

Limits to the Use of the Zachman Framework in Developing and Evolving Architectures for Complex Systems of Systems

Philip Boxer, Suzanne Garcia

May 6th 2009



Software Engineering Institute | CarnegieMellon

© 2009 Carnegie Mellon University

Agenda

- The Enterprise/Architecture relationship
 - The demands of collaborative systems of systems
 - Limits to the use of the Zachman Framework & the consequences for DODAF 2.0
 - Summary



Software Engineering Institute | CarnegieMellon

Limits to the use of the Zachman Framework

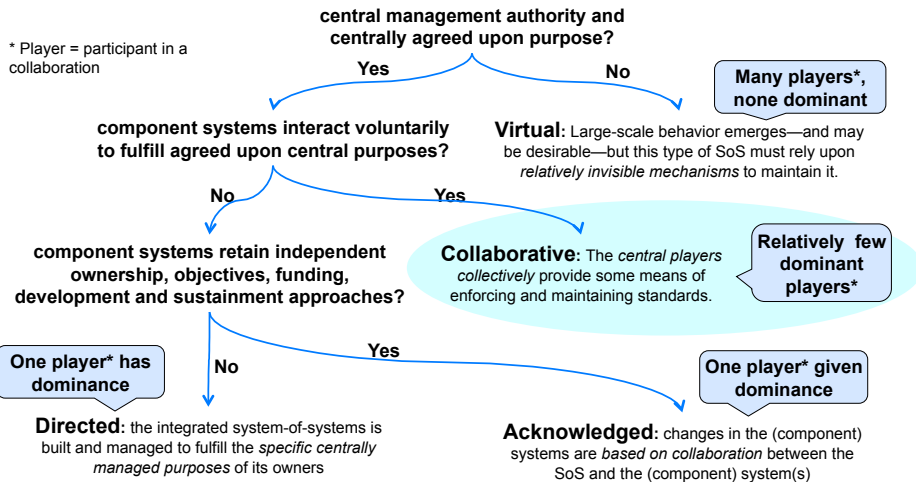
Philip Boxer & Suz Garcia, May 6th 2009

© 2009 Carnegie Mellon University

2



Systems of Systems: 4 Types defined by OSD SE Guide for Systems of Systems



Source of definitions: Systems Engineering Guide for Systems of Systems, OSD, Version 1.0 August 2008. Brackets added.



Software Engineering Institute

CarnegieMellon

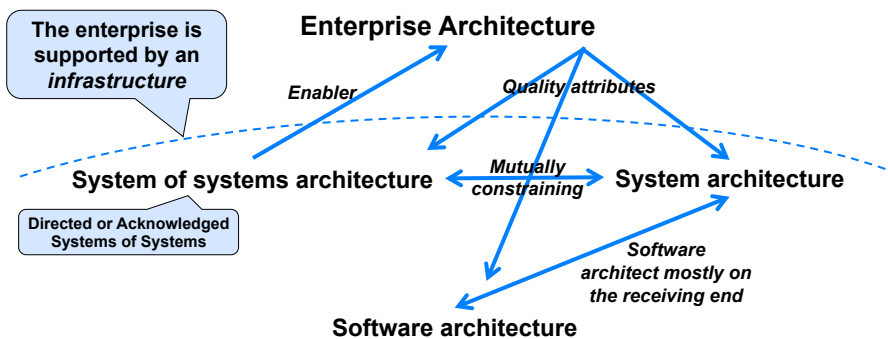
Limits to the use of the Zachman Framework

Philip Boxer & Suz Garcia, May 6th 2009
© 2009 Carnegie Mellon University

3

Architectural Genres: different genres for different purposes

The primary interfaces across genres as evidenced by working group discussions:



These genres reflect a *supply-side perspective* on the enterprise

Source: U.S. Army Workshop on Exploring Enterprise, System of Systems, System, and Software Architectures, John Bergey, Stephen Blanchette, Jr., Paul Clements, Mike Gagliardi, John Klein, Rob Wojcik, Bill Wood, March 2009 TECHNICAL REPORT CMU/SEI-2009-TR-008



