caBIG™
cancer Biomedical Informatics Grid™

Semantic Interoperability: The Promise of Rigorous Data Modeling in SOA

Charlie Mead MD, MSc
Chief Technology Officer
NCI CBIIT
Senior Associate
Booz Allen Hamilton

I’ve modified the title…

- **To clarify that although *human* semantic interoperability is important in some contexts, this talk will focus on *computable semantic interoperability (CSI)***
- **To indicate that although *‘rigorous data modeling’* is indeed important in achieving CSI, the term ‘SOA’ provides a context in which it is possible to rigorously define both *static and dynamic* semantics**
- **As such, this talk is really more aptly titled *<u>Achieving Comprehensive CSI in the Context of an SOA</u>***
  - However, since the program was already printed…

caBIG™
cancer Biomedical Informatics Grid™

(Computable) Semantic Interoperability: The Promise of Rigorous Data Modeling in SOA

Charlie Mead MD, MSc
Chief Technology Officer
NCI CBIIT
Senior Associate
Booz Allen Hamilton

The Goal

- **Computable Semantic Interoperability (CSI)**
  - The exchange of *meaning* between two or more computational resources
    - Exchange of meaning does *not* necessarily translate into 'identical processing' by the various computational resources that receive the 'meaning' from a source machine
    - It *does* mean that all resources involved in the exchange engage in processing from the same 'understanding' of the exchanged structure(s)
    - 'Meaning' is not restricted to 'static data structures' and 'exchange' is not restricted to passive exchange of data

- **The 'Four Pillars of CSI' are still standing…they've just been extended to deal with both static *and* dynamic semantics**

The Four Pillars of Computable Semantic Interoperability

Necessary but not Sufficient

- **#1 - Common model across all domains-of-interest**
  - Information model vs Data model
  - The semantics of common structures – Domain Analysis Model
  - <u>Includes, but does not necessarily require, dynamic semantics</u>
    - **Functions**
    - **Behavior**
    - **Interactions**

- **#2- Model bound to robust data type specification**
  - HL7 V3 Abstract Data Type Specification (R2)
  - ISO DT Specification

The Four Pillars of Computable Semantic Interoperability

Necessary but not Sufficient

- **#3 - Methodology for binding terms from concept-based terminologies**
  - Domain-specific semantics
- **#4 - A formally defined process for defining specific structures to be exchanged between machines, i.e. a “data exchange standard”**
  - RM-OPD- and MDA-based Service Specification Methodology
  - Specify service interfaces and interactions
  - Bind ‘semantics on the wire’ to static semantics expressed through Pillars #1, #2, #3

The “Meaning of Meaning”

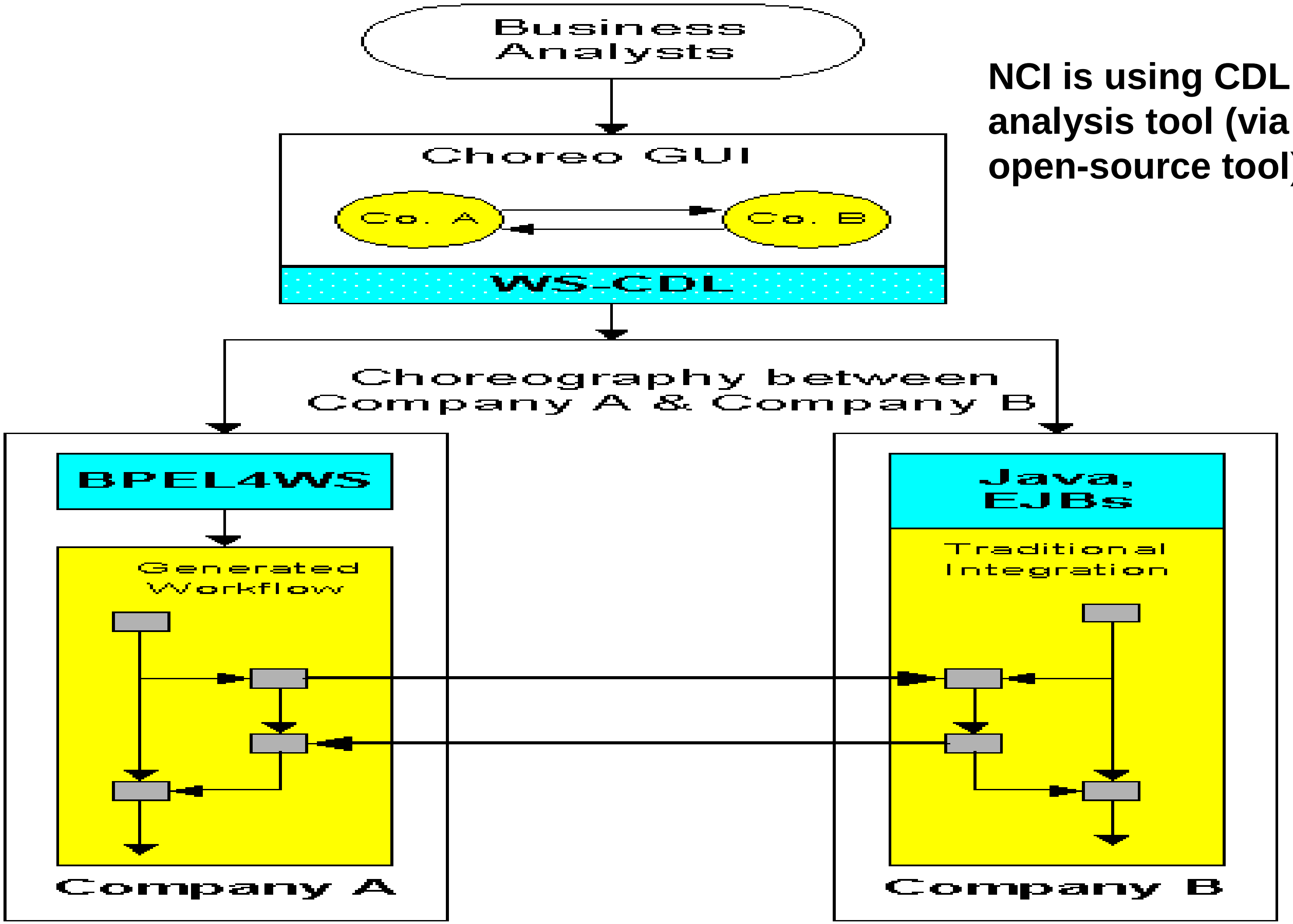
- **Semantics (the term begged by introducing CSI)**
  - Static – the ‘meaning’ of data and meta-data elements *and their various assemblages, collections, descriptions and relationships*
    - To be computationally rigorous, these semantics must be expressed using a *formal* expression language, e.g. UML, OWL, etc. *and* be bound to an equally robust ‘underpinning’ of a robust-defined Abstract/Complex Data Type specification e.g. HL7 V3 ADT Specification (R2) or the ISO DT specification
  - Dynamic – the rigorous description of *function, behavior,* and/or *interactions* by/between computational resources
    - Can be described from several perspectives
    - Isolated
    - Point-to-Point (e.g. BPEL)
    - Global (CDL)

