Applying a Model Driven Methodology for Architecting Successful SOA Healthcare Solutions

April 2008 SOA In Healthcare Conference, Chicago, IL

John Koisch, Booz Allen Hamilton
Alan Honey, Kaiser Permanente

Background

- Based on ***Healthcare Interoperability: A Service-Based Paradigm, Conformance Model, and HL7 Dynamic Model***
  - From Healthcare Services Specification Project (HSSP)
  - Link: http://hssp-Implementation.wikispaces.com/SOAInteroperabilityParadigm
  - Still Work-In-Progress

Contents

- Introduction
- Vision
- Analysis Framework (RM-ODP)
- SOA Methodology
- Standardization
- Dynamic Models and CDL
- Sample
- Current Usage
- Conclusions

Introduction: What we *are* talking about …

- A Service-Oriented methodology that:
  - Is consistent and durable
  - Has the flexibility to derive different technological implementations
  - Supports rich models
  - Does not necessarily require “full” MDA (i.e. automated transforms and code generation)
  - Helps manage the complexity of large distributed architectures

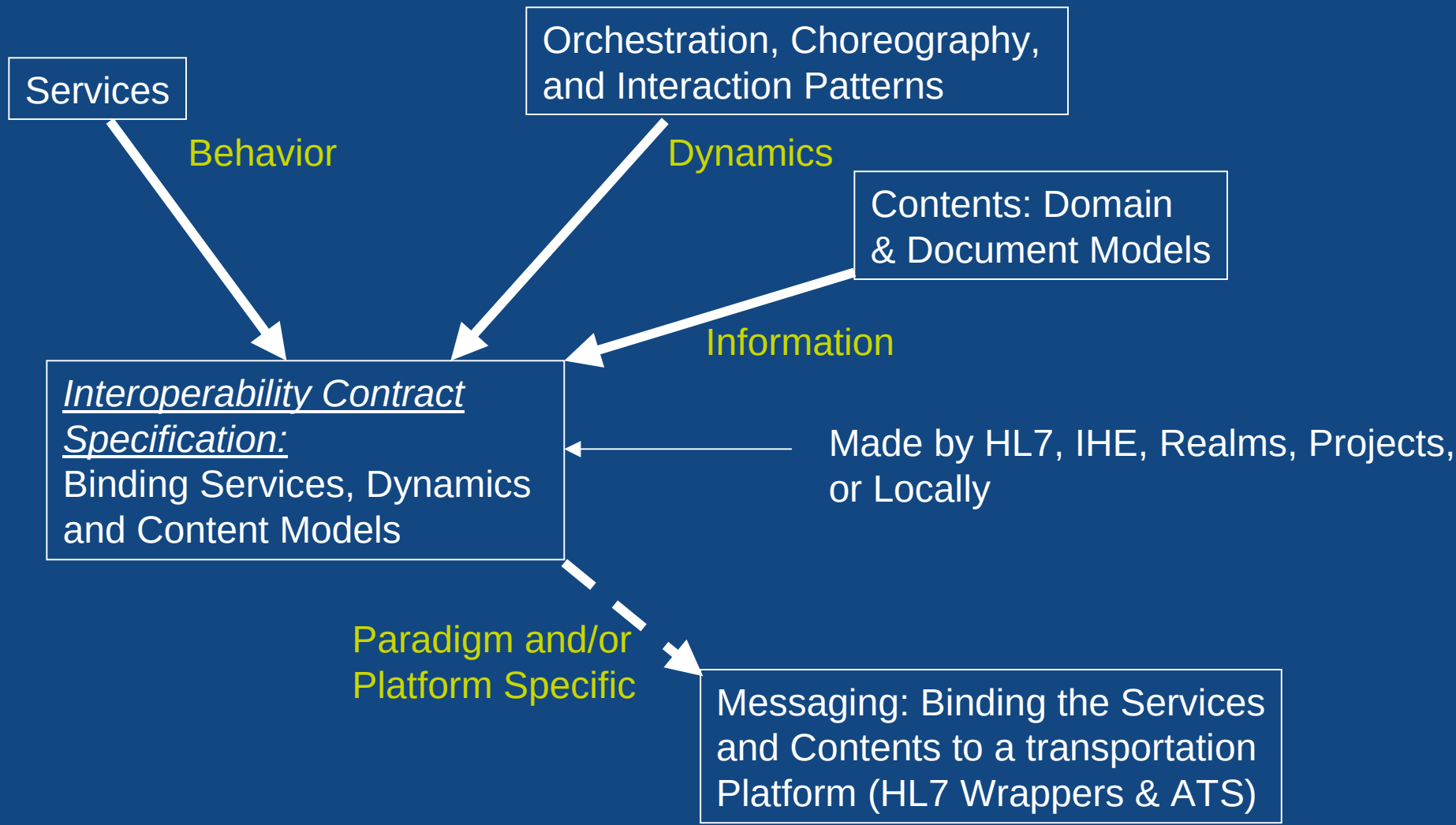
Introduction: What we aren't talking about …

