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

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
**Systems of Systems: 4 Types defined by OSD SE**

**Guide for Systems of Systems**

* Player = participant in a collaboration

- **Directed**: the integrated system-of-systems is built and managed to fulfill the specific centrally managed purposes of its owners
  - One player* has dominance
  - No

- **Virtual**: Large-scale behavior emerges—and may be desirable—but this type of SoS must rely upon relatively invisible mechanisms to maintain it.
  - No
  - Many players*, none dominant

- **Collaborative**: The central players collectively provide some means of enforcing and maintaining standards.
  - Yes
  - Relatively few dominant players*

- **Acknowledged**: changes in the (component) systems are based on collaboration between the SoS and the (component) system(s)
  - Yes
  - One player* given dominance


---

**Architectural Genres:**

different genres for different purposes

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

- The enterprise is supported by an infrastructure
- Enterprise Architecture
  - Enabler
  - Quality attributes
- System of systems architecture
  - Directed or Acknowledged Systems of Systems
- System architecture
  - Directed or Acknowledged Systems of Systems
- Software architecture
  - Software architect mostly on the receiving end

These genres reflect a supply-side perspective on the enterprise

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

**SCOPE** (Contextual content) Planning

**BUSINESS MODEL** (Conceptual) Planning

**SYSTEM MODEL** (Logical) Designing

**TECHNOLOGY MODEL** (Physical) Building

**DETAILED REPRESENTATIONS** (out-of-modeling context) Building

Focus on defined value-creating relationships

The context defining that focus is the Enterprise


---

**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
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.


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

Effects on Demand/Threat

... constrains what is possible up here

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

The demand-side perspective: creates gaps in Zachman

Different collaborations imply different physical realities

Multiple players in multiple collaborations

Different collaborations imply different types of value-creating relation to demand

DODAF 2.0 Entities and Views: what gets modeled?

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

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
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

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”
Limits to the use of the Zachman Framework

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

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