Beyond Static Data

- **In the context of an SOA, dynamic semantics must be rigorously defined**
  - ***A <u>formal</u> representation of a <u>set of activities</u> and deliverables that occur as the result of one or more participating entities <u>requesting</u> or <u>responding to</u> well-defined events in a control flow.***
    - **Pre- and Post-conditions**
    - **Inputs and Outputs**
    - **Error conditions/Exceptions**
    - **<u>Semantic Bindings/Profiles/Signifiers</u>**
      - **The static semantics passed between the various participants in a given interaction**

- **Dynamic semantics may be expressed in either a *messaging* or a *services-oriented* integration environment and apply to operations and sequences of operations**

Choreography: an Analysis Perspective



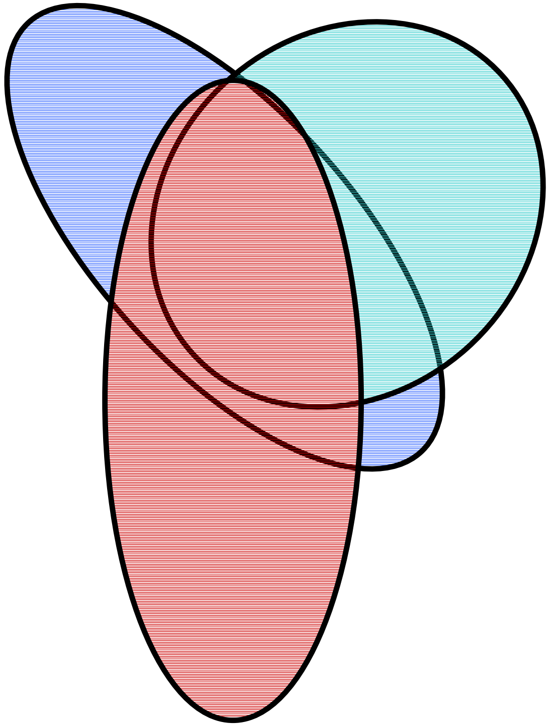
NCI is using CDL as an analysis tool (via pi4soa open-source tool)

NCI's Next Step:
Embracing Enterprise Architecture

- **The Problem Space (translational medicine) is complex**
  - Multiple disciplines and stakeholders
  - Historical isolation
  - Fundamental commitment to change ("We do discovery")
- **The Solution Space is complex**
  - Federated deployment of large number of computational resources
  - Requirement for shared information semantics
  - Requirement for coordinated automated interactions
- **Architecture can be defined as the "stuff that is hard to change." (Fowler)**
- ***Enterprise Architecture is not an emergent property***
  - *Architecture requirements are defined 'bottom up'*
  - *Architecture frameworks and governance are defined 'top down'*

CSI, SOA, and Enterprise Architecture

- The intersection of Enterprise Architecture, SOA, and CSI provide a *goal*, a *methodology*, and the *framework* for defining robust, durable business-oriented constructs that provide extensibility, reuse, and governance.