but which is still very important …

Issues	***Notes***
Service Topologies or Architectures	Affects what the services ultimately look like; Granularity; Contract issues
Applications in a Service-Oriented world	Affects migration of legacy capabilities; Interplay with workflows
Computable Semantic Interoperability	Supported by the methodology
Information Modeling	Heavily interdependent with modeling Services

The Unified Vision:

Services, Documents, Messages can share artifacts

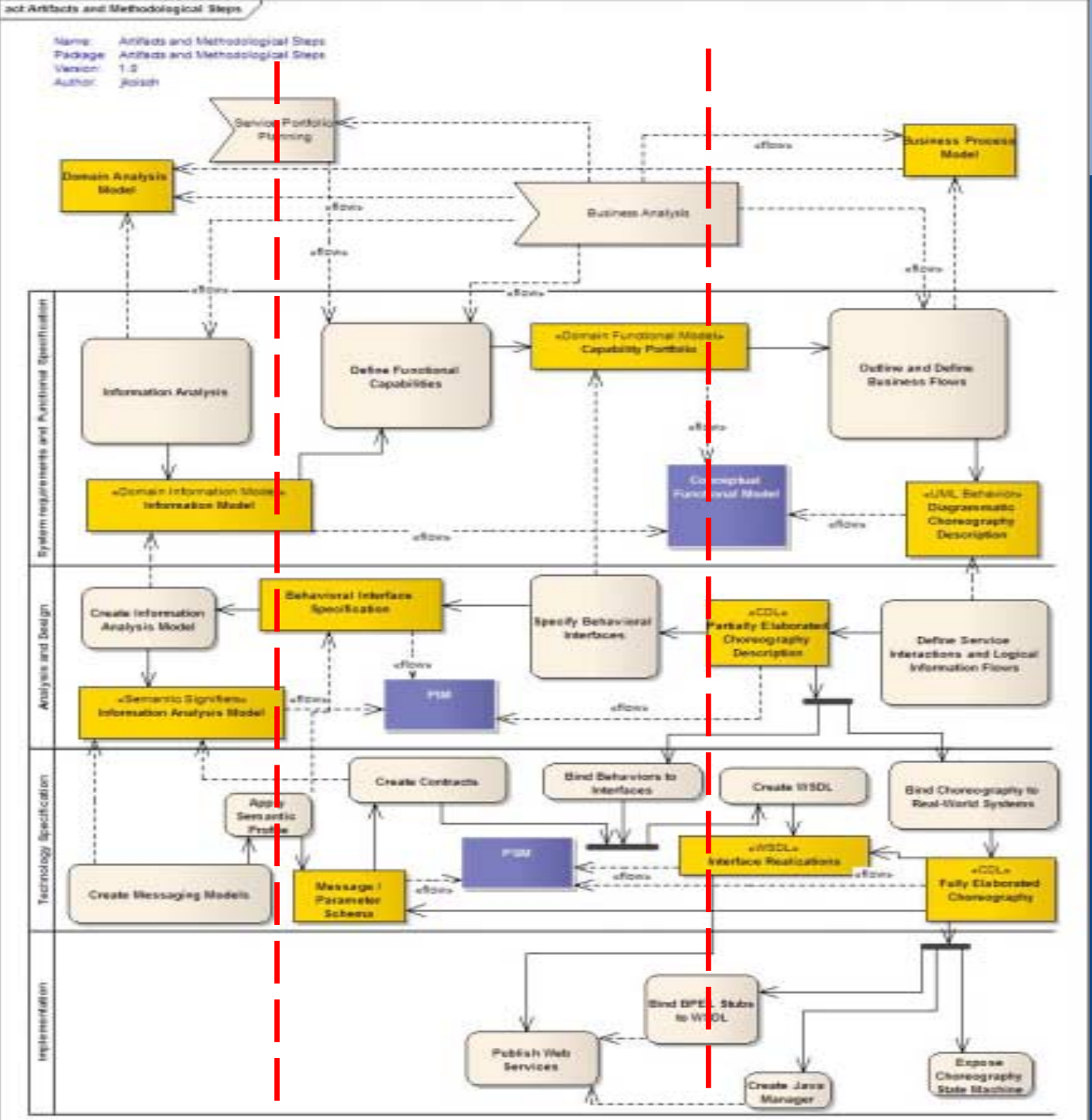


Reference Model for Open Distributed Processing
(RM – ODP, ISO/IEC IS 10746 | ITU-T X.900)