Software Engineering Institute

CarnegieMellon

Limits to the use of the Zachman Framework

Philip Boxer & Suz Garcia, May 6th 2009
© 2009 Carnegie Mellon University

4



Software Engineering Institute

CarnegieMellon

© 2008 Carnegie Mellon University

2

The Enterprise Architecture defines the way it creates value: *Zachman roots to DODAF*

	DATA (WHAT) e.g. data	FUNCTION (HOW) e.g. function	NETWORK (WHERE) e.g. network	PEOPLE (WHO) e.g. organisation	TIME (WHEN) e.g. schedule	MOTIVATION (WHY) e.g. strategy
SCOPE (Competitive context) Planning	List of Things Important to the Business Entity = Class of	List of Processes the Business Performs Process = Class of	List of Locations in Which the Business Operates Node = Major Business	List of Organisations Important to the Business People = Major	List of Events/Cycles Important to the Business Time = Major Business	List of Business Goals/Strategies End/Means = Major Business Goal/Strategy
BUSINESS MODEL (Conceptual) Owning	e.g. Semantic Model Entity = Business Entity Relationship = Business Relationship	e.g. Business Process Model Process = Business Process IO = Business Resource	e.g. Business Logistics System Node = Business Location Link = Business Linkage	e.g. Work Flow Model People = Organisation Unit Work = Work Product	e.g. Master Schedule Time = Business Event Cycle = Business Cycle	e.g. Business Plan End = Business Objective Means = Business Strategy
SYSTEM MODEL (Logical) Designing	e.g. Logical Data Model Entity = Data Entity Relationship = Data Relationship	e.g. Application Architecture Process = Application Function IO = User Views	e.g. Distributed System Architecture Node = IS Function (Process, Storage, etc.) Link = Line Characteristics	e.g. Human Interface Architecture People = Role Work = Deliverable	e.g. Processing Structure Time = System Event Cycle = Processing Cycle	e.g. Business Rule Model End = Structural Assertion Means = Action Assertion
TECHNOLOGY MODEL (Physical) Building	e.g. Physical Data Model Entity = Segment/Table/etc. Relationship = Pointer/Key/etc.	e.g. System Design Process = Computer Function IO = Data Element/sets	e.g. Technology Architecture Node = HW/System Software Link = Line Specifications	e.g. Presentation Architecture People = User Work = Screen Formats	e.g. Control Structure Time = Execute Cycle = Component Cycle	e.g. Role Design End = Condition Means = Action
DETAILED REPRESENTATIONS (out-of-modelling-context) Subcontracting	e.g. Data Definition Entity = Field Relationship = Address	e.g. Program Process = Language Statement IO = Control Block	e.g. Network Architecture Node = Address Link = Network	e.g. Security Architecture People = Identity Work = Job	e.g. Timing Definition Time = Interrupt Cycle = Machine Cycle	e.g. Role Specification End = Solution Means = Action

Source of coloured squares: Zachman Framework, www.zifa.com

Focus on defined value-creating relationships

The context defining that focus is the Enterprise



Software Engineering Institute

CarnegieMellon

Limits to the use of the Zachman Framework

Philip Boxer & Suz Garcia, May 6th 2009
© 2009 Carnegie Mellon University

5

Agenda

The Enterprise/Architecture relationship

➔ The demands of collaborative systems of systems

Limits to the use of the Zachman Framework & the consequences for DODAF 2.0

Summary



Software Engineering Institute

CarnegieMellon

Limits to the use of the Zachman Framework

Philip Boxer & Suz Garcia, May 6th 2009
© 2009 Carnegie Mellon University

6



Software Engineering Institute

CarnegieMellon

© 2008 Carnegie Mellon University

3

Speech by Secretary Gates:

There are two paradigms that must coexist

The need for state of the art systems – particularly longer range capabilities – will never go away...

We also need specialized, often relatively low-tech equipment for stability and counter-insurgency missions.

- How do we institutionalize rapid procurement and fielding of such capabilities?
- Why do we currently have to go outside the normal bureaucratic process?

Our conventional modernization programs seek a 99% solution in years.

Stability and counter-insurgency missions require 75% solutions in months.

- The challenge is whether in our bureaucracy and in our minds these two different paradigms can be made to coexist.

