



Calhoun: The NPS Institutional Archive
DSpace Repository

Faculty and Researchers

Faculty and Researchers' Publications

2011

What metrics are there to measure the cost effectiveness of standards implementation?

Baer, Wolfgang

Monterey, California. Naval Postgraduate School

<http://hdl.handle.net/10945/48010>

This publication is a work of the U.S. Government as defined in Title 17, United States Code, Section 101. Copyright protection is not available for this work in the United States.

Downloaded from NPS Archive: Calhoun



Calhoun is the Naval Postgraduate School's public access digital repository for research materials and institutional publications created by the NPS community. Calhoun is named for Professor of Mathematics Guy K. Calhoun, NPS's first appointed -- and published -- scholarly author.

Dudley Knox Library / Naval Postgraduate School
411 Dyer Road / 1 University Circle
Monterey, California USA 93943

<http://www.nps.edu/library>

Effectiveness Metrics and Cost Benefit Analysis Methodology for Machine-to-Machine Interoperability Standards

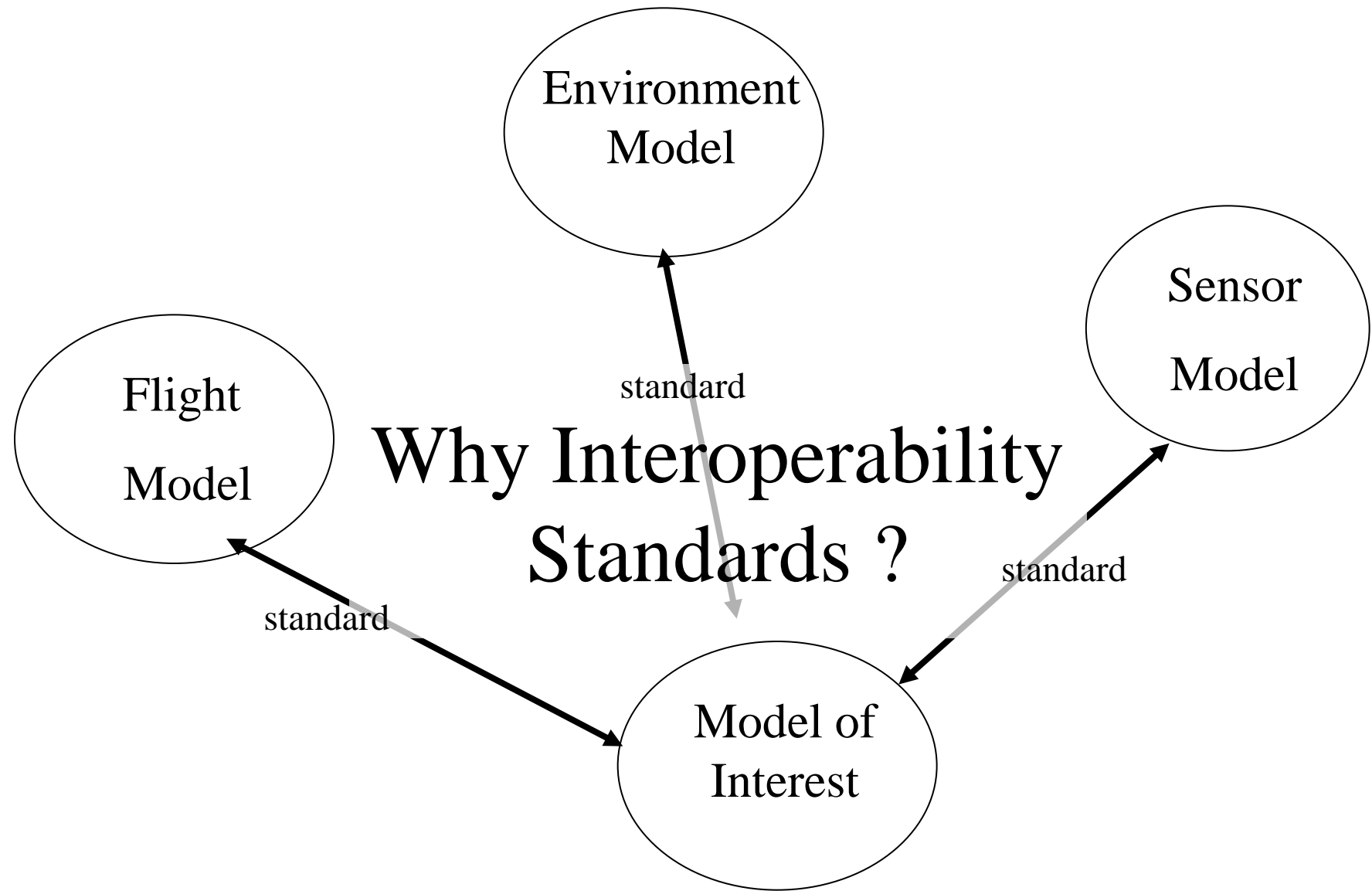
By

Wolfgang Baer

Associate Research Prof.