- Five *non-hierarchical* Viewpoints - each with multiple layers
- Ontology which facilitates communication between and among viewpoints and provides a language for Conformance Testing

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

Detailed View



Three views:

1. Information

2. Function

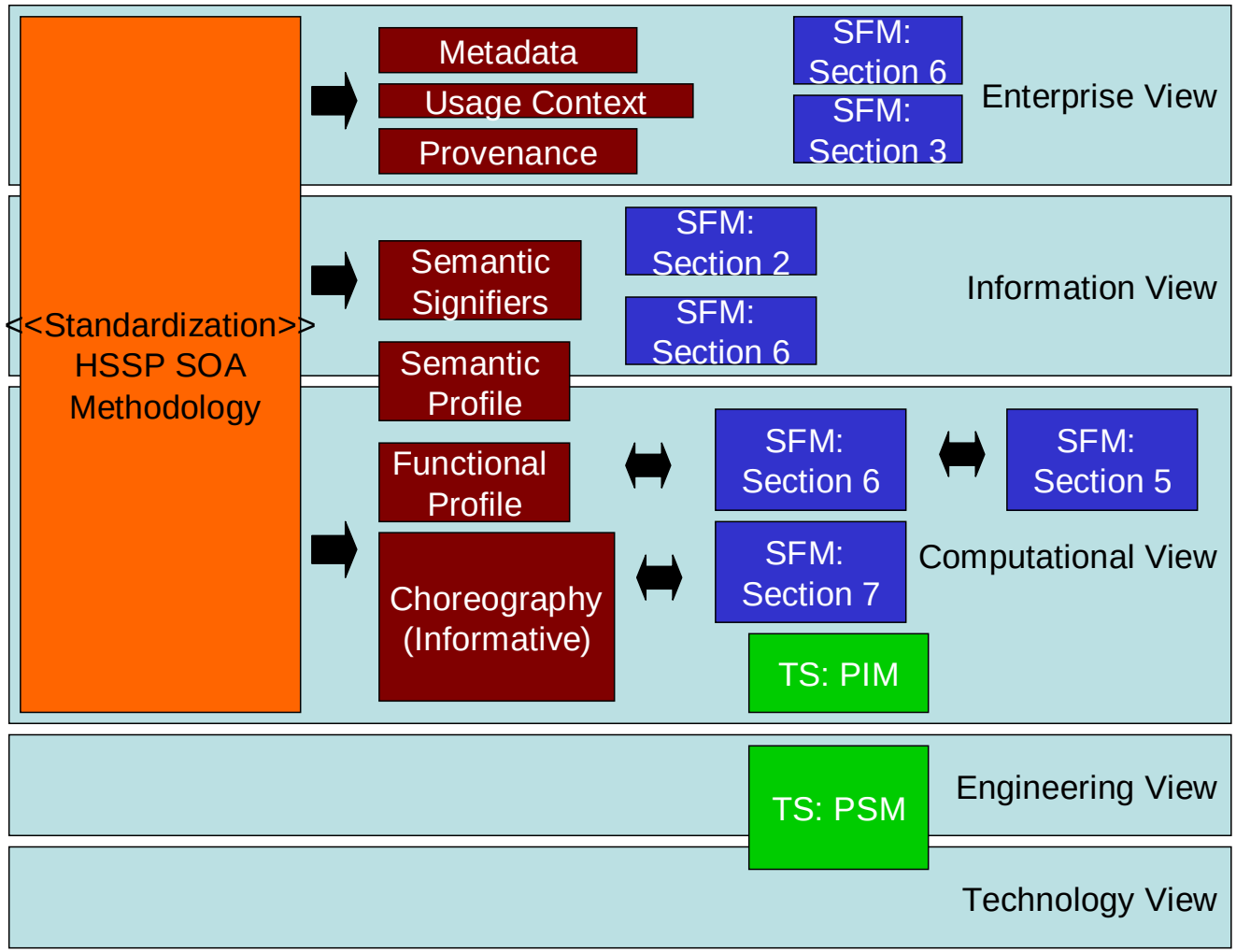
3. Dynamic

See Ann Wrightson's Presentation for more details on the Information View

Functional Model: Service Interface Specification

- A good (rich) Service Specification is critical
  - Design Service Interfaces Carefully
    - Cohesive and complete set of operations
    - Abstraction level / reuse potential – Lab Order Management or Order Management or general Update and Retrieve (RLUS)
  - Operation design qualities
    - Granularity – generally coarse grained preferred
    - Extendable / flexible message schema design (within reason!)
    - Coupling: do not expose unnecessary internal details
    - Determinism – single purpose, minimize processing flags and switches
- Functional Profiles support flexibility and separate service provider and service consumer (Interface) viewpoints