Extracted from speech delivered by Secretary of Defense Robert M. Gates, National Defense University, Washington, D.C. September 29, 2008 <http://www.defenselink.mil/speeches/speech.aspx?speechid=1279>



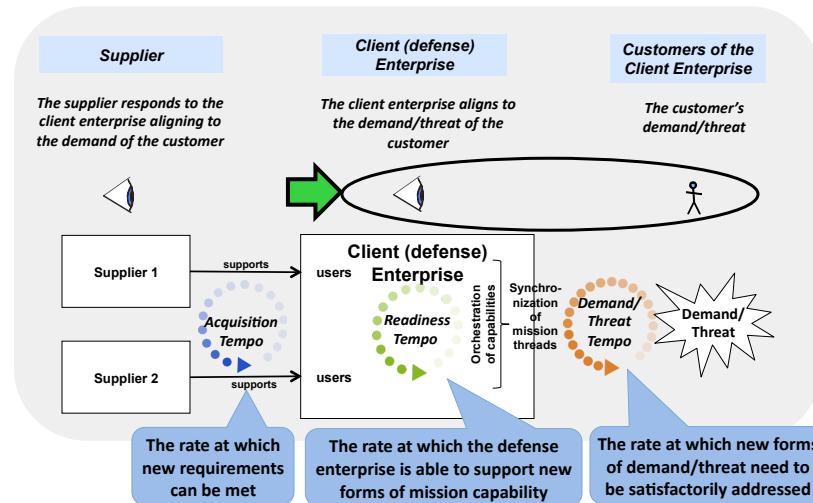
Software Engineering Institute | CarnegieMellon

Limits to the use of the Zachman Framework

Philip Boxer & Suz Garcia, May 6th 2009
© 2009 Carnegie Mellon University

7

The three tempos: analyzing the impact of the enterprise's relation to customers' changing demands



Software Engineering Institute | CarnegieMellon

Limits to the use of the Zachman Framework

Philip Boxer & Suz Garcia, May 6th 2009
© 2009 Carnegie Mellon University

8



Managing diverging tempos: *the readiness tempo has to be managed in its own right*

The two paradigms are about diverging acquisition and demand/threat tempos

- Their coexistence depends on managing the readiness tempo in its own right

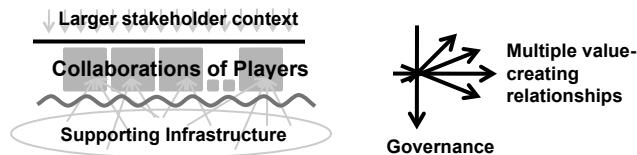
Managing the readiness tempo means:

- sustaining *multiple collaborations* between players able to address *concurrent* types of demand/threat
- building *organizational agility* into the supporting socio-technical infrastructures



Governance of a Collaborative SoS: *involves multiple collaborations with a supporting infrastructure*

The players in a collaboration can be spread across multiple enterprises and/or different parts of a single enterprise



It is the players participating in a particular collaboration who will define

- Their system-of-interest and its environment
- The stakeholders they judge to be relevant
- The way they want their collaboration supported by the infrastructure



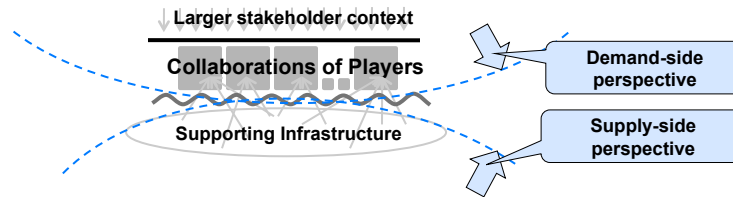
And so... a demand-side perspective needs to be added