NPS





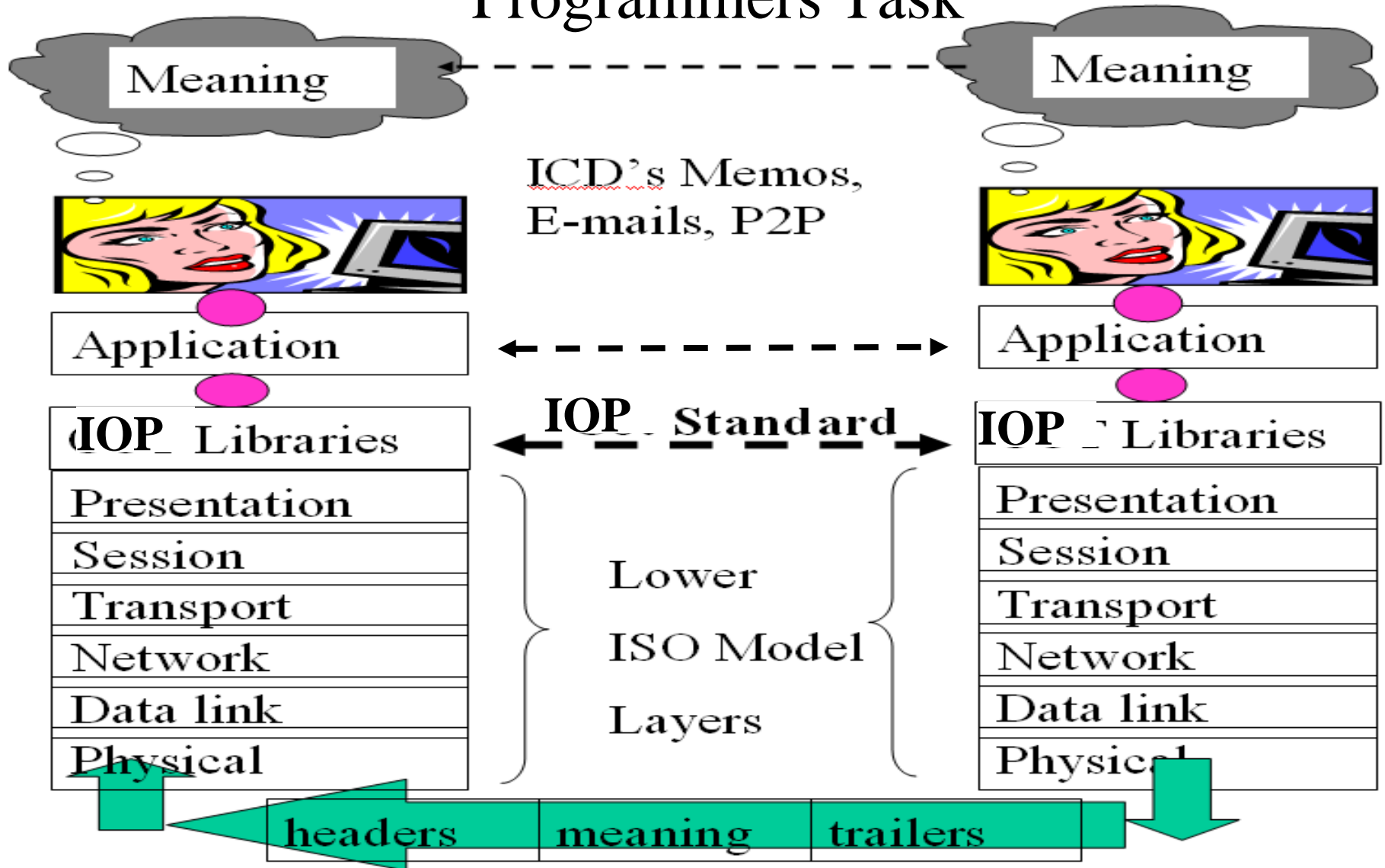
Interchangeable parts saves money

What Metrics are there to measure the Cost Effectiveness of standards Implementation?