Standardization (HSSP) and the Methodology



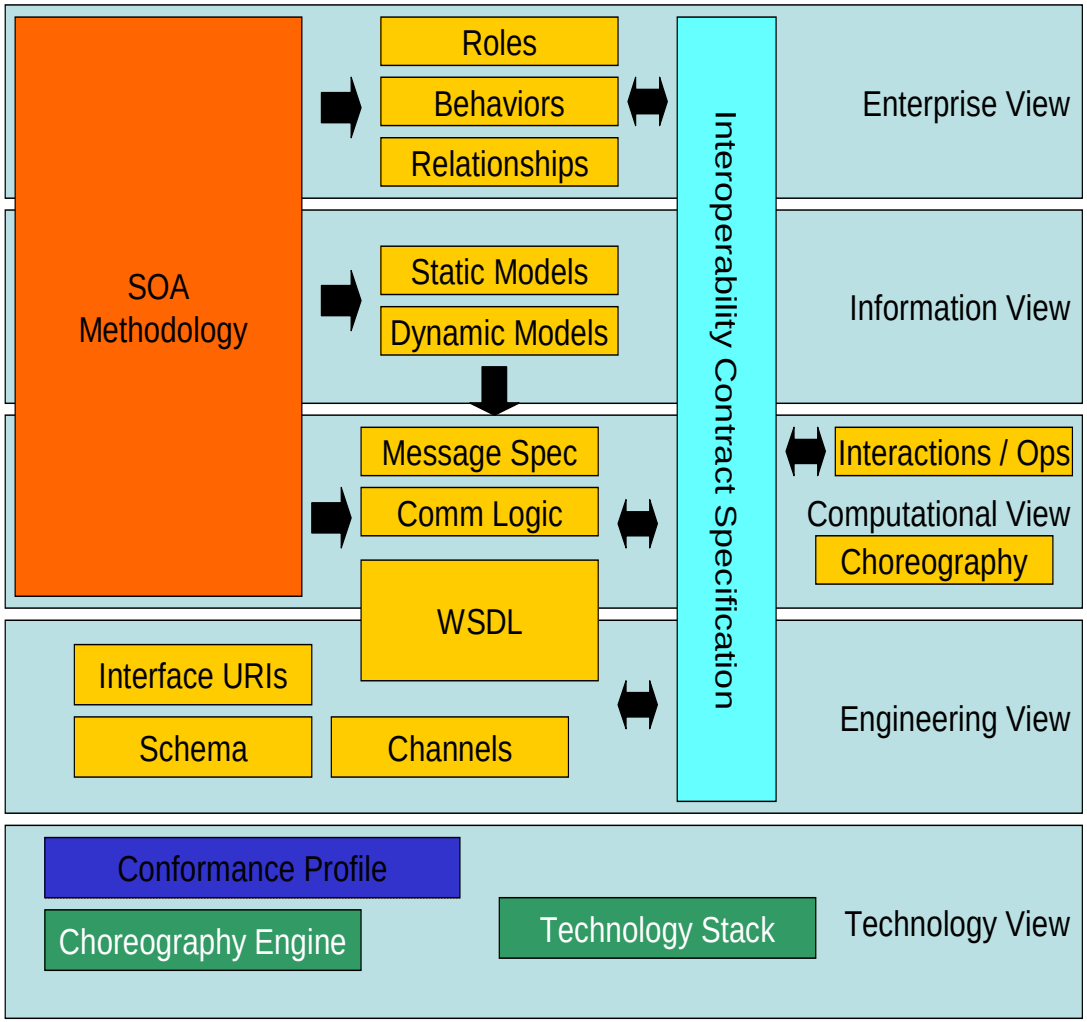
SFM (Blue):
(Service) Functional Model

TS (Green):
Technology Specification
PIM:
Platform Independent Model
PSM:
Platform Specific Model