Collaborative SoS present a different order of complexity

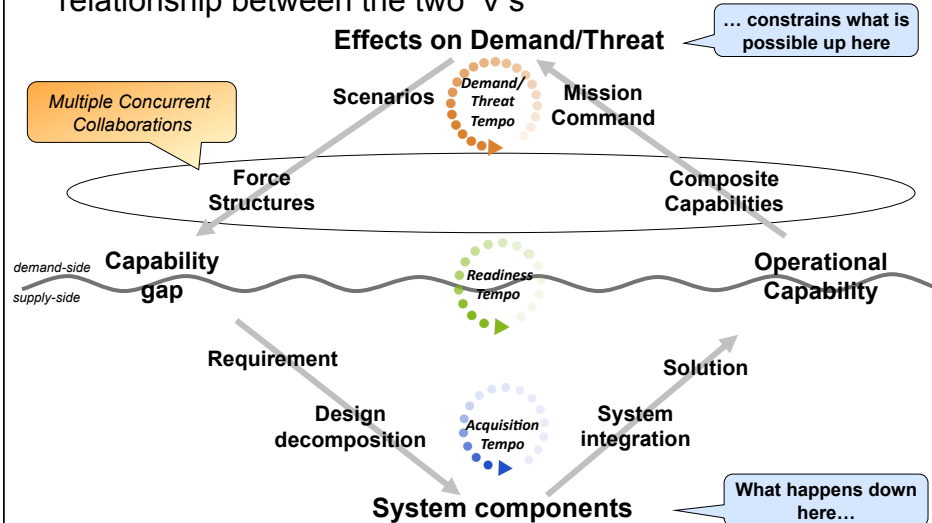
This complexity arises because

- multiple collaborations between players exist concurrently,
- each with its own relationship to demand/threat, and
- supported by a shared infrastructure

It means adding a demand-side perspective on the collaborations



Managing both paradigms: means managing the relationship between the two 'V's



Boxer, P.J. (2007) Managing the SoS Value Cycle, January 2007, <http://www.asymmetricdesign.com/archives/85>



Agenda

The Enterprise/Architecture relationship

The demands of collaborative systems of systems

➔ Limits to the use of the Zachman Framework & the consequences for DODAF 2.0

Summary



Software Engineering Institute

CarnegieMellon

Limits to the use of the Zachman Framework

Philip Boxer & Suz Garcia, May 6th 2009
© 2009 Carnegie Mellon University

13

The demand-side perspective: *creates gaps in Zachman*

	EVENT (WHAT) e.g. things done	DATA (WHAT) e.g. data	FUNCTION (HOW) e.g. function	NETWORK (WHERE) e.g. network	PEOPLE (WHO) organisation	TIME (WHEN) e.g. schedule	USE CONTEXT (WHO for WHOM) e.g. particular client	MOTIVATION (WHY) e.g. strategy
SCOPE (Competitive context) Planning		List of Things Important to the Business Entity = Class of Business Thing	List of Processes the Business Performs Process = Class of Business Process	List of Locations in Which the Business Operates Node = Major Business Location	List of Organisations Important to the Business People = Major Organisational Unit	List of Events/Cycles Significant to the Business Time = Major Business Event/Cycle		List of Business Goals/Strategies Ends/Means = Major Business Goal/Strategy
COLLABORATIVE MODEL (Collaboration) Governance								
BUSINESS MODEL (Conceptual) Owning		e.g. Semantic Model Entity = Business Entity Relationship = Business Relationship	e.g. Business Process Model Process = Business Process IO = Business Resource	e.g. Business Logic System Node = Business Location Link = Business Linkage	e.g. Work Flow Model People = Organisation Unit Work = Work Product	e.g. Master Schedule Time = Business Event Cycle = Business Cycle		e.g. Business Plan End = Business Objective Means = Business Strategy
SYSTEM MODEL (Logical) Designing		e.g. Logical Data Model Entity = Data Entity Relationship = Data Relationship	e.g. Application Architecture Process = Application Function IO = User Input	e.g. Distributed System Architecture Node = Sys/Network Software Link = Data Specification	e.g. Information Architecture People = User Work = Screen Formats	e.g. Processing Structure Time = System Event Cycle = Processing Cycle		e.g. Business Role Model End = Structural Assertion Means = Action Assertion
TECHNOLOGY MODEL (Physical) Building		e.g. Physical Data Model Entity = Segment/Table/etc. Relationship = Point-to-Point	e.g. System Design Process = Computer Function IO = Data Element/Set	e.g. Technology Architecture Node = Hardware Link = Protocol	e.g. Presentation Architecture People = User Work = Screen Formats	e.g. Control Structure Time = Execute Cycle = Component Cycle		e.g. Role Design End = Condition Means = Action
DETAILED REPRESENTATIONS (out-of-modelling-context) Subcontracting		e.g. Data Definition Entity = Field Relationship = Address	e.g. Program Process = Language Statement IO = Control Block	e.g. Network Architecture Node = Address Link = Protocol	e.g. Security Architecture People = Identity Work = Job	e.g. Timing Definition Time = Interrupt Cycle = Machine Cycle		e.g. Role Specification End = Sub-condition Means = Step

