indispensable

Backup Slides

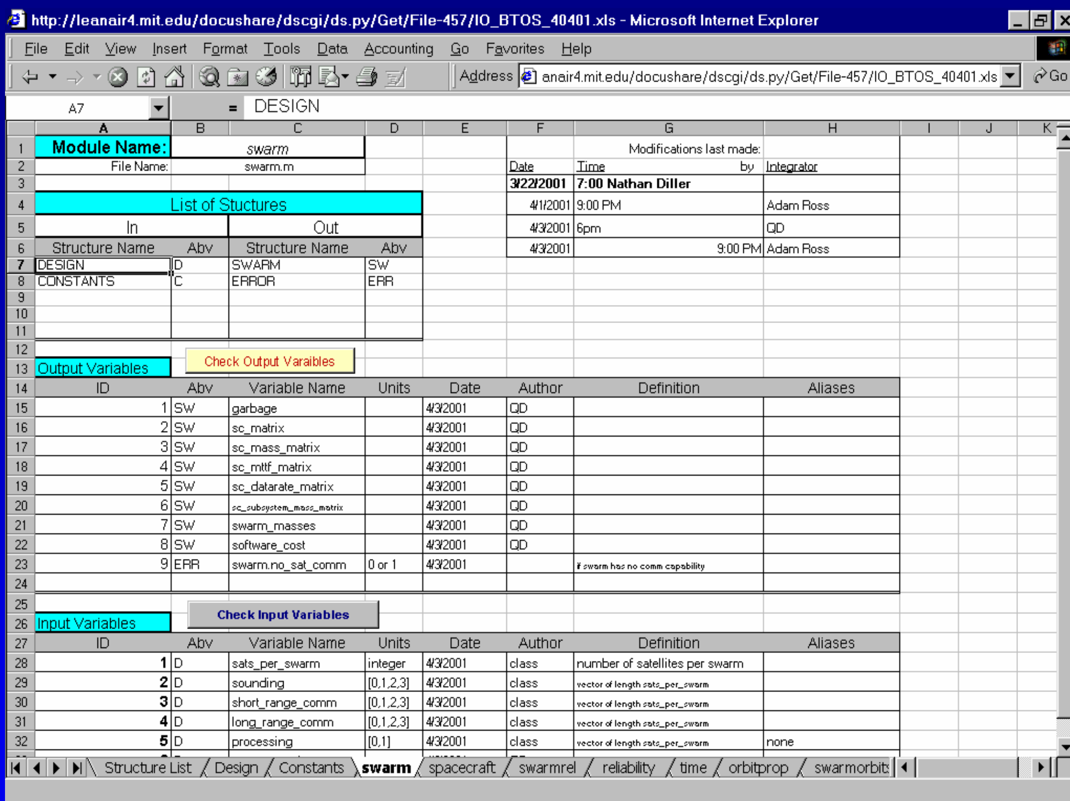
Appendix Slides

- Integration Process and Tools

Integration Process

- Started coding process with A-TOS modules
- Developed and maintained I/O sheets to manage interface consistency between modules
  - Facilitated communication between teams through I/O sheets
- Constructed N^2 diagram to show information flow between modules

I/O Sheets



http://leanair4.mit.edu/docushare/dscgi/ds.py/Get/File-457/IO_BTOS_40401.xls - Microsoft Internet Explorer

A7 = DESIGN

Module Name: swarm

File Name: swarm.m

Modifications last made:

Date	Time	by	Integrator
3/22/2001	7:00 Nathan Diller		
4/1/2001	9:00 PM		Adam Ross
4/3/2001	6pm		QD
4/3/2001	9:00 PM		Adam Ross

List of Stuctures

In		Out	
Structure Name	Abv	Structure Name	Abv
DESIGN	D	SWARM	SW
CONSTANTS	C	ERROR	ERR

Output Variables — Check Output Varaibles

ID	Abv	Variable Name	Units	Date	Author	Definition	Aliases
1	SW	garbage		4/3/2001	QD		
2	SW	sc_matrix		4/3/2001	QD		
3	SW	sc_mass_matrix		4/3/2001	QD		
4	SW	sc_mttf_matrix		4/3/2001	QD		
5	SW	sc_datarate_matrix		4/3/2001	QD		
6	SW	sc_subsystem_mass_matrix		4/3/2001	QD		
7	SW	swarm_masses		4/3/2001	QD		
8	SW	software_cost		4/3/2001	QD		
9	ERR	swarm.no_sat_comm	0 or 1	4/3/2001		if swarm has no comm capability	

Input Variables — Check Input Variables

ID	Abv	Variable Name	Units	Date	Author	Definition	Aliases
1	D	sats_per_swarm	integer	4/3/2001	class	number of satellites per swarm	
2	D	sounding	[0,1,2,3]	4/3/2001	class	vector of length sats_per_swarm	
3	D	short_range_comm	[0,1,2,3]	4/3/2001	class	vector of length sats_per_swarm	
4	D	long_range_comm	[0,1,2,3]	4/3/2001	class	vector of length sats_per_swarm	
5	D	processing	[0,1]	4/3/2001	class	vector of length sats_per_swarm	none

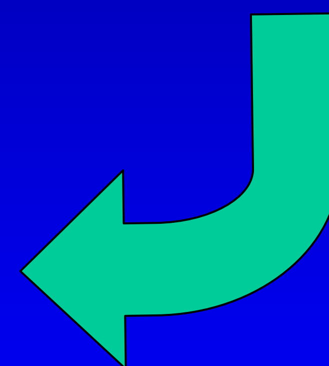
Structure List / Design / Constants / swarm / spacecraft / swarmrel / reliability / time / orbitprop / swarmorbits

Checked the interface variable consistency using an embedded macro

N^2 Diagram

		Design	Constants	Swarm	Swarmrel	Reliability	Orbit	Orbitprop	Launch	Operations	Costing	Time	Calculate_Attributes	Spacecraft	Utility Function	output_BTOS
	Module Name	**D**	**C**	**SW**	**SWR**	**R**	**O**	**ORP**	**L**	**OPS**	**Cost**	**T**	**A**	**SC**	**U**	**out**
D	Design															
C	Constants															
SW	Swarm	x	x													
SWR	Swarmrel	x	x	x												
R	Reliability	x	x		x											
O	Orbit	x	x													
ORP	Orbitprop	x	x	x												
L	Launch	x	x	x												
OPS	Operations	x	x	x												
Cost	Costing	x	x	x					x	x						
T	Time	x	x		x											
A	Calculate_Attributes	x	x				x					x				
SC	Spacecraft	x	x													
U	Utility Function	x	x										x			
out	output_BTOS	x	x	x		x	x		x		x		x		x	

Information Flow Direction



Valuable Lessons from N^2 Diagram

- N^2 diagram shows waterfall process
- Coding process is highly iterative
- N^2 diagram is good at capturing stable processes and improving it
  – N^2 diagram can be used to direct action for C-TOS if codes are similar and reduce design iterations
- Process of learning about the relationship between modules is highly iterative

Valuable Lessons from Integration

- Process showed that accurate and routinely updated I/O sheets were important
- Individual module verification can reduce integration workload
- Adding functionality (error trapping) at mid-point in code development was helpful but problematic
- Spring Break added difficulty to communication at a crucial time in process

Generalized Information Network Analysis (GINA)

Description

A process to model space systems as an information network.

Application in B-TOS

Assisted generating code structure as per A-TOS code. Identified major module areas.