Conformance and usage context (maroon)

Profiles provide flexibility and enable localization

Moving from Behavioral Interfaces ***OUT***



- CDL is being used at the NCI as a way to bind the viewpoints together
  - First class citizen in the Architecture and Methodology
- Also being used within certain service specifications (CRFQ) as a means of aligning the vision within cross-functional, distributed teams

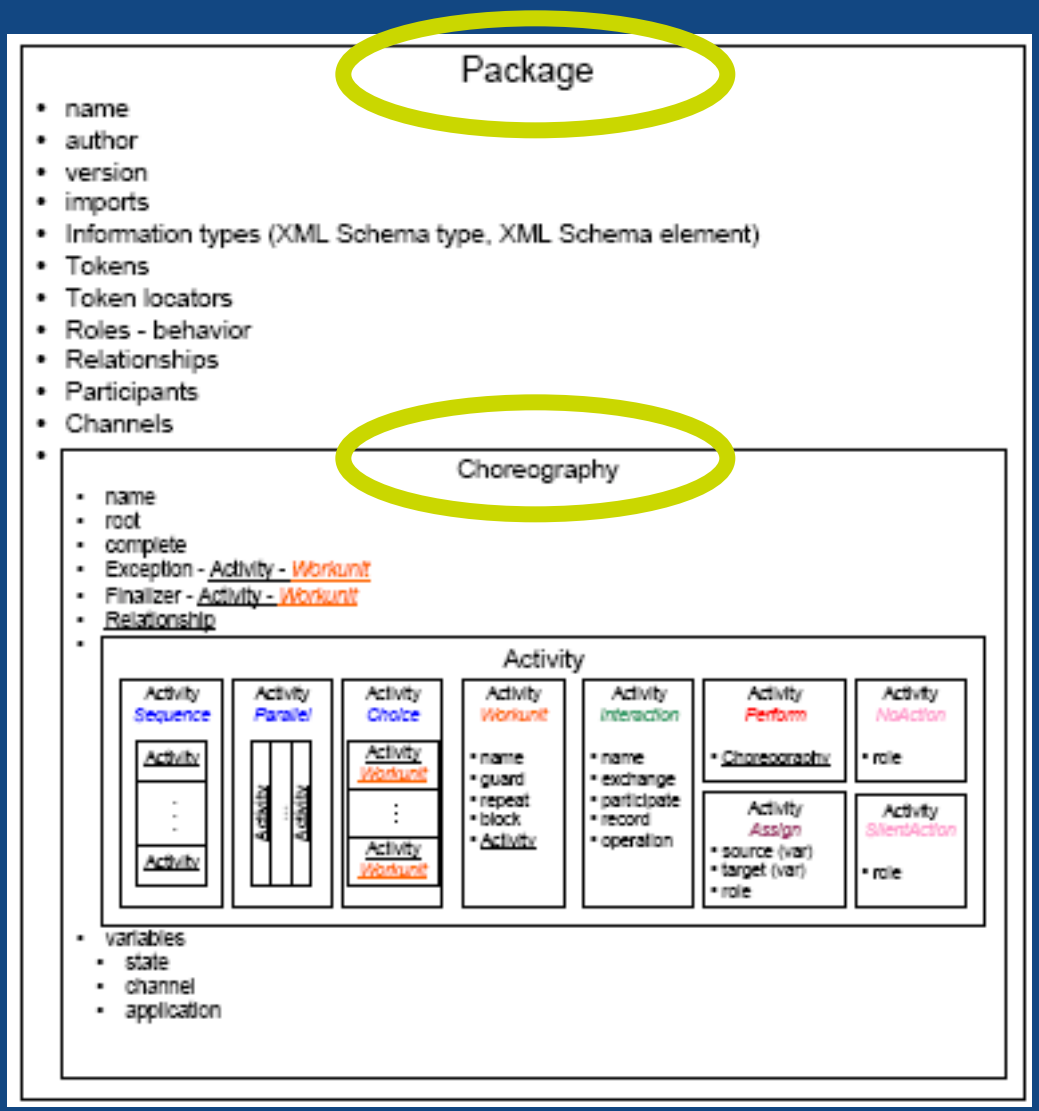
Interoperability Contract Specification

- Uses W3C Choreography Description Language (WS-CDL)
- Interoperability Contract Specification ~ CDL Package
  - Roles, Behaviors, Responsibilities, Participations
  - Channels, Information Types
  - Relationships, Behaviors
- Supports test driven development
- Is a first class analysis artifact that also bridges the gap between architecture and design
  - Ties together the complete semantics in an analysis artifact (informational, functional, dynamic)
  - Serves as an appropriate PIM for Distributed Systems development
  - Supports different layering and structuring of interfaces, operations and behaviors (in line with HSSP Functional profiles)