Source of gaps: Philip Boxer, Modeling structure-determining processes. <http://www.asymmetricdesign.com/archives/59>, December 2006



Software Engineering Institute

CarnegieMellon

Limits to the use of the Zachman Framework

Philip Boxer & Suz Garcia, May 6th 2009
© 2009 Carnegie Mellon University

14



Software Engineering Institute

CarnegieMellon

© 2008 Carnegie Mellon University

7

DODAF 2.0 Entities and Views: what gets modeled?

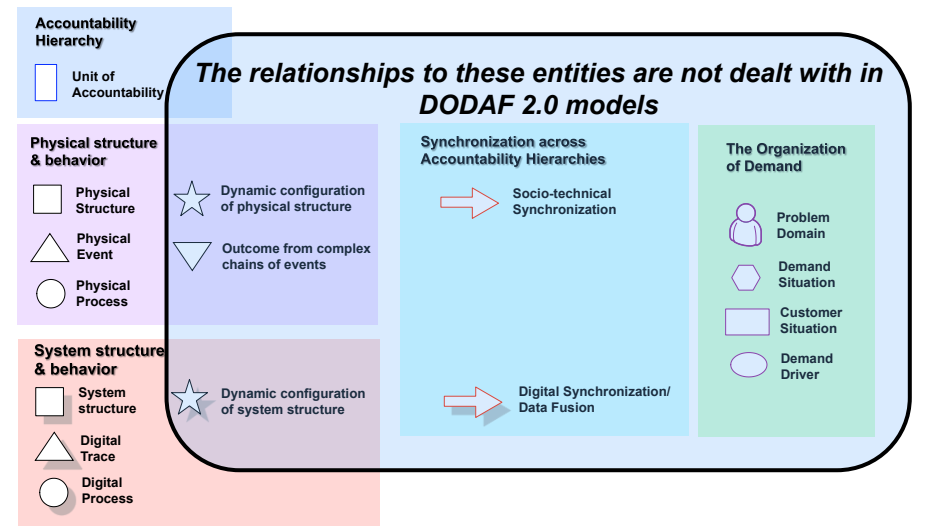
DODAF TAXONOMY TYPES and CADM 2.0 PRINCIPAL INDEPENDENT ENTITIES	All Views (AV)		Operational View (OV)							System View (SV)											Tech View (TV)			
	1	(2)	1	(2)	(3)	(4)	(5)	(6)	(7)	(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)	(9)	(10)	(11)	(1)	(2)		
Operational Nodes <i>Organizations, Types of Organizations, and Operational Roles</i>	-		-	-	-	-	-	-																
Performance Attributes	-		Z	Z	Z					Z											Z	Z		
Technical Standards <i>Info Processing, Info Transfer, Data, Security, and Human Factors</i>	-									Z	Z	Z	Z	Z	Z	Z	Z	Z	Z					
Technology Areas	-																				Z			
Physical Nodes <i>Facilities, Platforms, Units, and Locations (including Features)</i>	-									Z	Z	Z									Z	Z		
Triggers/Events	-																							
Operational Activities (and Tasks)	-		Z																					
Technology Areas	-																				Z			
Systems <i>Families of Systems, Systems of Systems, Networks, Applications, Software, and Equipment</i>	-																				Z	Z		
Information Elements (and Data Elements)	-		Z																		Z			
System Functions	-																				Z	Z		

- = Taxonomy element plays a primary role Z = element plays a secondary role [] = unstructured text or graphics