Strengths

- Streamlines modeling process by identifying major code components
- Provides framework for comparing thousands (or more) of architectures using common metrics

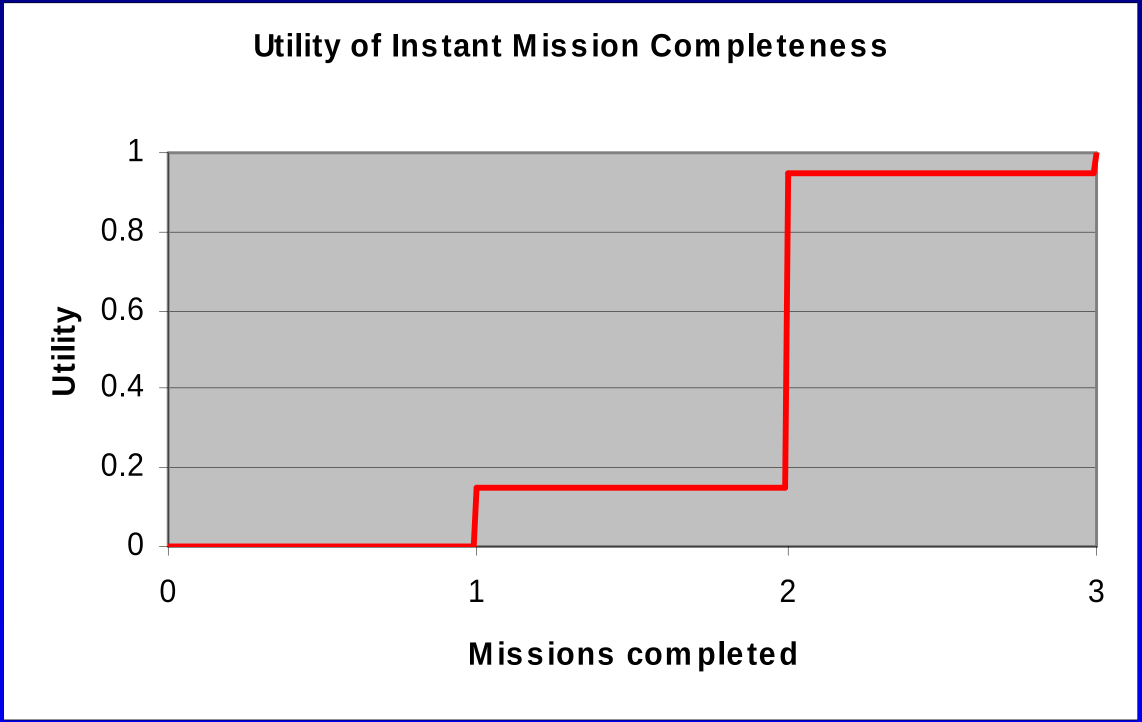
Limitations

- Strict GINA process has information metrics that may not relate to customer preferences
- Difficulty thinking in terms of information flow

Evolution of B-TOS Utility Attributes

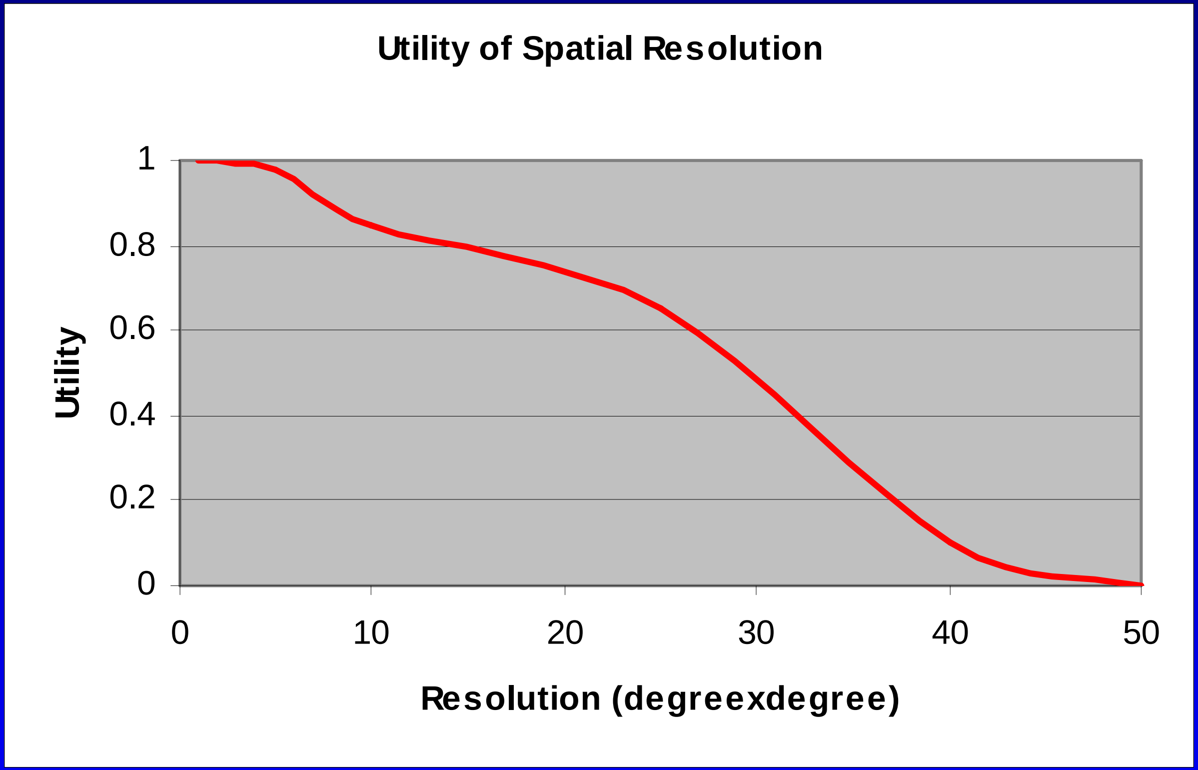
- *Time resolution* changed to *revisit time* during utility interview
- *Accuracy* defined by two attributes with different metrics and relative importance
  - Electron Density Profile (EDP) accuracy
  - Beacon Angle of Arrival (AOA) accuracy
- Discussions with customer to understand candidate attributes and resolve misunderstandings
- Understand tasking issues for mission completeness