- Not much in the literature
- Personal experience with SISO, SEDRIS, HLA, DIS, CoT, DTED, GPS, etc. etc.
- Some are good and save time and money
- Some are real dogs

How to analyze interoperability?

Step1. Reduce Interoperability to a Programmers Task



Step2: Analyze Interoperability Architecture

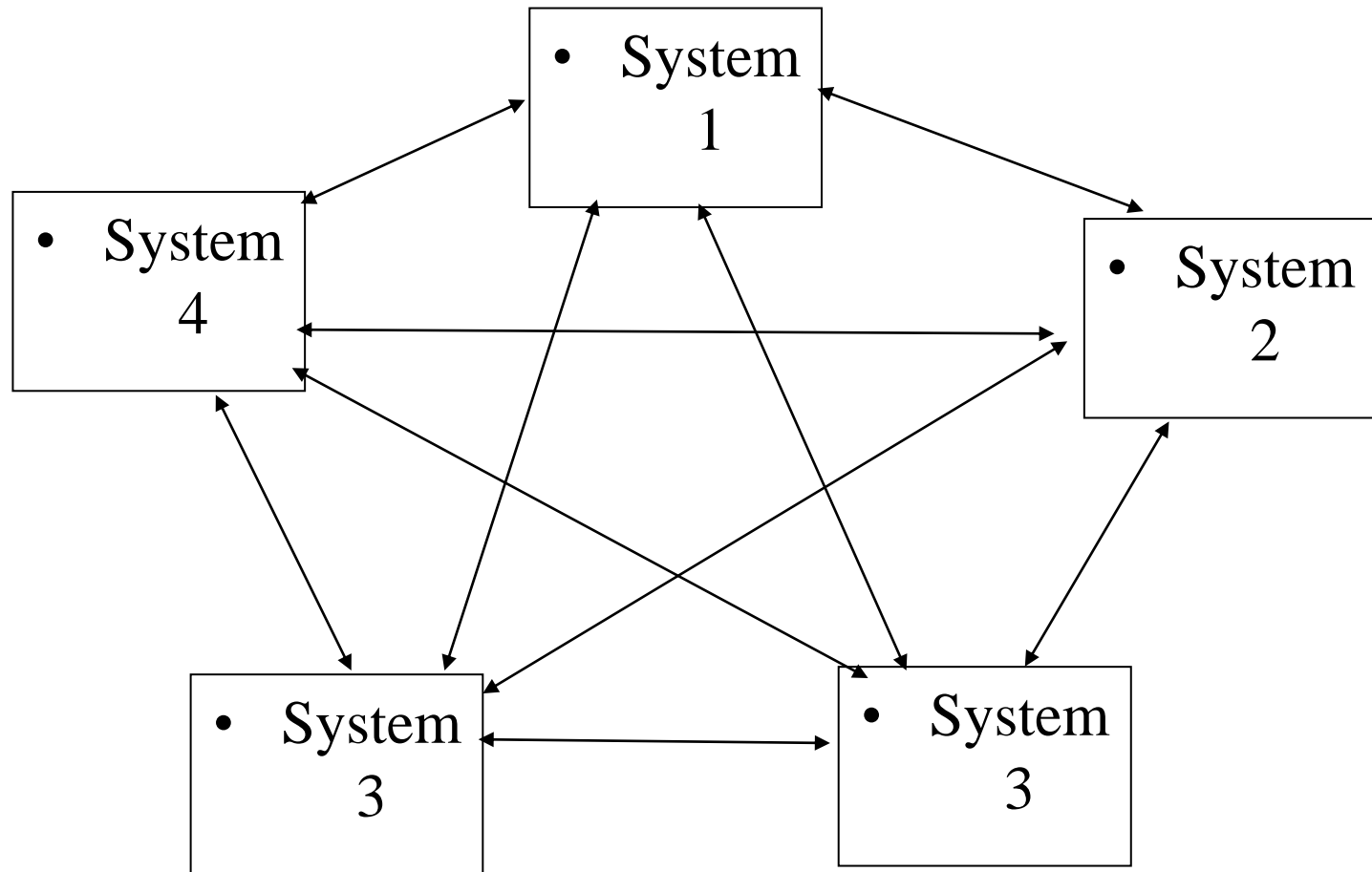


Fig. 2. – Five Interoperable Systems without standards
Requires $(N-1)N$ interoperability tasks ($N=5$)