SOA at NCI

- ***SOA means Architecture, not technology***
  - Use of RM-ODP-like framework (*sans ontology*) to emphasize the role of ‘technology binding’ (e.g. web/Grid services)
    - Assertions made from other non-hierarchical, non-orthogonal perspectives are tested/verified by a given technology binding
      - Enterprise (business rules)
      - Information (static semantics),
      - Computational (dynamic/interaction semantics)
      - Engineering (deployment semantics)
    - There can be more than one technology binding for a given set of assertions
    - The collection of assertions and technology bindings defines a given architecture

The RM-ODP Framework

<u>Enterprise View</u>: concerned with the purpose, scope and policies governing the activities of the specified system within the organization of which it is a part	**Why?**
<u>Information View</u>: concerned with the kinds of information handled by the system and constraints on the use and interpretation of that information;	**What?**
<u>Computational View</u>: concerned with the functional decomposition of the system into a set of objects that interact at interfaces – enabling system distribution;	**How?**
<u>Engineering View</u>: concerned with the infrastructure required to, and distribution of, the computing resources defined in the Computational View.	**Where?**
<u>Technology View</u>: concerned with the choice of technology to support system distribution	**True?**

Services vs Applications

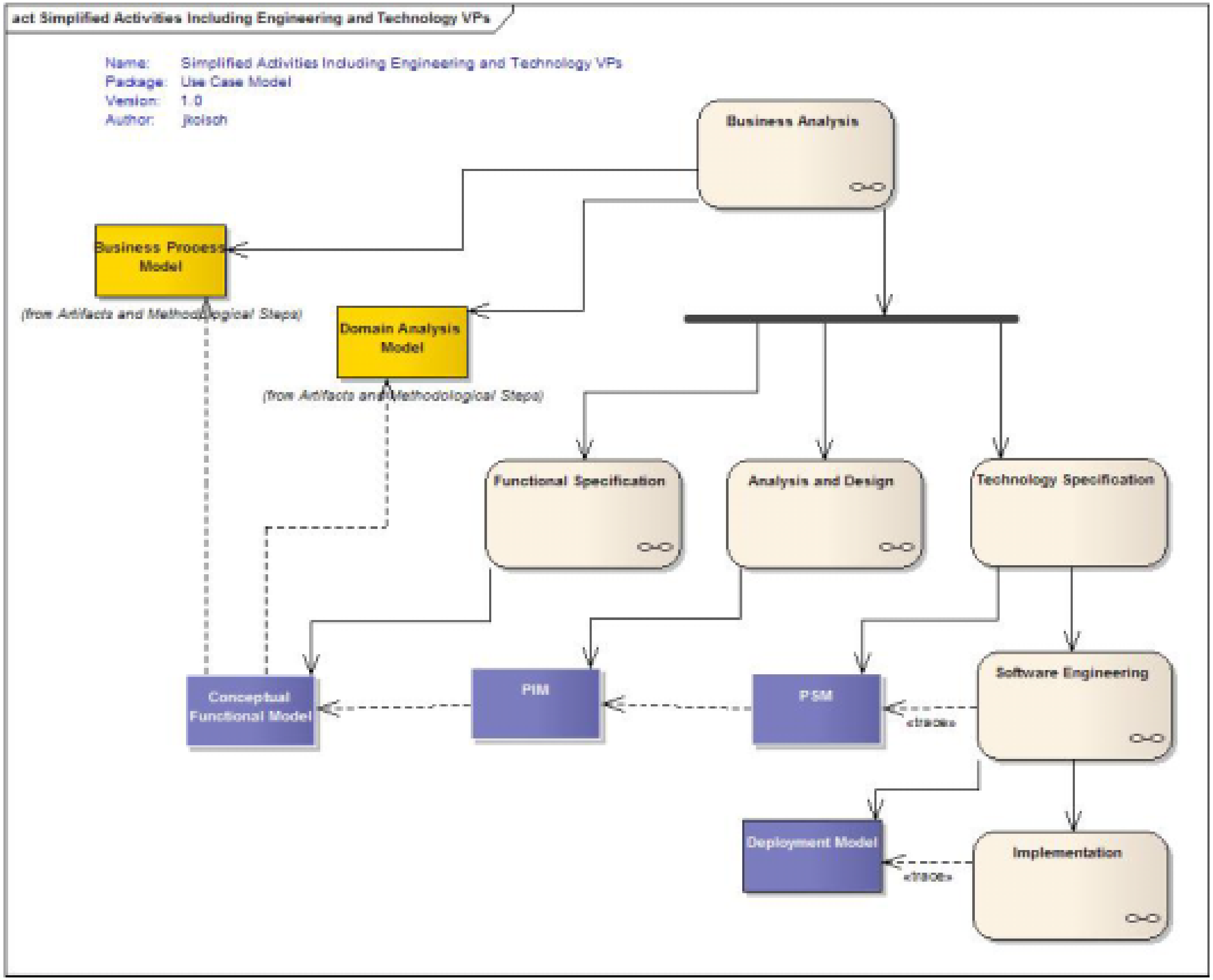
Feature	Services		Applications
	Basic	Choreographies, Aggregated	
Presentation Logic + GUI	N/A	N/A	User Defined
Business Logic	Virtualized	Encapsulated, Specified, and Explicit	Sparse (Possibly realizes of defined choreographies)
Testing Focus	Unit, System	Integration	Integration and User Acceptance
Run Time Dependencies	Possibly other Services	Other Services	Services, Choreographies
Ties to Technology	Loosely coupled	Very Loosely Coupled	Tightly Coupled
Description / Discovery	Explicit Capability, Registered and Advertised	Explicit collections of capabilities, tied to modeled business process	N/A
Reusability	High	Medium	Very Low