Mission Completeness



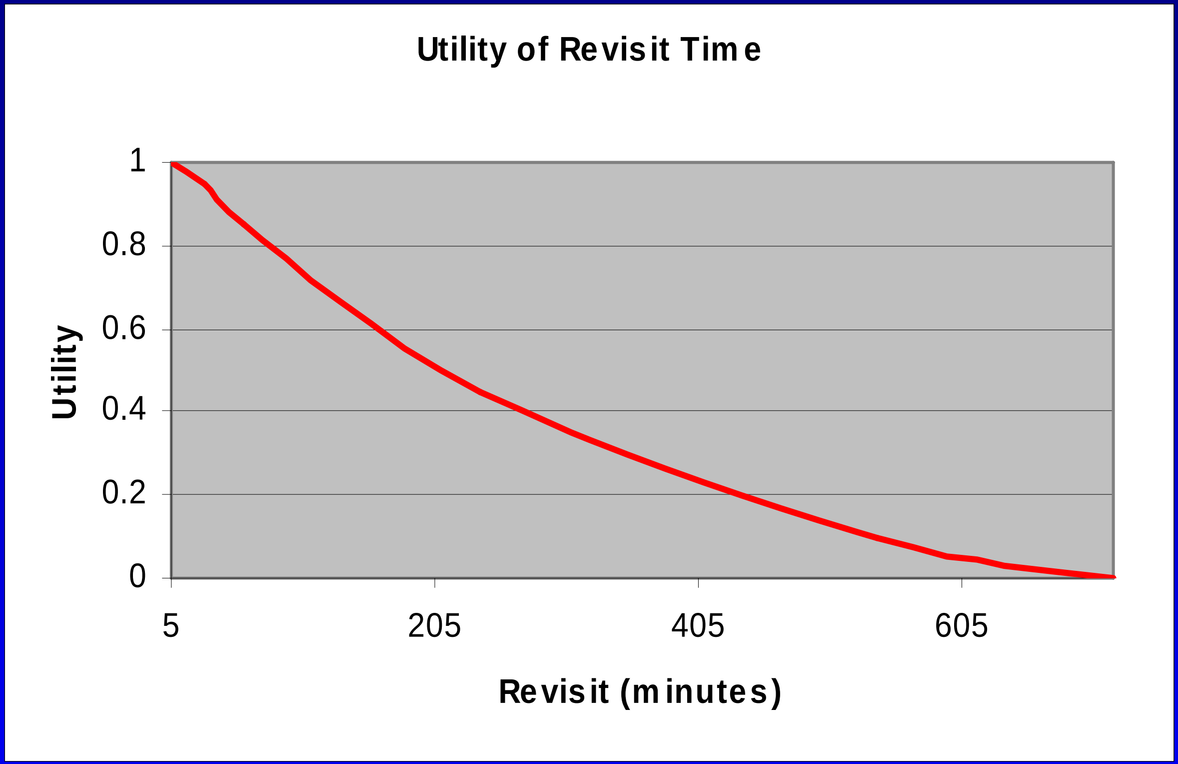
- Mission Completeness is a step function representing the combinations of measurement missions performed.
- 0,1,2,3: EDP, EDP/Turb, EDP/AOA, EDP/AOA/Turb

Spatial Resolution



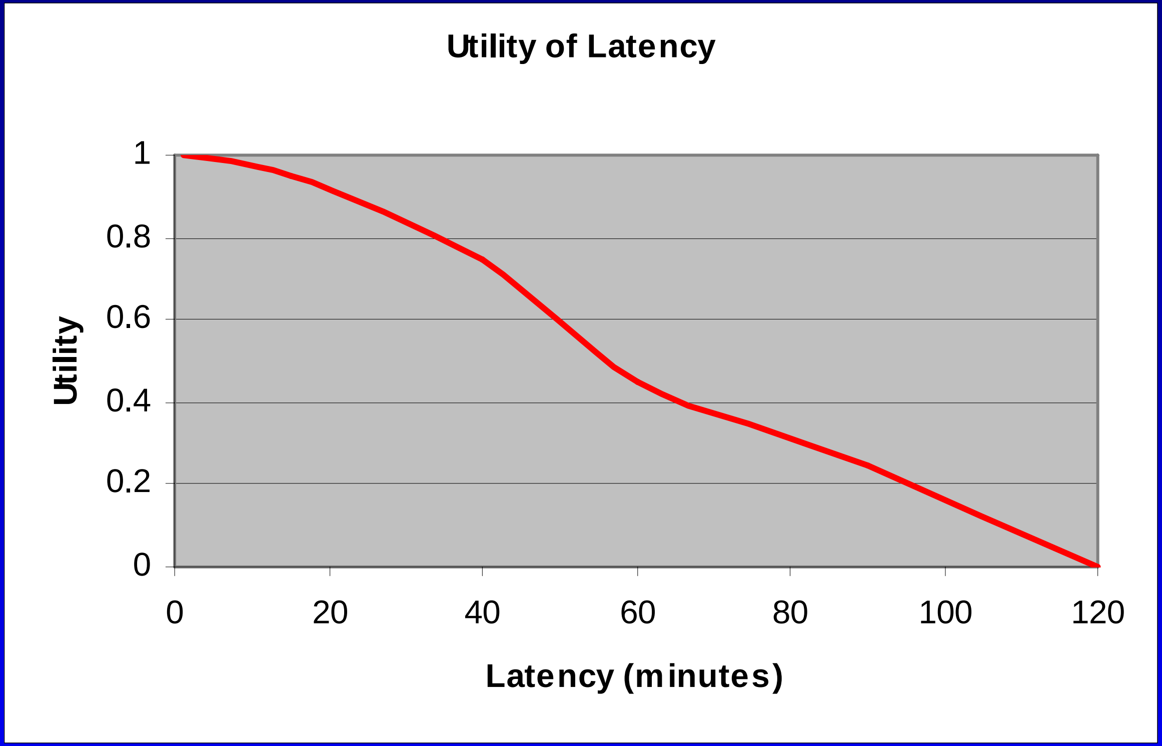
- The spatial resolution is the size of a measurement pixel (as determined by the time between data sets).

Revisit Time



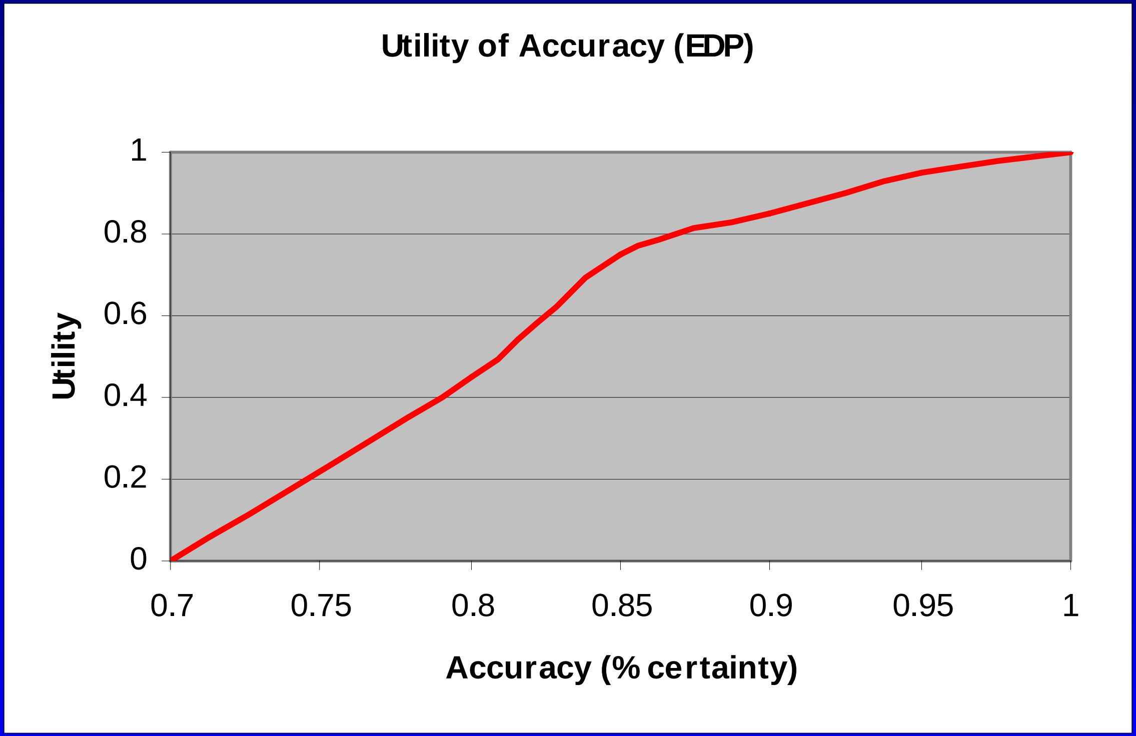
- Revisit time is the time elapsed between measurement sets at the same point. The points are represented by a grid based upon spatial resolution which is projected upon the Earth.