Dynamic Model: Orchestration and Choreography

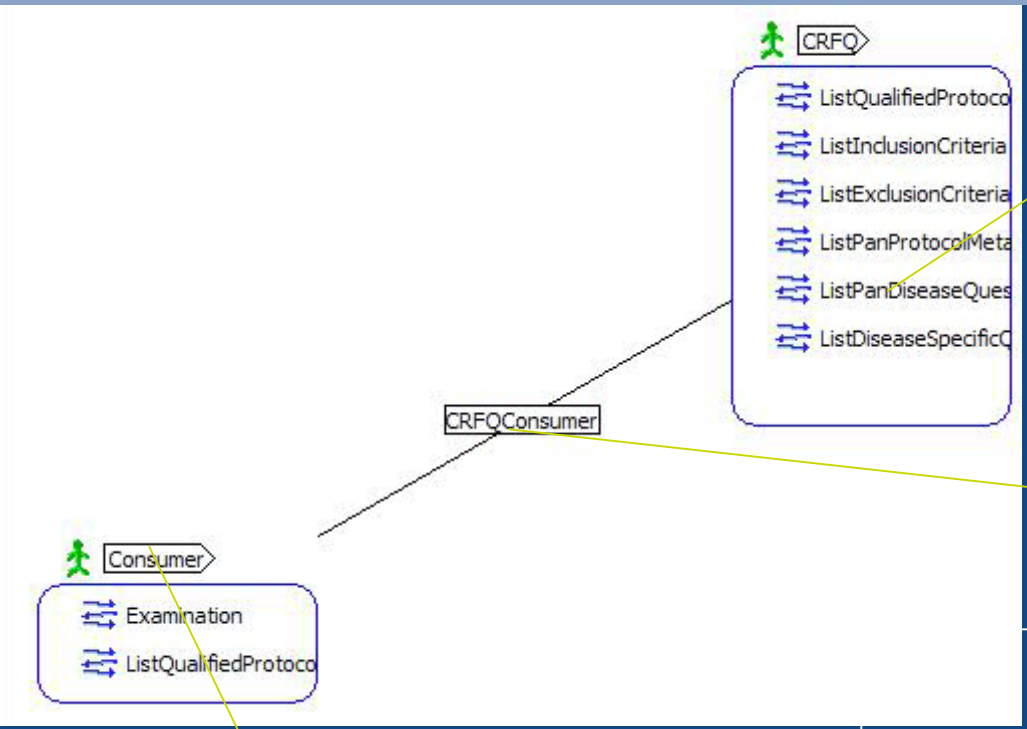
- Orchestration and BPM
  - Services in SOA need to be composed and orchestrated (within a single domain) using BPM techniques and approaches
  - This provides flexibility and ability to configure and change, even at run time
- Choreography
  - This provides a ‘global view’ that can be used for contractual purposes among interoperating parties (across domains, organizations or systems within an organization)
  - Focuses on **peer-to-peer** relationships between parties (services or others) and their interaction responsibilities
  - Processes can be individual components within an overall choreography
- These are complementary, but…
  - Our premise is that for standards and overall organization architecture, a rigorous **choreography** definition is needed to provide the formal basis for usage and management of services
  - Choreographies provide the intersection of standardization and local specifications

CDL Structure



- Choreographies are key artifacts that bridge the gap between analysis and design
  - Business Process scopes a choreography *package*
  - The *package* contains *roles*, which contain *behavior*
  - *Relationships* are between *roles*
  - *Real World Systems* are identified as *participants*
- Choreographies embody the contractual nature of *participation* by real world systems in an identified business context
  - *Participants* take on a *role* (and support the *role's behavior*)
  - *Interactions* realize *relationships* by invoking a *channel* to pass *information*

CDL captures Key Analysis Components



Technology Agnostic Behaviors (note … granularity can differ)