MDA as a basis for Service Specification

- **Use of OMG-like ‘model-driven architecture’ as the framework for a Service Specification Methodology**
  - Conceptual Functional Model (CFM)
  - Platform-Independent Model (PIM)
  - Platform-Specific Model (PSM)
  - Deployment Model
  - Implementation
    - into multiple technologies

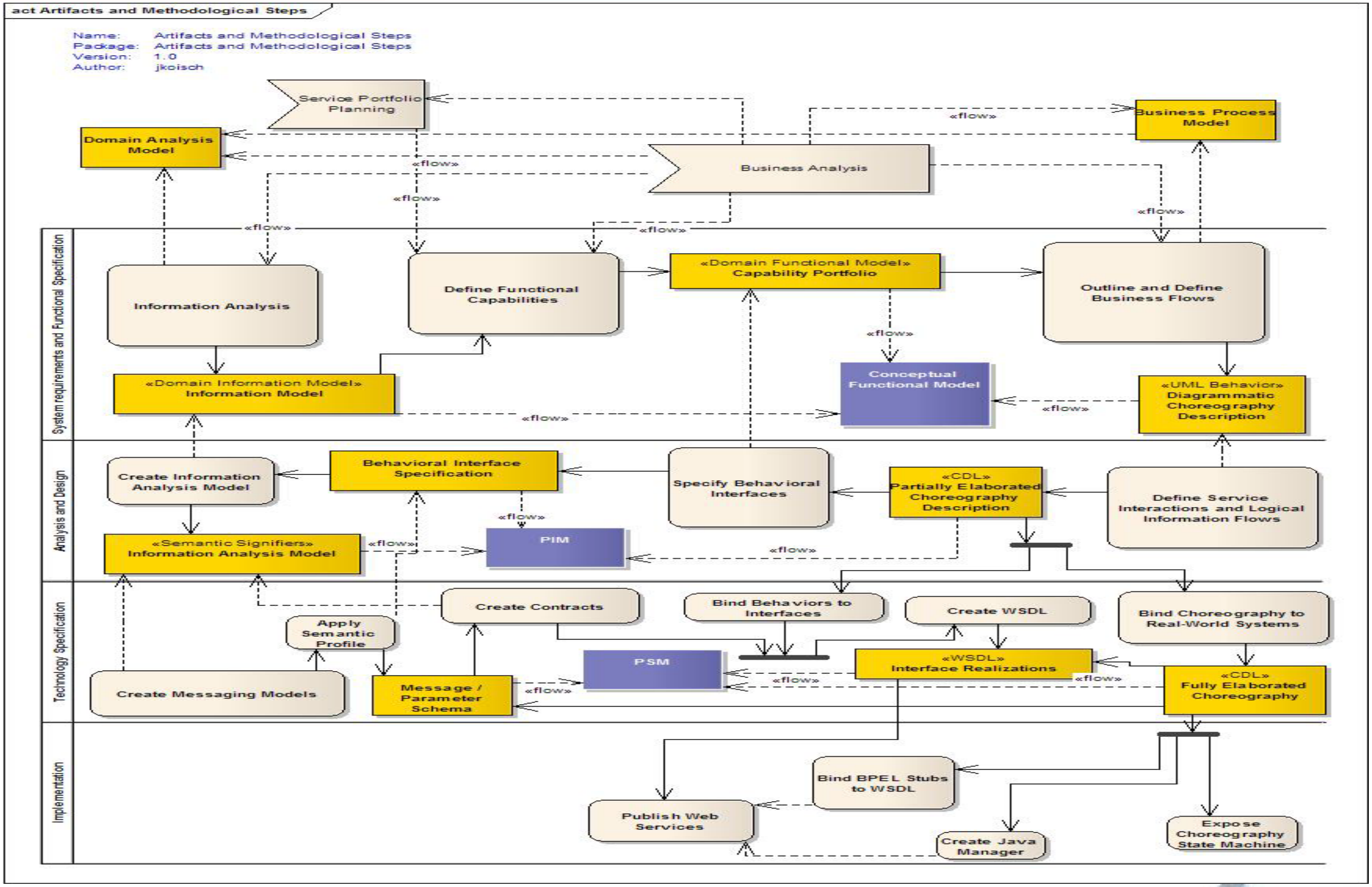
- **Methodology is a modification of HSSP Service Specification Framework (SSF) mapped to Unified Process and encompassing *application and service development***

MDA meets SOA (a)