Latency



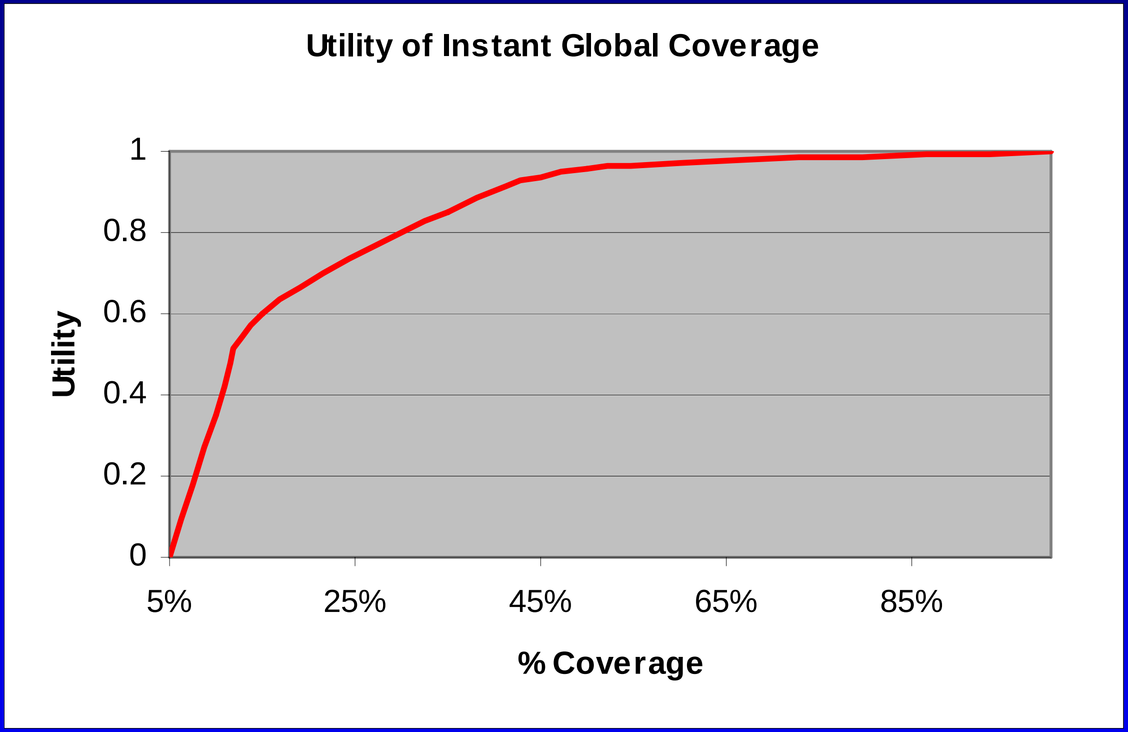
- Latency is the time elapsed between data collection and reception of processed data by the user.

Accuracy (EDP)



- EDP accuracy represents the size of the error bars on the EDP measurement.

Instantaneous Global Coverage



- Instantaneous global coverage is the percentage of the globe over which measurements are taken within the time resolution of the system.

Design Vector Evolution

- ✓ Method for developing design vector employed
- ✓ Eliminated binary mothership design trade but maintained concept through selectable satellite functionality
- ✓ Design vector rigid enough to define unique architectures through model development, yet flexible enough to allow honing:
  - – With weighting of attribute importance
  - – Range of attributes

Launch

Description

•Selects the lowest cost launch vehicle that can deploy all of the satellites for a single swarm.

•Once a launch vehicle is selected, total cost for initial deployment is computed.

Key Assumptions

•Launch vehicle and cost is function of: number of satellites/swarm, stowed dimensions of satellite, orbital altitude, launch vehicle mass capacity, and launch vehicle payload dimensions

•Assumes 100% launch success rate

Fidelity Assessment

•First iteration makes use of average satellite mass

•Considers 6 different launch vehicle possibilities and 14 altitudes

Verification

•Tested over range of satellite numbers, satellite masses and swarm sizes

•Fully integrated into B-TOS master design code

Orbit

Description

•Propagates orbital trajectories from initial conditions using Satellite Tool Kit (STK)

•Calculates coverage and revisit time statistics

•Determines satellite distribution within the swarm

Key Assumptions

•Orbit maintenance assumed; used two-body propagation over one day

•Walker constellation of swarms

•One sub-plane per swarm and log spacing between sub-orbits

•Horizontal circular projection of swarm

•Effective FOV of swarm based on spatial resolution

Fidelity Assessment

•STK used to ensure high fidelity of orbit trajectories at the expense of developing a MATLAB-STK interface

Verification

•Visual inspection of swarm geometry in 3-D

•Examined extreme cases for errors

Swarm

Description

•This function, by calling on the spacecraft module, outputs vectors defining the following parameters for the entire swarm:

•Mass, cost, reliability, dimensions

Assumptions

•Assumes that the every possible configuration of swarms can be built

•Again it assumes that spacecraft calculations are correct

Fidelity

•This module's depends almost entirely on the accuracy of the spacecraft module

•One variable somewhat independent of spacecraft that must be improved upon is the software cost as a result of a swarm configuration

Verification

•Code was fairly simple to test, creating a list of output variables from spacecraft one can examine each of the swarms outputted matrices.

Time

Description

• Check which missions the system can do taking into account degradation of the system over time: calculate the new mission_to_task

• Calculate the minimum number of receivers necessary to fill the swarm for ambiguity and check if the architecture tested eliminate ambiguity

• Calculate time delays for latency and time resolution at three different times

Assumptions

• Algorithm to calculate the minimum number of receivers needed to eliminate ambiguity. We don’t take into account radial baselines for ambiguity

• No processing delay

• Time resolution is based on time of measurement

• Divide the frequencies over all sounders

Fidelity

• New mission to task takes into account degradation, functionnalities needed to complete the mission and minimum altitude for edp measurement

• Algorithm for ambiguity has been improved

• Time resolution can be improved with data on processing delays and autonomy

Verification

• New_mission_to_task has been tested with a case study (various combinations of functionnalities, degraded satellites and initial mission_to_task)

• Ambiguity calculation has been tested using a spreadsheet to see the effect of different swarm radii and instrument phase errors (study of the effect on accuracy and number of suborbits)

Operations

Description

•Calculates operations personnel and facilities

- •Workload calculations account for complexity/reliability of spacecraft

•Calculates recurring and non-recurring operations costs

Key Assumptions

•Uses 7 different types of personnel

•Costs account for new facility construction

•Ops personnel capability adjusted with learning curves

Fidelity Assessment

•Impact of swarm autonomy, or lack thereof, not included

•TDRSS access costs were guess ($500k/beam)

•Ground software development not included

Verification

•Code closely derived from a previously used operations module

Reliability

Description

•Determines the probability that a particular number of satellites are operational in any swarm at a given time

Key Assumptions

•Mean time to failure for each satellite type is properly specified

•'Rounding' of number of operational satellites is done to nearest whole number

•Markov model is appropriate

Fidelity Assessment

•Able to accept mean time to failure for each different satellite type

•Computes steady state reliability matrix for any specified time during the mission

Verification

•Module tested over wide range of mean time to failure for different satellite types

•Able to accept any number of satellite types and give system state for beginning, middle and end of 5 year mission