Modeling Elements

- Accountability Hierarchy**
 - Unit of Accountability
- Physical structure & behavior**
 - Physical Structure
 - Physical Event
 - Physical Process
- System structure & behavior**
 - System Structure
 - Digital Trace
 - Digital Process

Entities not modeled by DODAF 2.0: the demand-side perspective is not included



Describing the demand-side: bridging the gaps

	EVENT (WHAT) <small>e.g. things done</small>	DATA (WHAT) <small>e.g. data</small>	FUNCTION (HOW) <small>e.g. function</small>	NETWORK (WHERE) <small>e.g. network</small>	PEOPLE (WHO) <small>e.g. organisation</small>	TIME (WHEN) <small>e.g. schedule</small>	USE CONTEXT (WHO for WHOM) <small>e.g. particular client</small>	MOTIVATION (WHY) <small>e.g. strategy</small>
SCOPE (Competitive context) Planning	List of Things Important to the Business Entity = Class of Business Thing	List of Processes in the Business Process = Class of Business Process	List of Locations in Which the Business Operates Node = Major Business Location	List of Organisations Important to the Business People = Major Organizational Unit	List of Events/Cycles Important to the Business Time = Major Business Event/Cycle	Ends/Means = Major Business Goal/Strategy		
COLLABORATIVE MODEL (Collaboration) Governance	Different collaborations imply different physical realities 4		Multiple Players in multiple collaborations 4			Different collaborations imply different types of value-creating relation to demand 8		
BUSINESS MODEL (Conceptual) Owning	e.g. Semantic Model Entity = Business Entity Relationship = Business Relationship	e.g. Business Process Model Process = Business Process IO = Business Resources	e.g. Business Logistics System Node = Business Location Link = Business Linkage	e.g. Work Flow Model People = Organization Unit Work = Work Product	e.g. Manufacturing Time = Business Event Cycle = Business Cycle	e.g. Business Plan Ends/Means = Business Strategy		
SYSTEM MODEL (Logical) Designing	e.g. Logical Data Model Entity = Data Entity Relationship = Data Relationship	e.g. Application Process = Application Function IO = User Interaction	DODAF 2.0 Entities System (etc.) Resource	e.g. Human Interface Architecture People = Role Work = Deliverable	e.g. Processing Structure Time = System Event Cycle = Processing Cycle	e.g. Business Rule Model End = Structural Assertion Means = Action Assertion		
TECHNOLOGY MODEL (Physical) Building	e.g. Physical Data Model Entity = Segment/Table/etc. Relationship = Pointer/Key/etc.	e.g. System Design Process = Computer Function IO = Data Elements/Sets	e.g. Technology Architecture Node = HW/System Software Link = Line Specifications	e.g. Presentation Architecture People = User Work = Screen Formats	e.g. Control Structure Time = Event Cycle = Control Cycle	e.g. Role Design End = Condition Means = Action		
DETAILED REPRESENTATIONS (out-of-modelling-context) Subcontracting	e.g. Data Definition Entity = Frame Relationship = Address	e.g. Program Process = Language Statement Link = Control Block	e.g. Network Architecture Node = Address Link = Protocol	e.g. Security Architecture People = Identity Work = Job	e.g. Timing Definition Time = Interrupt Cycle = Machine Cycle	e.g. Role Specification End = Sub-condition Means = Step		

Source of gaps: Philip Boxer, Modeling structure-determining processes. <http://www.svmmetricdesign.com/archives/59>, December 2006

Software Engineering Institute | CarnegieMellon

Limits to the use of the Zachman Framework
Philip Boxer & Suz Garcia, May 6th 2009
© 2009 Carnegie Mellon University

Agenda

- The Enterprise/Architecture relationship
- The demands of collaborative systems of systems
- Limits to the use of the Zachman Framework & the consequences for DODAF 2.0

➤ Summary

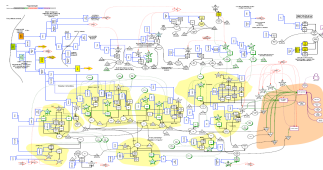
Software Engineering Institute | CarnegieMellon

Limits to the use of the Zachman Framework
Philip Boxer & Suz Garcia, May 6th 2009
© 2009 Carnegie Mellon University