Step3: Reduce The interoperability Tasks by Introducing Standards

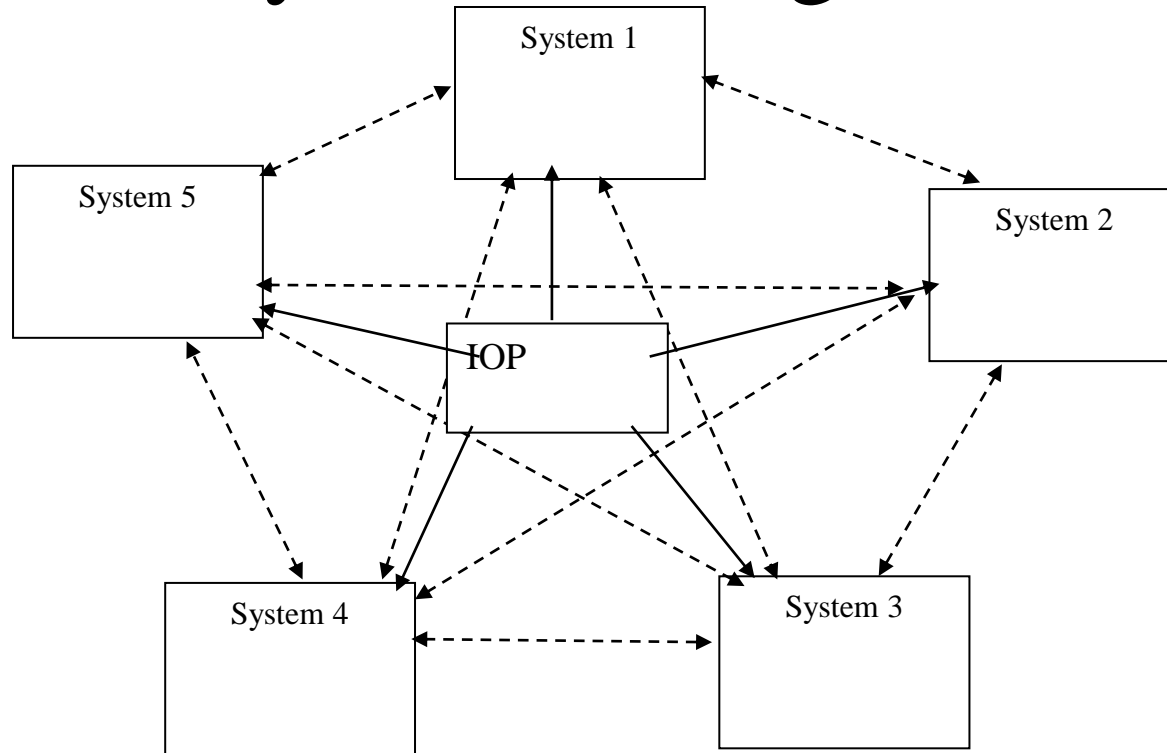


Fig 3 – Five Interoperable Systems types with standards

Reduces the number of tasks from
 $N*(N-1)$ to N

Step 4: Calculate the value of introducing a standard

- (ISPV) Interoperability Standards Project Value
- (IoPT) Interoperability programming task
- (SER) Standards Effectiveness ratio
- (SCR) Standards Complexity ratio

Value of the Standard $\underline{=}$ Cost with no standard minus Cost with a standard

$$\text{ISPV} = N(N-1)*\text{IoPT} - \{ N*\text{SCR} + N*(N-1)*(1-\text{SER}) \}*\text{IoPT}.$$

Step5: Introduce a comparative metric

Interface Standards Project Effectiveness (ISPE)

- (ISPVi) Value of the ideal standard with
 $SER=SCR=1$
- $ISPVi = \{N(N-1) - N\} * IoPTs$
- (ISPE) Standards Project Effectiveness
- $ISPE = ISPV / ISPVi$
- Which reduces to the basic metric equation:

$$ISPE = [(N-1)*SER - SCR] / (N-2)$$

Cost Reduction Assumption

A standard is ideal under two assumptions. These are:

- Standards Complexity Ratio $SCR=1$; i.e. the task cost of executing the interoperability task to the standard is equal to the cost of the point to point interoperability task.
- Standards Effectiveness Ratio $SER=1$ if there are no latent point to point tasks required in addition to the interface to the standard..