Attributes

Description

• Calculates the value of the 6 attributes at three different times (BOF, mid-mission, EOF) for utility function

• Calculation takes reliability into account

• Error flags indicate if the attributes are out of range

Key Assumptions

• EDP accuracy: based on time resolution

• Beacon accuracy: determined with an interferometric relation based on the maximum baseline

• Latency: based on communication delay (calculated with an estimation of the data rates), Processing delay set to 0

Fidelity Assessment

• Coverage and revisit time calculated by STK

•Mission completeness considers the number of satellites down

• Latency can be improved by taking processing delay and autonomy into account

• Main issue: accuracy (EDP and beacon accuracy) has to be modified

Verification

• Tested with a sample module simulating the inputs of other modules

• Tested with runs (fixed problems of units compatibility and attribute ranges)

• Check for consistency with outputs of other modules and inputs needed by utility

Utility function

Description

•Module captures the relative "value" tradeoffs of the customer for various combined sets of attributes (metrics) of the architecture

Assumptions

•Utility independence

•Preferential independence

•Customer can perceive gradation of value for different levels of attributes

Fidelity

•Validation interview matched model fairly well, especially for showing preference. Absolute level of utility may not "be right", but utility is on a relative scale anyway, so this problem is minimal.

Verification

•Held a validation interview with customer and checked output.

•Verified code with interview responses.

•Checked out of bounds errors.

Costing

Description

•Includes spacecraft, operations, launch, and program level costs

•Uses CER for spacecraft/program level costs (including error bars)

•Incorporates learning curve for different spacecraft types

Key Assumptions

•Cost model assumes small satellites (20-400 kg)

•No replenished satellites

Fidelity Assessment

•Error bars are ~20-40% of spacecraft costs

•Error increases with decreasing satellite mass and increased learning curve affect

Verification

•Spacecraft and program level costs were checked by hand calculation

Multi-Attribute Utility Function

$$KU(\underline{X})+1=\prod_{i=1}^{6}\left(Kk_iU(X_i)+1\right)$$

Normalization constant

Multi-attribute utility function

Relative "weight"