Summary: *both supply-side and demand-side perspectives need to be modeled*

Supporting the development of collaborative systems of systems involves modeling more than the supply-side entities in Zachman-rooted representations like DODAF 2.0

- Including a demand-side perspective means being able to account for
 - cross-cutting synchronization, not just hierarchical accountability
 - multi-enterprise development and co-evolution
 - inherent variation in the way user's demands emerge and evolve
 - the resultant tempo of the ongoing development of systems of systems



Software Engineering Institute

CarnegieMellon

Limits to the use of the Zachman Framework

Philip Boxer & Suz Garcia, May 6th 2009
© 2009 Carnegie Mellon University

19

If you're a software architect...so what?

If you think/know you're involved in a SoS collaboration,

- It is likely that the requirements you are working to do NOT account for sufficient demand-side variety
 - Don't over-constrain your software architecture too early
 - Look for architectural mechanisms that can accommodate later information on interfaces and implementations
- Try to find out the level of awareness of SoS issues that is present on the part of your systems engineers
 - The more they are aware of their lack of control over organizational and technical interactions across the collaboration, the less likely they will be to pass down over-constraining architecture requirements to the software
 - If awareness of SoS issues is low, find out how they are planning to deal with some of the demand-side constructs discussed here
- Start thinking about your customers' "operations architecture" – the components and interfaces that they are operating with and that you are supporting with your software
 - Look for points of complementarity and conflict between your software architecture and your customer's "operations architecture"



Software Engineering Institute

CarnegieMellon

Limits to the use of the Zachman Framework

Philip Boxer & Suz Garcia, May 6th 2009
© 2009 Carnegie Mellon University

20



Software Engineering Institute

CarnegieMellon

© 2008 Carnegie Mellon University

10

NO WARRANTY

THIS CARNEGIE MELLON UNIVERSITY AND SOFTWARE ENGINEERING INSTITUTE MATERIAL IS FURNISHED ON AN "AS-IS" BASIS. CARNEGIE MELLON UNIVERSITY MAKES NO WARRANTIES OF ANY KIND, EITHER EXPRESSED OR IMPLIED, AS TO ANY MATTER INCLUDING, BUT NOT LIMITED TO, WARRANTY OF FITNESS FOR PURPOSE OR MERCHANTABILITY, EXCLUSIVITY, OR RESULTS OBTAINED FROM USE OF THE MATERIAL. CARNEGIE MELLON UNIVERSITY DOES NOT MAKE ANY WARRANTY OF ANY KIND WITH RESPECT TO FREEDOM FROM PATENT, TRADEMARK, OR COPYRIGHT INFRINGEMENT.

Use of any trademarks in this presentation is not intended in any way to infringe on the rights of the trademark holder.

This Presentation may be reproduced in its entirety, without modification, and freely distributed in written or electronic form without requesting formal permission. Permission is required for any other use. Requests for permission should be directed to the Software Engineering Institute at permission@sei.cmu.edu.

This work was created in the performance of Federal Government Contract Number FA8721-05-C-0003 with Carnegie Mellon University for the operation of the Software Engineering Institute, a federally funded research and development center. The Government of the United States has a royalty-free government-purpose license to use, duplicate, or disclose the work, in whole or in part and in any manner, and to have or permit others to do so, for government purposes pursuant to the copyright license under the clause at 252.227-7013.



Software Engineering Institute

CarnegieMellon

Limits to the use of the Zachman Framework

Philip Boxer & Suz Garcia, May 6th 2009
© 2009 Carnegie Mellon University

21

Contact Information

Philip Boxer, Suzanne Garcia

Research, Technology and Systems Solutions Program,
Software Engineering Institute, Carnegie Mellon University

Email: pboxer@sei.cmu.edu, smg@sei.cmu.edu

Mail:
Software Engineering Institute
4500 Fifth Avenue
Pittsburgh, PA 15213-2612
USA



Software Engineering Institute

CarnegieMellon

Limits to the use of the Zachman Framework

Philip Boxer & Suz Garcia, May 6th 2009
© 2009 Carnegie Mellon University

22



Software Engineering Institute | CarnegieMellon

© 2008 Carnegie Mellon University

11