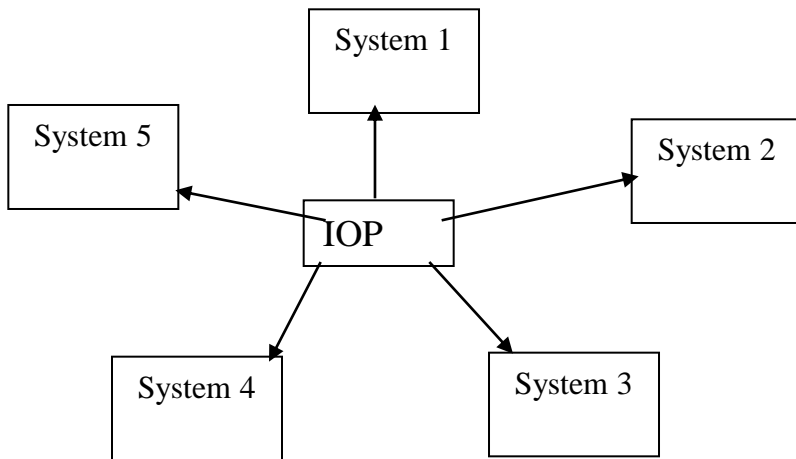


Fig 3 – Five Interoperable Systems types with standards for an ideal standard

Interoperability Standards Project
Effectiveness (ISPE) ratio

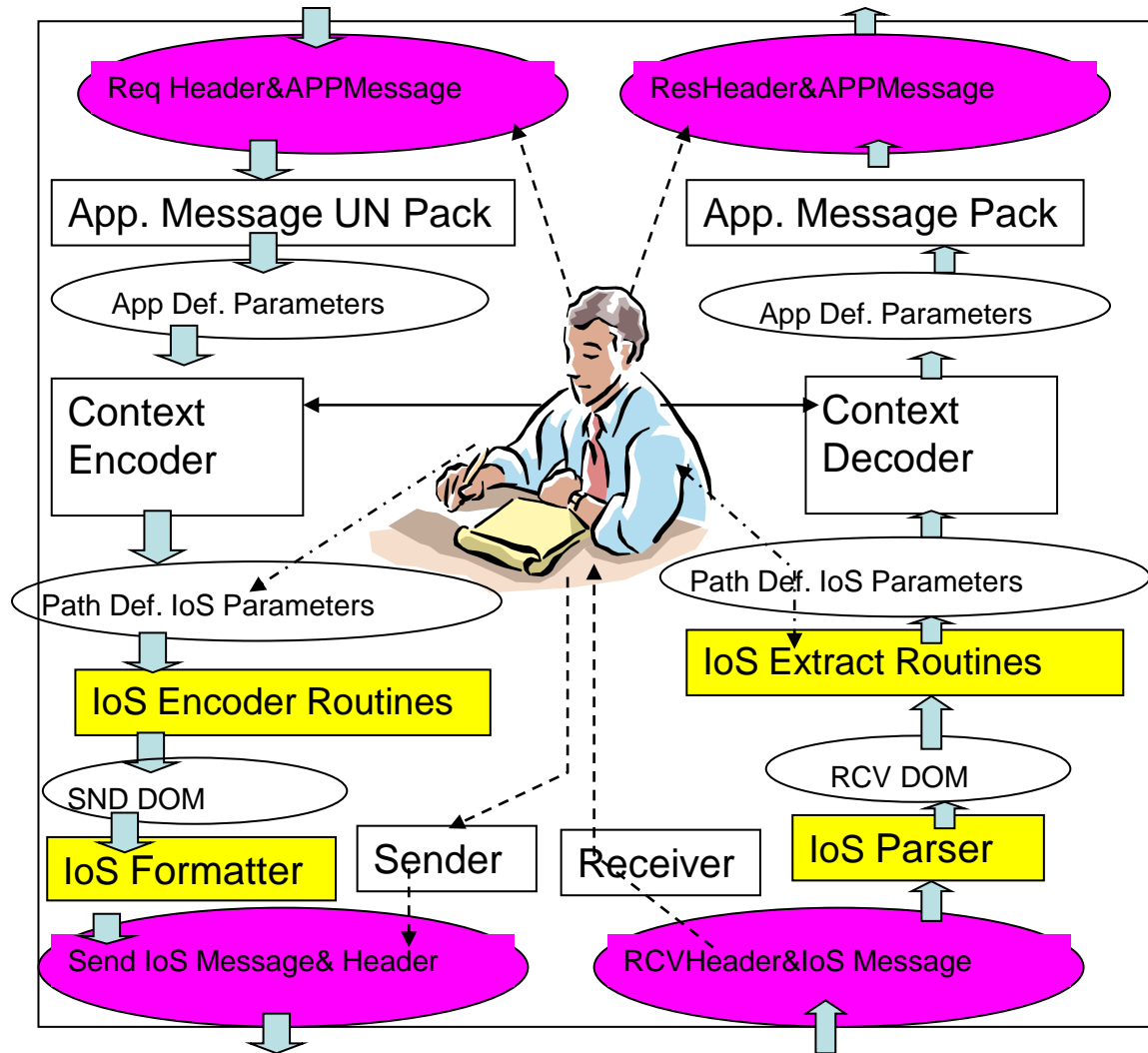
$$ISPE = [(N-1)*1 - 1]/(N-2) = 1$$

In this case we have an
ideal standard.

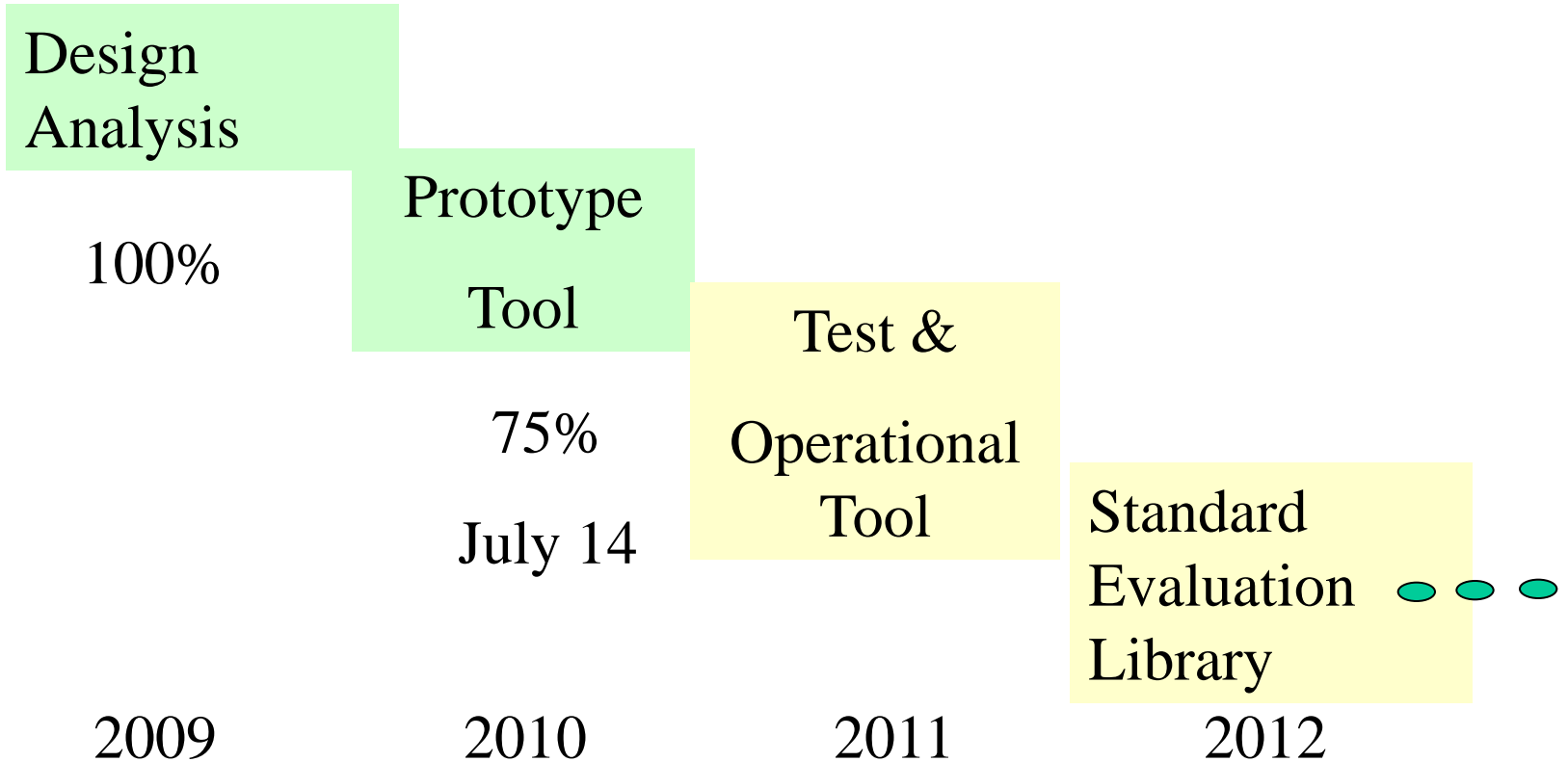
Step 6: To Evaluate Real Standards

- Estimate the cost of the interoperability Programmers task (IoPTs)
- Estimate the standards Effectiveness ratio (SER)
- Estimate the standards Complexity ratio (SCR)
- And do so as a function of standards characteristics

Define the Interoperability Programmers Task



Work Plan



Prototype Tool

Main Control Dialog

Standards Cost Effectiveness

OK Cancel

Name Language

Project Who When Where VC++ Save Proj. Get Proj.