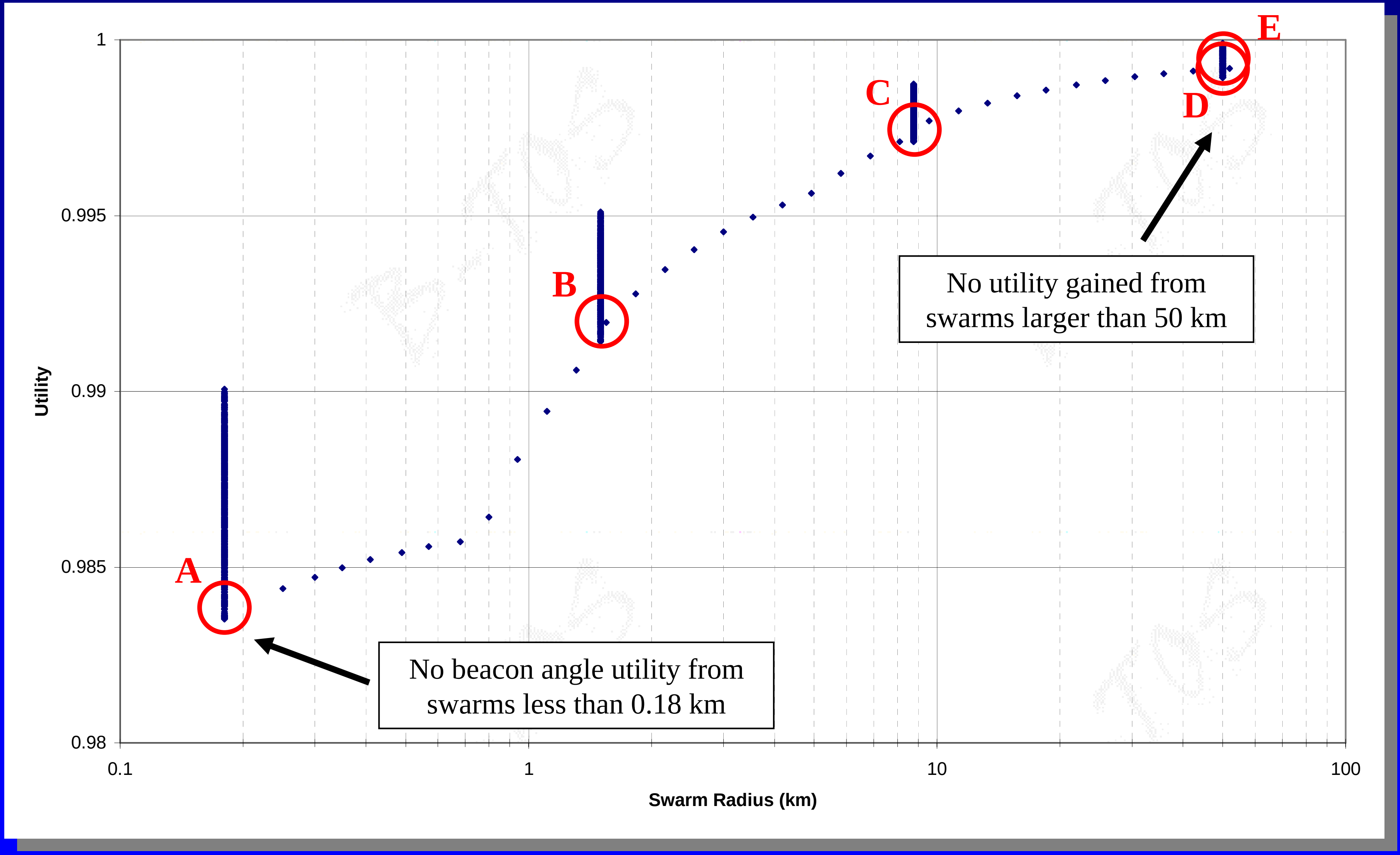
Single attribute utility

Spacecraft Characteristics

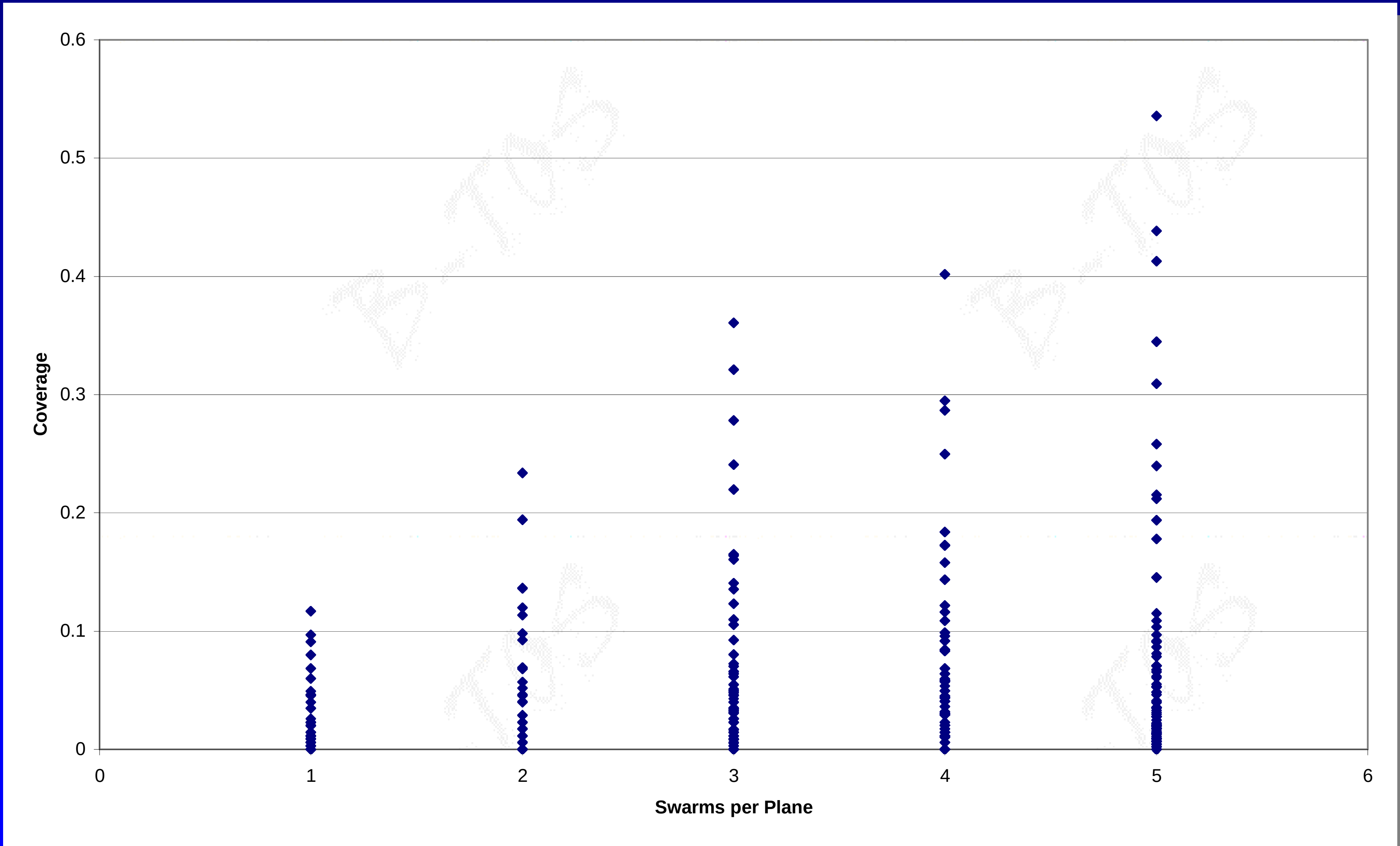
	Mothership	Daughter
Spacecraft mass (kg)	165	72
Subsystem mass breakdown:		
ADACS:	8	8
CDH:	10	4
Payload:	32	17
Power:	33	13
Propulsion:	11	11
Structures:	33	14
Telecom:	29	1
Thermal:	8	4
Downlink data rate (bps)	30000	15000
Average power required (W)	191	73