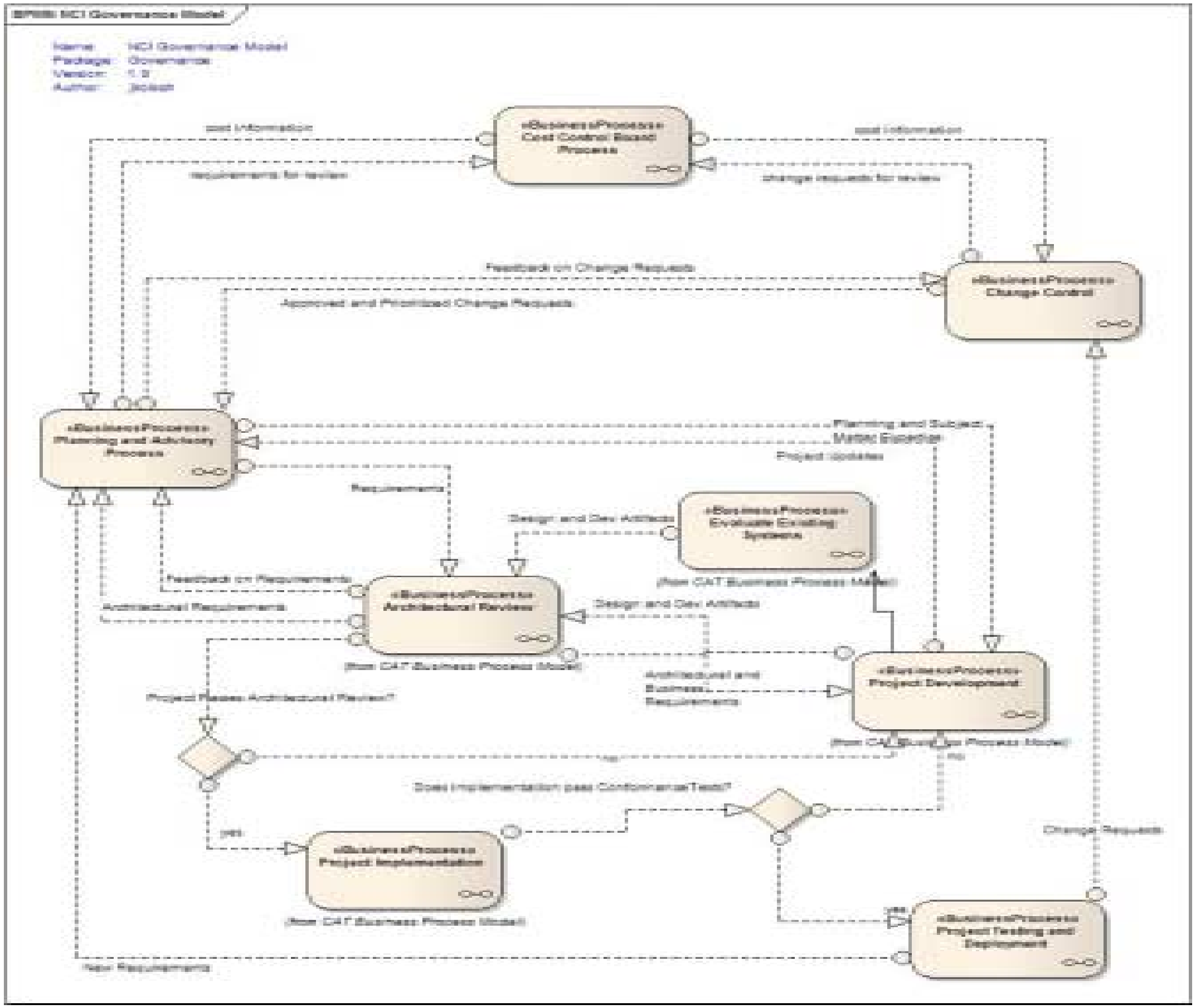
Iteratively and Incrementally define the architecture

MDA meets SOA (b)