Language Standard Application

Programmer Qualifications 40 averag 4 average i 4 familiar wit

App. Msg. Unpack App. Msg. Pack

Standard Point2Point Save As Ref Save Std. Get Std.

Context Encoder Context Decoder

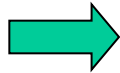
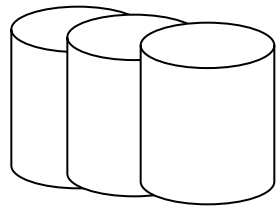
IoS Load Routines IoS Extract Routines

IoS Formatter IoS Parser

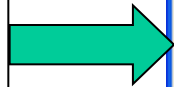
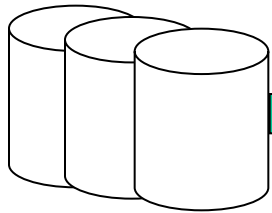
Sender Receiver

	Hrs.	\$	# of Nodes	learn 1/2 life
Calculate Standard Interface	0.0	0.0	1	1.0
Calculate Reference Interface	0.0	0.0	1	1.0

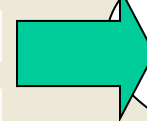
Comparative
Implementation
Costs



Project Files



Standards
Characteristics
Files



Prototype Tool

Standards Questionnaire Dialogs

UnPack Application Parameters

Cancel Average OK

of Items KC hrs. WC loc

Simple Values 0 .1 k 3 assign

Complex Values 0 .1 k 3 assign

Enumerated List 0 .1 k 3 assign

Formatted 0 .1 k 3 assign

Unformatted 0 .1 k 3 assign

Depth 0 .1 k 3 assign

Sub Field 1 0 .1 k 3 assign

Cost in Hours 0.0

Application Input: App. Pack

Cancel Average OK

of Items KC hrs. WC loc

Simple Values 0 .1 kn 3 assign

Complex Values 0 .1 kn 3 assign

Enumerated List 0 .1 kn 3 assign

Formatted Field 0 .1 kn 3 assign

Unformatted Field 0 .1 kn 3 assign

Depth 0 .1 kn 3 assign

Sub Field 1 0 .1 kn 3 assign

Cost in Hrs. 0.0

Standards Cost Effectiveness

DK Cancel

Project Who When Where V/C++ Save Proj. Get Proj.

Language Standard Application

Programmer Qualifications 40 average 4 average i 4 familiar wit

App. Msg. Unpack App. Msg. Pack

Standard Point2Point Save As Ref Save Std. Get Std.

Context Encoder Context Decoder

IoS Load Routines IoS Extract Routines

IoS Formatter IoS Parser

Sender Receiver

Calculate Standard Interface Hrs. \$ # of Nodes learn 1/2 life

Calculate Reference Interface 0.0 0.0 1 1.0

0.0 0.0 1 1.0

Application Input: App. Pack

Cancel Average OK

of Items KC hrs. WC loc

Simple Values 0 .1 kn 3 assign

Complex Values 0 .1 kn 3 assign

Enumerated List 0 .1 kn 3 assign

Formatted Field 0 .1 kn 3 assign

Unformatted Field 0 .1 kn 3 assign

Depth 0 .1 kn 3 assign

Sub Field 1 0 .1 kn 3 assign

Cost in Hrs. 0.0

Standards Code Acquisition Dialog

Cancel Parser Save

Executable Code	Cost \$	Applicable %	Availability hrs	Reliability
Executable Code	0.00	100.0	0.0L Bui	10% Bugg
Executable Code Doc	0.00	100.0	5.0L Bui	100% Exc
Source Code	0.00	100.0	10.0L Bu	100% Exc
Source Code Documentation	0.00	100.0	5.0L Bui	100% Exc
test case	0.00	100.0	10.0 Bui	100% Exce

Number of Standard Items 10 Number of Items Needed in Application 4

Number of Items Parsed 10 Standards Message Complexity 7 Conditional He

Calculate Hrs. 0.0 \$ 0.00000

Future Tasks

- 2010 –
 - Finalize and document the Standards Cost Effectiveness (SCE) prototype
- 2011-
 - Work with VMASC on “Standards in Modeling and Simulation: Next Ten Years” project Hon. Randy Forbes
 - SISO presentations and papers advertise capability
 - Test and work with Standards Program offices (CoT, Ucore, DIS, ...)