Relationships

Roles

Relationships and HSSP Functional Profiles align

Functional Profile

1 .. *

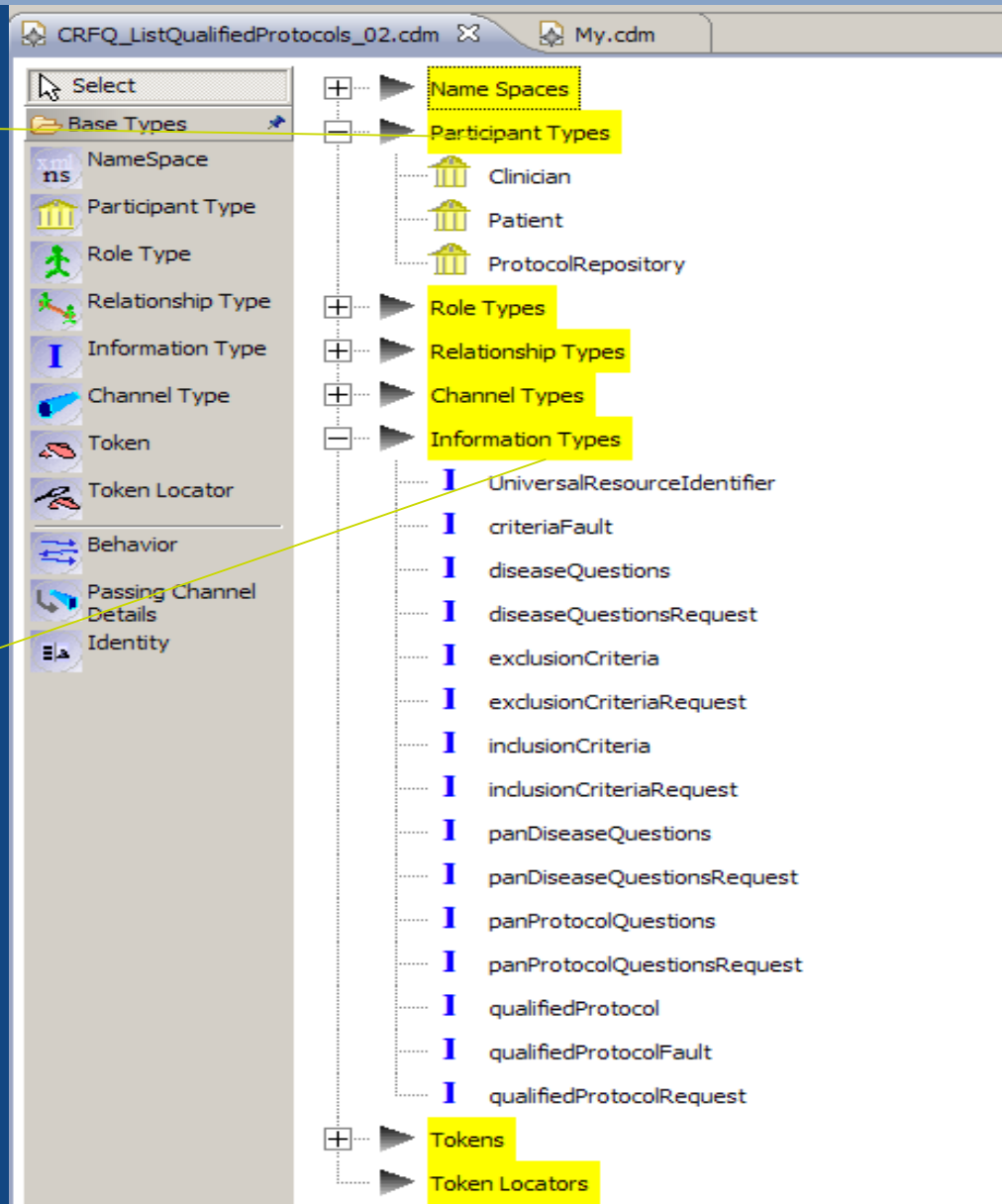
CDL Relationship

Images and Models made using the pi4SOA tool suite (http://sourceforge.net/projects/pi4soa)

Key Analysis Components (cont'd)