caBIG

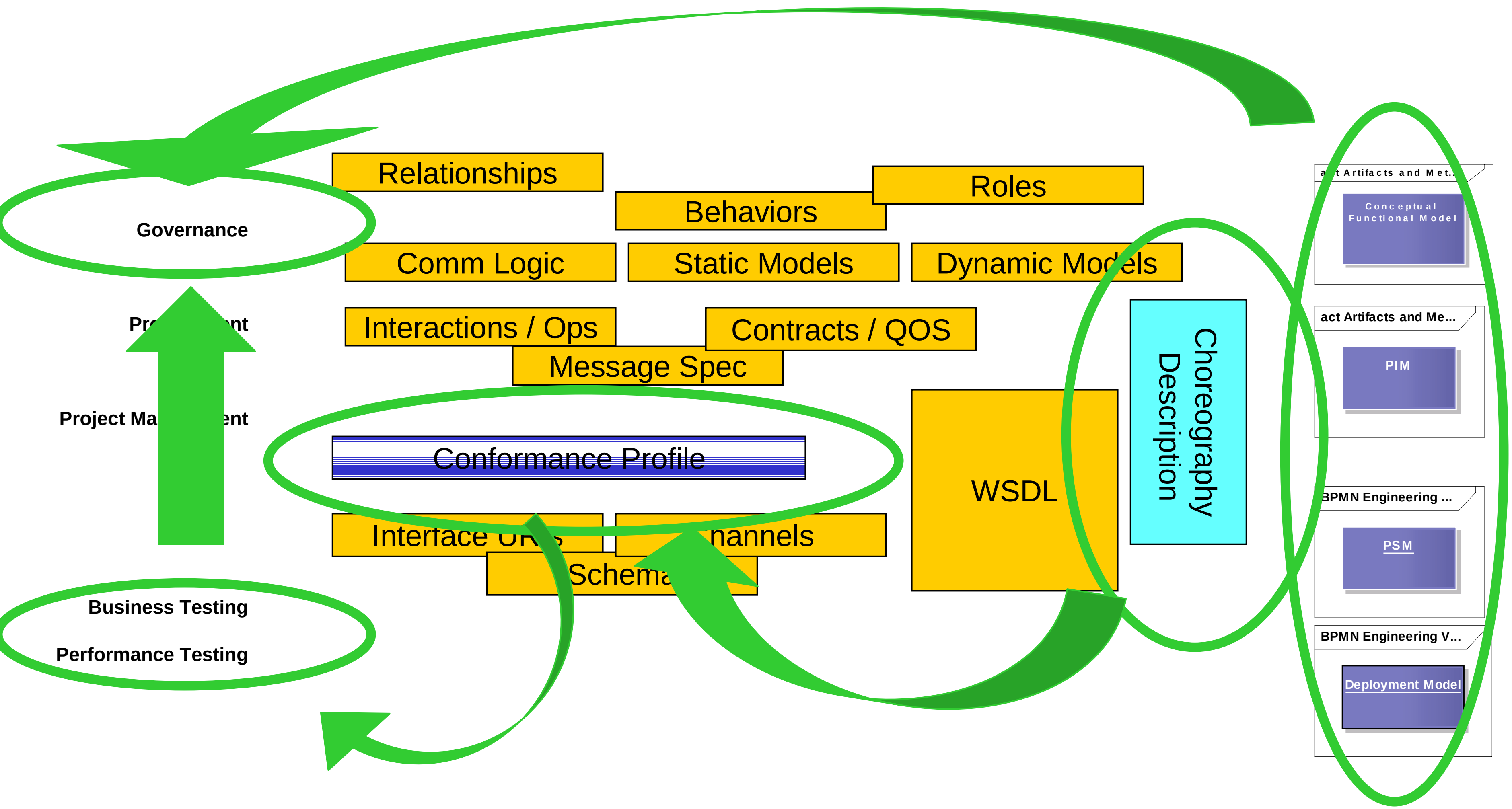
Governance:
Step 0 in building a viable SOA



- You can't have a SOA without Architecture
- *Because you don't know what to build – no semantics defined*

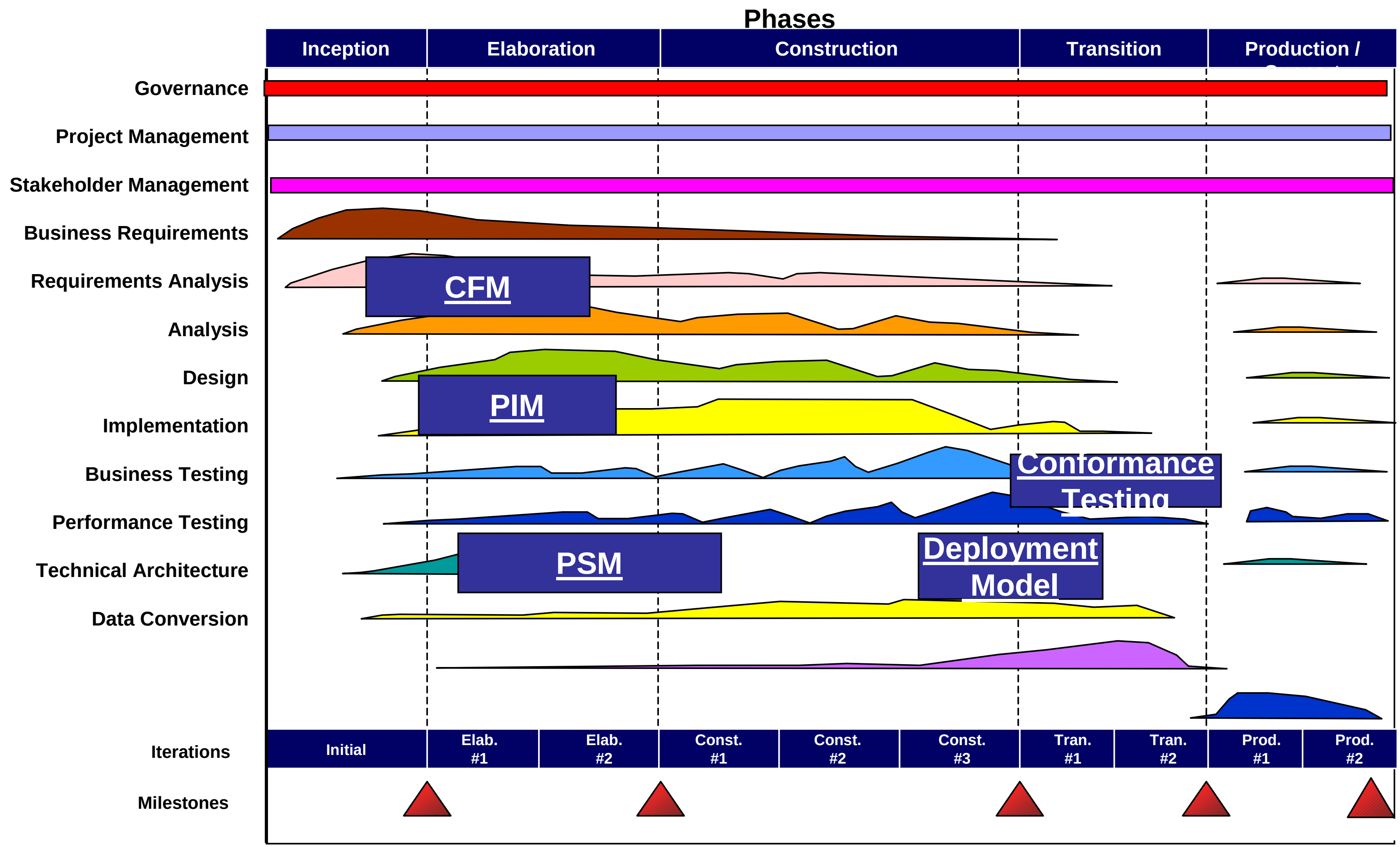
- You can't have a SOA without governance
- *Because you don't know if you've succeeded or even how to succeed*