Contact Information

- Prof. Wolfgang Baer Dep. of Information Science
Code IS, Naval Postgraduate School, 1 University
Circle, Monterey, CA 93943, Tel 831-656-2209

Baer@nps.edu

•

Acknowledgement

This work was partially supported by the Navy Modeling & Simulation Office (NMSO) through the Modeling, Virtual Environments and Simulation Institute (MOVES) at the Naval Postgraduate School, Monterey California.

Backup Slides

Example Task Estimation Formula

$$\begin{aligned} \text{IoPT} = & \text{KDOMpF} * \text{F} + \text{WPARS} + \text{WEXTpF} * \text{F} + \text{KSF2LpF} * \text{N} + \text{WSF2LpF} * \text{N} \\ & + \text{KENCpF} * \text{N} + \text{WFOR} + \text{WENCpF} * \text{N} + \text{KL2SFpF} * \text{N} + \\ & \text{WL2SFpF} * \text{N} \quad (18) \end{aligned}$$

Where:

F	Number of fields or individual data items handled in the standard
KDOMpF .1hrs	Knowledge to decode DOM location to IoS field name per field
WPARS 8hrs	Write Parser and message decoder
WEXTpF 1hr	Write extraction routine per field
KSF2LpF 1hr	Knowledge to understand IoS in terms of local parameter per field
WSF2LpF 2hr	Write IoS field to local parameter translator per field
KENCpF .1hr	Knowledge to encode IoS Filed names into DOM locations per field
WFOR 8hrs	Write a formatter that builds a IoS message buffer
WENCpF 1hr	Write encoder from IoS Filed names into DOM locations per field
KL2SFpF 1hr	Knowledge to understand local variables in Ios terms per field
WL2SFpF 2hr	Write local variable to IoS symbol translators per field



I got these values by coding a test interface in C++ and keeping track of my time per coding task I got 726hrs

SCR

Complexity estimation considerations

- **IoS Message Parser and Extract Routines**
- **Context Decoder Information Requirements**
- **Context Encoder Information Requirements**
- **Programmer qualifications**

SCR

Complexity Estimation

By substituting the values estimated in this example for the 100field case used in section 4 we estimate a task cost

$$\text{IoPTc} = .5 * F + 160 + 1 * F + 2 * F + 2 * F + .75 * F + 24 + 3 * F + 1 * F + 2 * F = 184 + 12.25 * F = 1409 \text{ man hours}$$

To be consistent we should add the same programmer qualification cost of 90hrs to get 1499hrs compared with 726hrs for the P2P case. The standard complexity ration is then

$$\text{SCR} = \text{IOPT}/\text{IOPTi} = 1499/726\text{hrs} = 2.06$$

Effectiveness Categories

- Incompleteness (M)
- Knowledge Ambiguities (O)
- Field Ambiguities (P)
- Undocumented Extensions (X)
- Subset extensions (SXS)

$$\text{SER} = \{F/(F+M)\} * \{F/(F+O)\} * \{F/(F+P)\} * \{F/(F+X+SXS)\} = 84.8\% \quad (16)$$

Example Test Case: F= for a 100 field standard, M=5, P=5, O=2 and all others zero.

Plugging into Standards Effectiveness Formulas for a 20 node 100 parameter interface

Cost Savings if we had an ideal standard

$$ISPVi = \{N(N-1) - N\} * IoPTs = (20*19-20)*726 = 13,068\text{hrs}$$

Effectiveness of our test example standard:

$$\begin{aligned} ISPE &= [(N-1)*SER - SCR]/(N-2) \\ &= [(20-1)*.84 - 2.06]/(20-2) = 14/18 = 77\% \end{aligned}$$

Cost savings if we introduce our test example standard:

$$ISPV = ISPE * ISPVi = .77 * 13,068\text{hr} = 10164\text{hrs}$$

Cost to the project because the example standard is has
deficiencies:

$$\mathbf{ISPV - ISPVi = 2904\text{hrs or 1.5 man-years}}$$