Loose Coupling between Participants and Roles

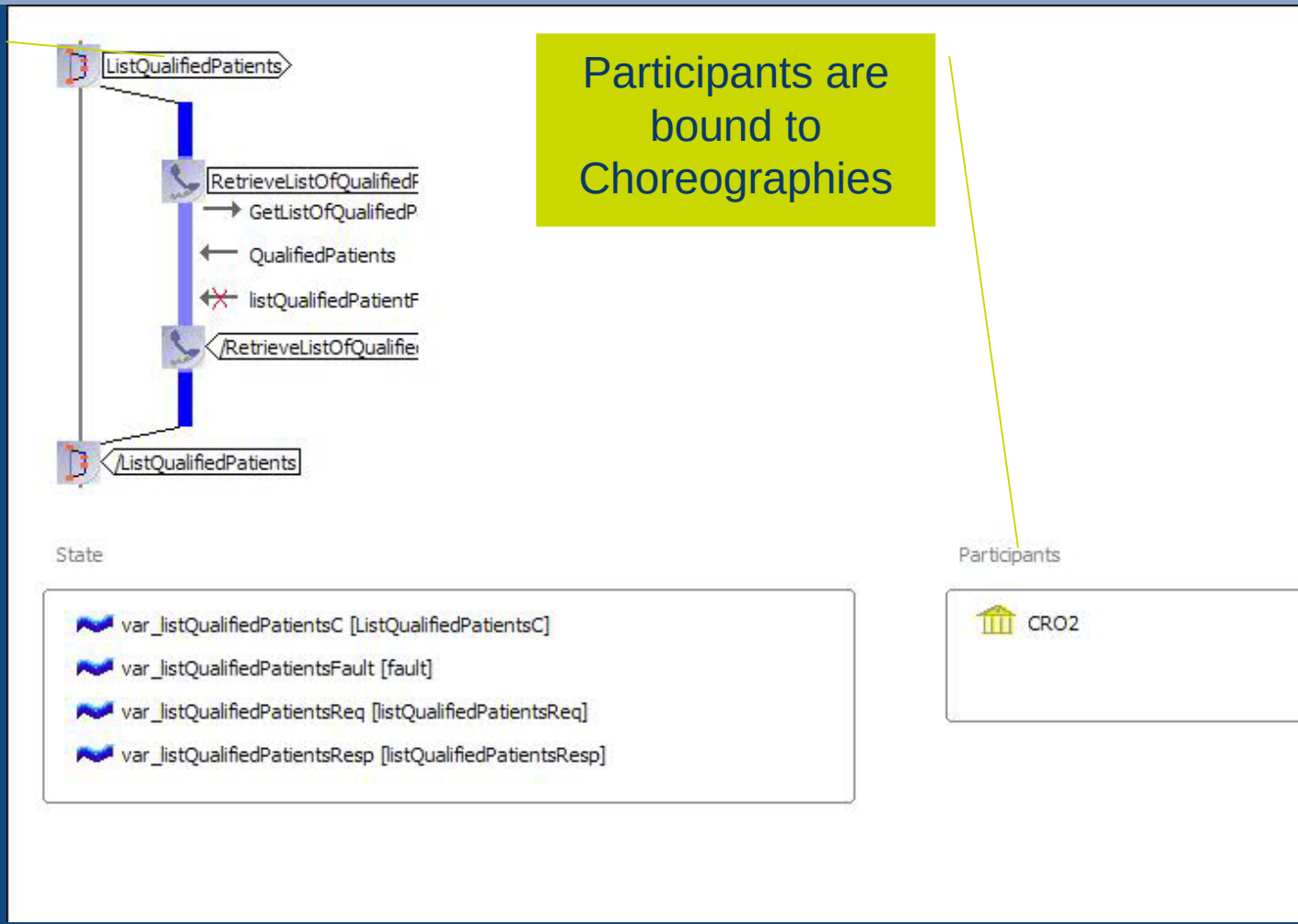
Domain-level Information types



Images and Models made using the pi4SOA tool suite (http://sourceforge.net/projects/pi4soa)

Choreographies – Logical Components and / or Realizations

Logical Choreography, including Interaction sequences, message exchanges, and faults



Images and Models made using the pi4SOA tool suite (http://sourceforge.net/projects/pi4soa)

Current Usage: National Cancer Institute

caBIG™ and Clinical Trials Management Suite

- Multiple distributed project teams, work groups, users, implementors
- The need to tie the *Architecture / Design* to *Implementations*
- The need to manage complexity
- The need to define and express the behavioral semantic behind information and systems

Conclusions

- True interoperability within a distributed environment is complex, technology and innovation can only go so far….
- We have presented a SOA Methodology that surfaces the three semantic dimensions necessary to specify distributed system behavior: Information, Functional, Dynamic
- We have explored the Contractual Foundation of a SOA (Interoperability Contract Specification) and use of CDL
- We have stressed the importance of *Choreography*: can serve as central dynamic model for specifications, as well as provide concrete communication pathways among a cross-functional team

Questions?

- Alan Honey
  - *Enterprise Architect, Kaiser Permanente*
  - *Alan.p.honey@kp.org*
- John Koisch
  - *Enterprise Architect, National Cancer Institute*
  - Koisch_john@bah.com
- HSSP
  - *www.healthinterop.org*