Governance in Context

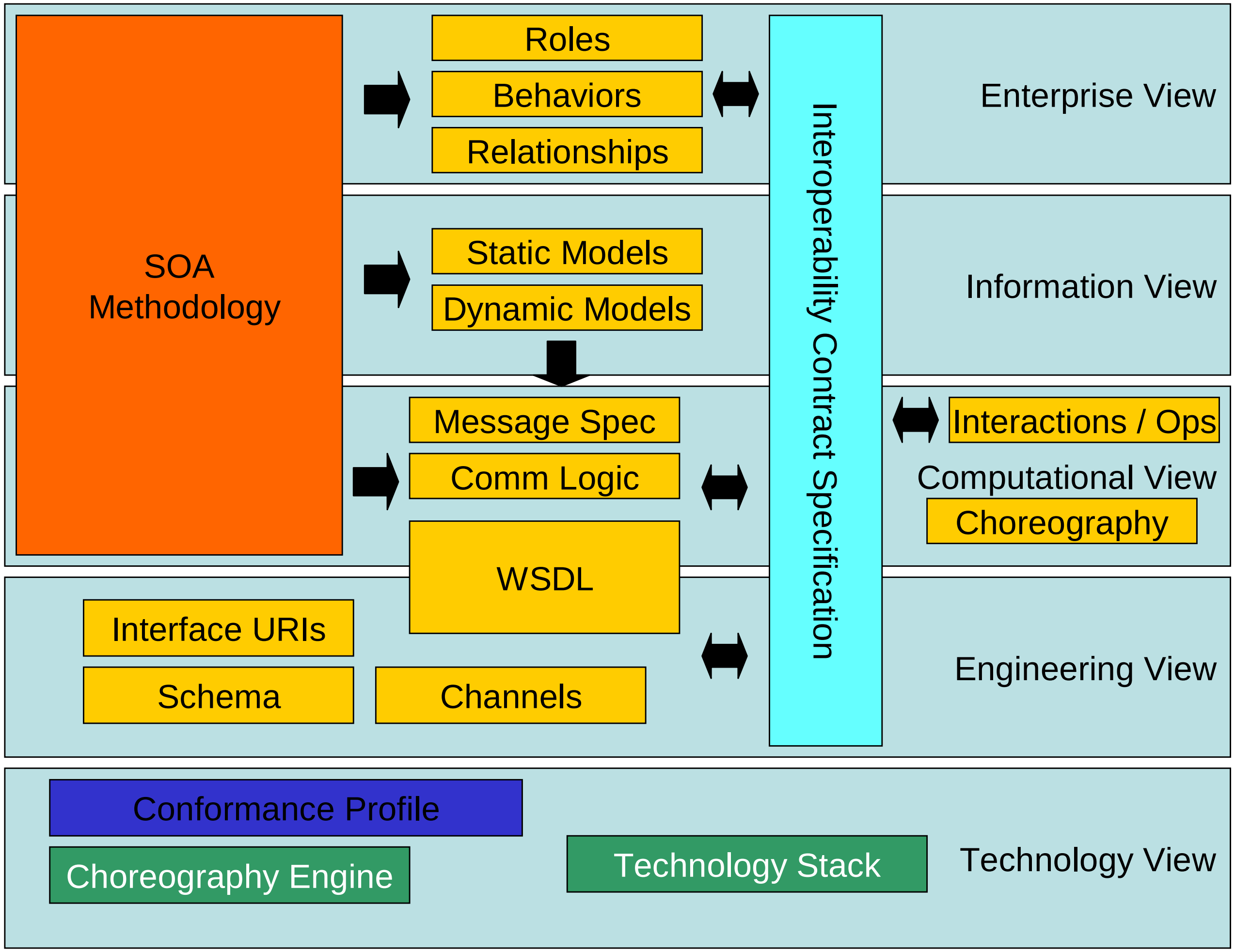


Governace is asserted through the Five Viewpoints

Mapping the UPF Work to Service Specification/MDA Artifacts



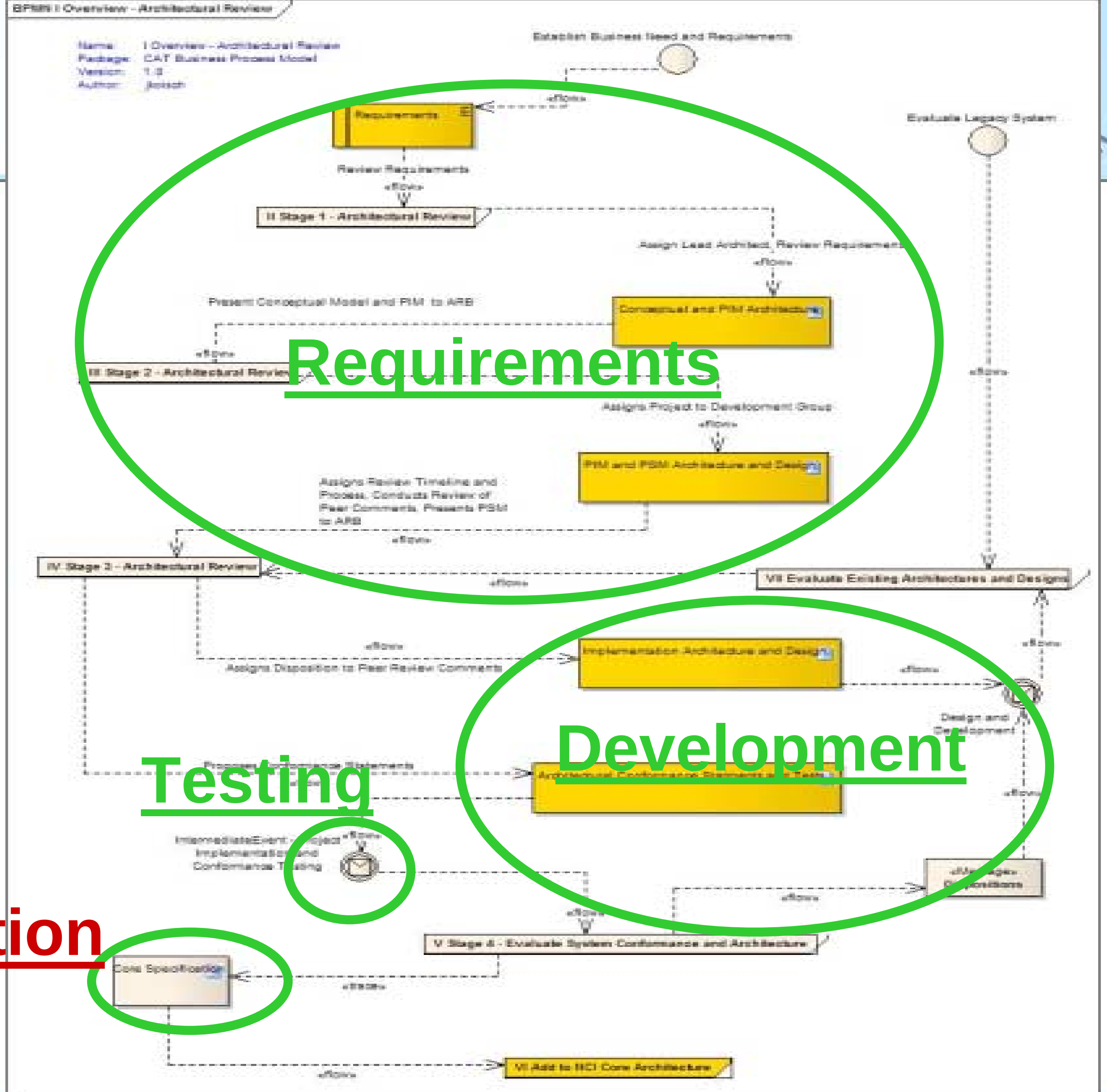
The Relationship between the Methodology and the Five Viewpoints



- **These artifacts are composed into separate models that align with the methodology**

- ***NOTE: CDL's package-level elements align with IHE profile descriptors***

Governance, Architecture, Procurement, and SOW's



Core Specification

Conformance and Compliance: *Working Definitions*

- **<u>Conformance</u> is a layered approach to interoperability within an organization**
  - Not necessarily *Hierarchical* or *Additive*
  - Testable conditions based on assertions (behavioral and/or information-based)
    - Can vary in granularity
    - May involve interactions with other services and their associated asserts/conformance criteria

- **<u>Compliance</u> is a measure of conformance (conformance level) that is encouraged, mandated, and / or enforced in order to achieve a business need (or needs) or to add value to a business process.**

Conformance and Compliance (2)

- Compliance balances global and local requirements

- Compliance within a SOA is focused on a business need for which a defined conformance level is deemed sufficient and for which services and applications can be provided

- Compliance adds value to the organization or to the conforming system

- If compliance is meant to be enforced, it needs to be testable at run-time

caBIG™
cancer Biomedical Informatics Grid™

Thank you!