All 5 frontier architectures have similar spacecraft

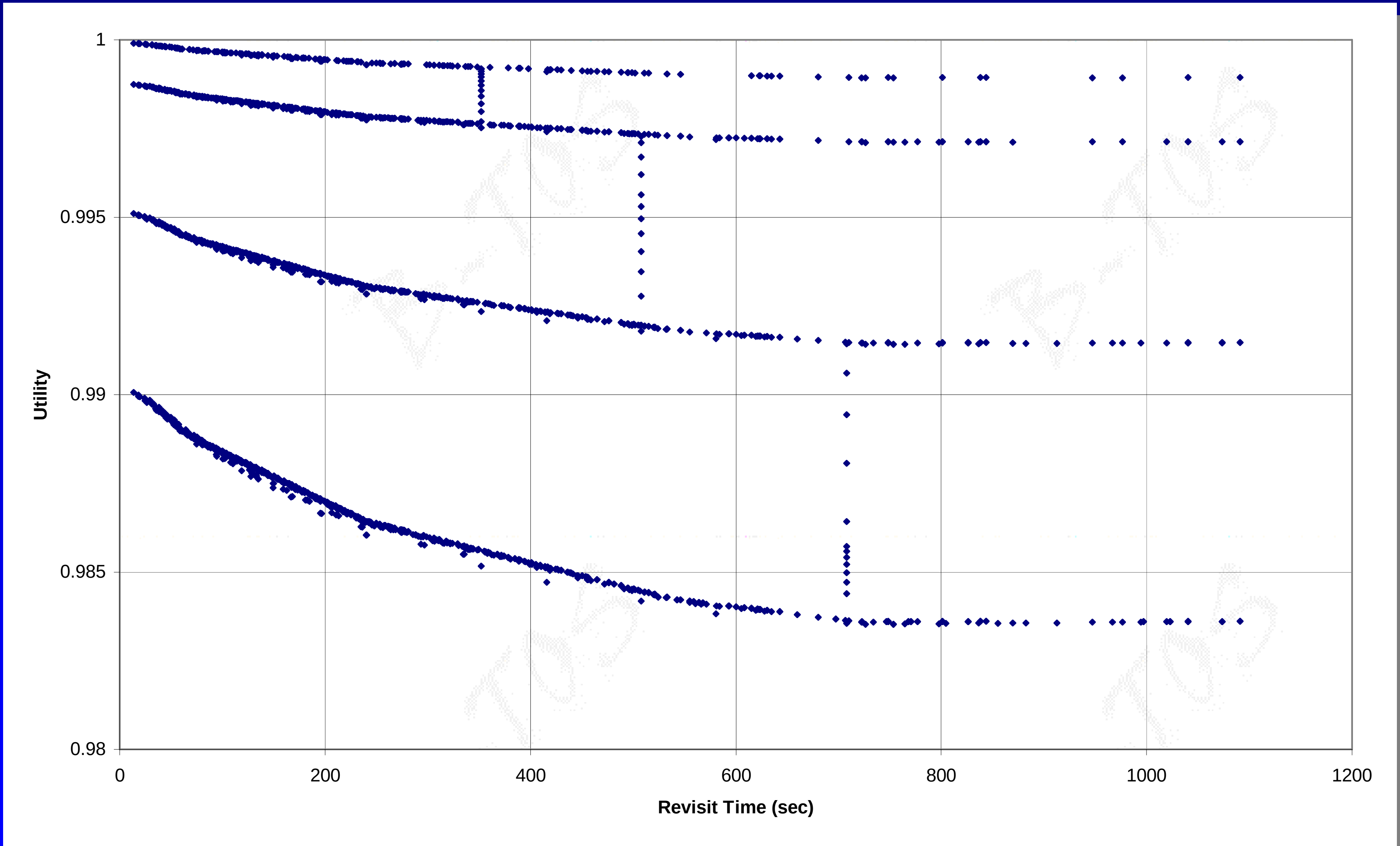
Swarm Radius vs. Utility



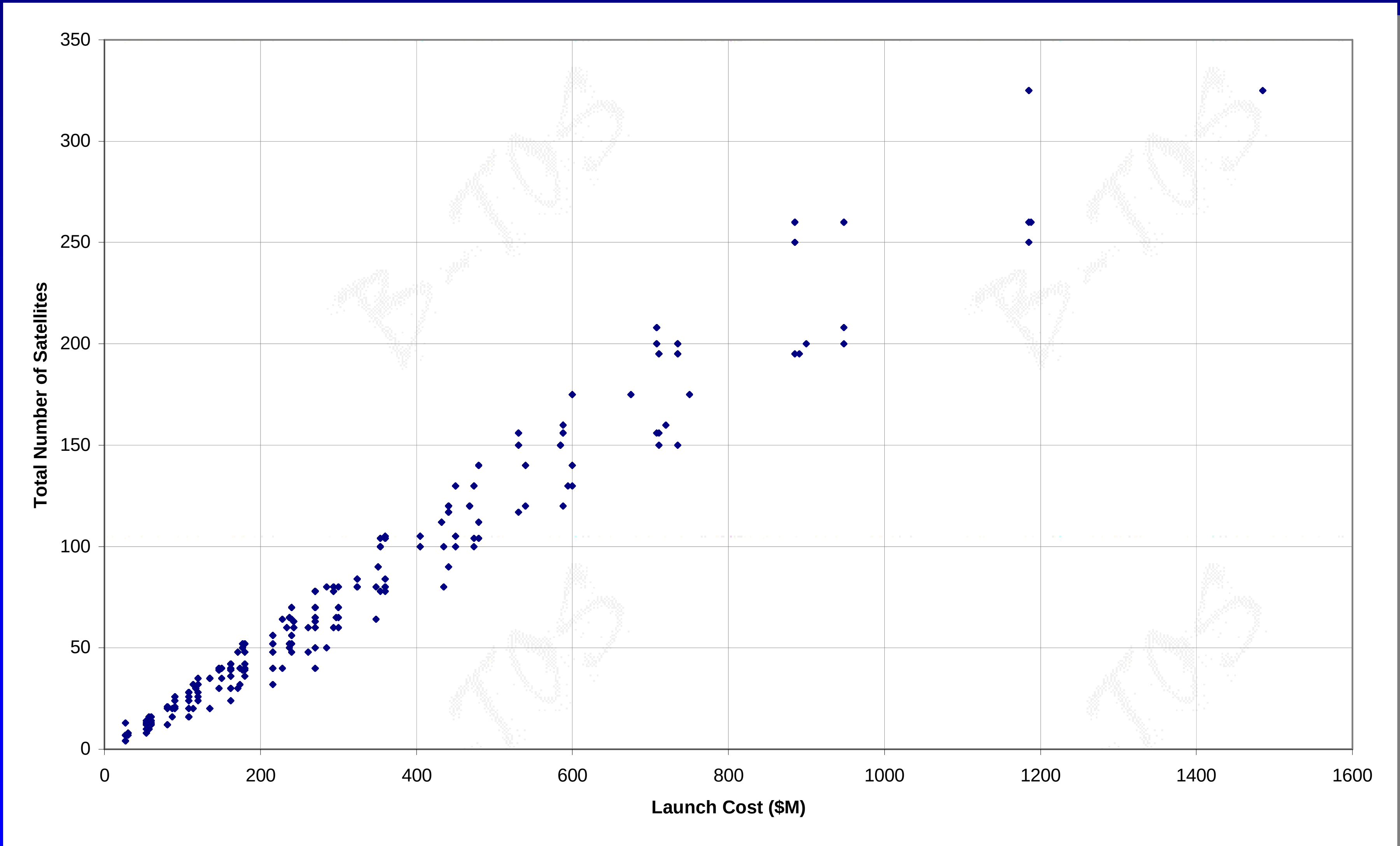
Swarms per Plane vs. Coverage



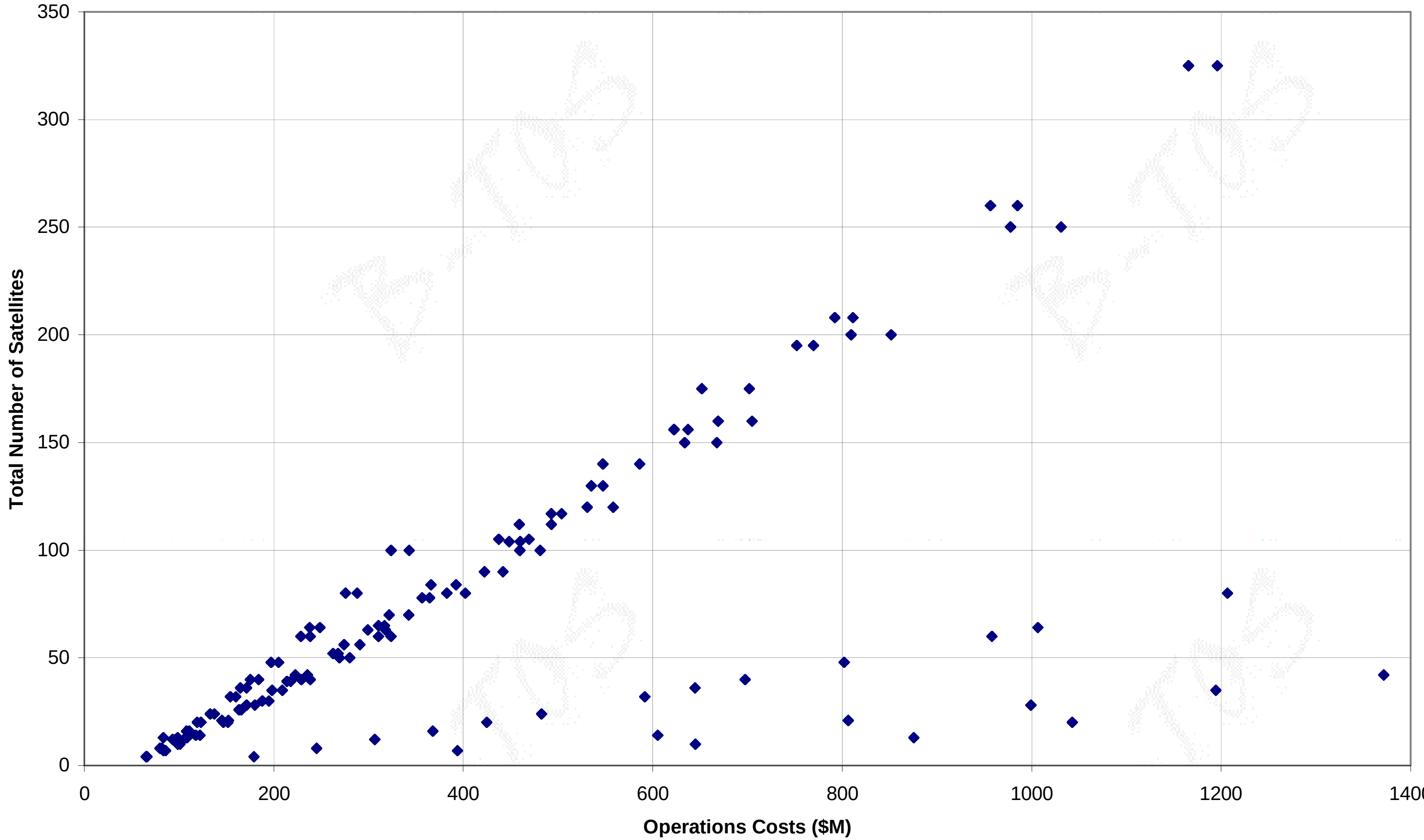
Revisit Time vs. Utility



Launch Cost Trend



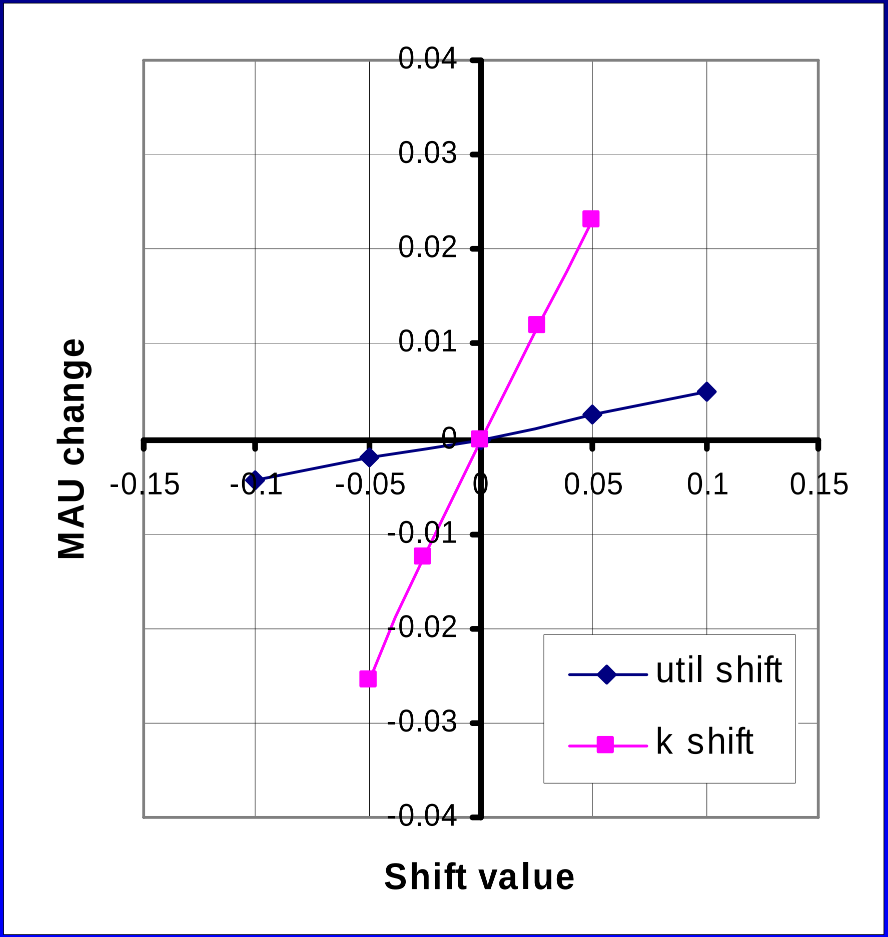
Operations Cost Trend



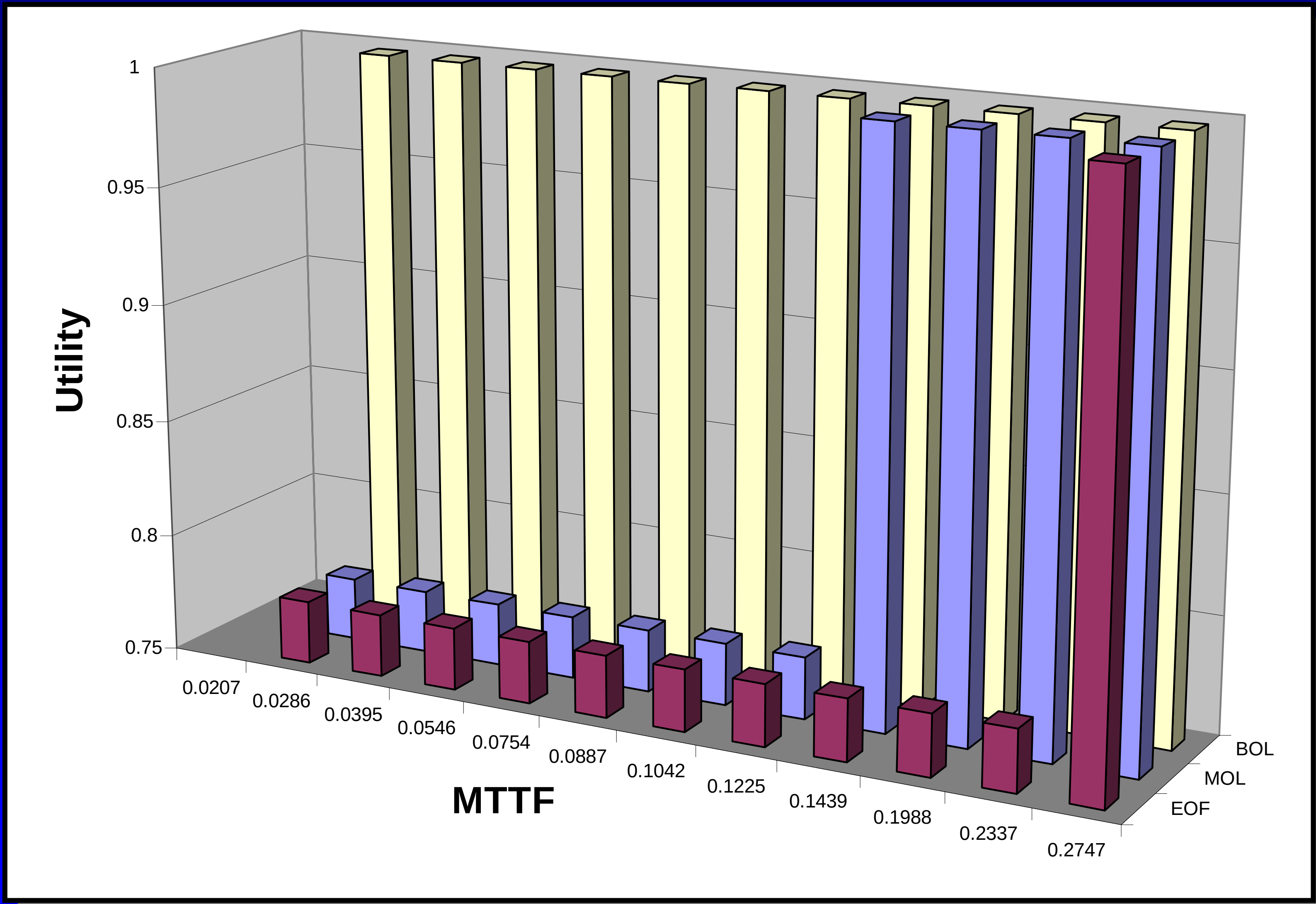
Utility Function Sensitivity

- Uncertainties in the relative weight of the attributes in the utility
- For a 10% change in ??? we get a change in utility of 0.005

Utility sensitivity is low enough to validate architecture results



Reliability Sensitivity



Integrated Concurrent Engineering (ICE)

- Simplified ICE design method used in B-TOS
- ICE is real-time concurrent design that eliminates communication/information bottlenecks and increases productivity
- Subsystem groups use design tools to model subsystems and information is shared via central database with other groups in real-time
- Data flow between groups is streamlined using an N^2 diagram analysis
- Facilitator guides teams through design iterations and works out design issues with all groups present in a design room

Facility Requirements & Characteristics

- Enough room for the team
- Work stations for each team member
- Work stations arranged around periphery to enhance communications
- LAN connections between stations
- A projection system that can monitor any station
  - Multiple projectors are preferable
- The brand new Aero/Astro 33-218 design room is setup exactly like this and was designed with ICE in mind

Benefits of ICE

- It is a definable, repeatable, measurable process
- Design iteration is managed, not chaotic
- Data entry is distributed around the room eliminating bottlenecks
- Team members can link their tools directly through the system eliminating the need for excessive data re-entry
- It can be applied predictably to many different processes
  - Requirements definition, cost estimation, proposal preparation, schedule planning, system design, IRAD investment planning and more

Lessons Learned

- Careful application of past experience:
  - A-TOS modules: late realizations of necessary changes
- Need to consider if and how changes affect all other sections of the code
- Divided modules before all equations or requirements were known
- Appropriate architecture selection: limited by model fidelity and customer-provided utility function