



Color Modeling and SOA

Daniel Vacanti

David J. Anderson & Associates



Agenda

- Color Modeling Intro – 30 mins
- Extracting Services from a Model – 20 mins
- Q & A – 10 mins



An introduction to Modelling in Colour



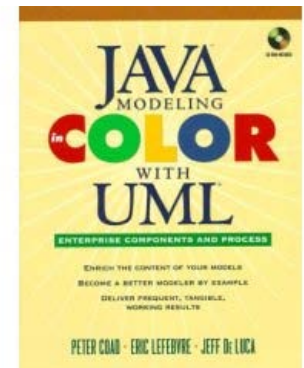
Why Colour Modeling?

- How do we get predictably repeatable *quality* models across a broad mix of developer ability
- What is a Quality Model?
 - Elegant
 - “Good Enough”
 - Extensible
 - Loosely Coupled
 - Robust



Peter Coad and the Archetypes

- Categories of roughly similar classes
- Not so similar that we can create a superclass/interface but similar enough to:
 - expect similar responsibilities
 - Properties – what I know myself
 - Operations – what questions I can answer, services I can provide
 - Associations – who I know, who I own, who I delegate to
 - participate in roughly similar patterns of
 - structure - in relation to classes in the same and other categories
 - behaviour – interactions, collaborations
- Four Class Archetypes
 - Coad, Lefebvre, De Luca, *Java Modeling in Color with UML*





Why are they called Archetypes?

Here's the concept we want to communicate:

A form or template for one of a small number of class categories. It specifies attributes, links, methods, plug-in points, and interactions that are typical for classes in that category.

Which is the better term for this concept?

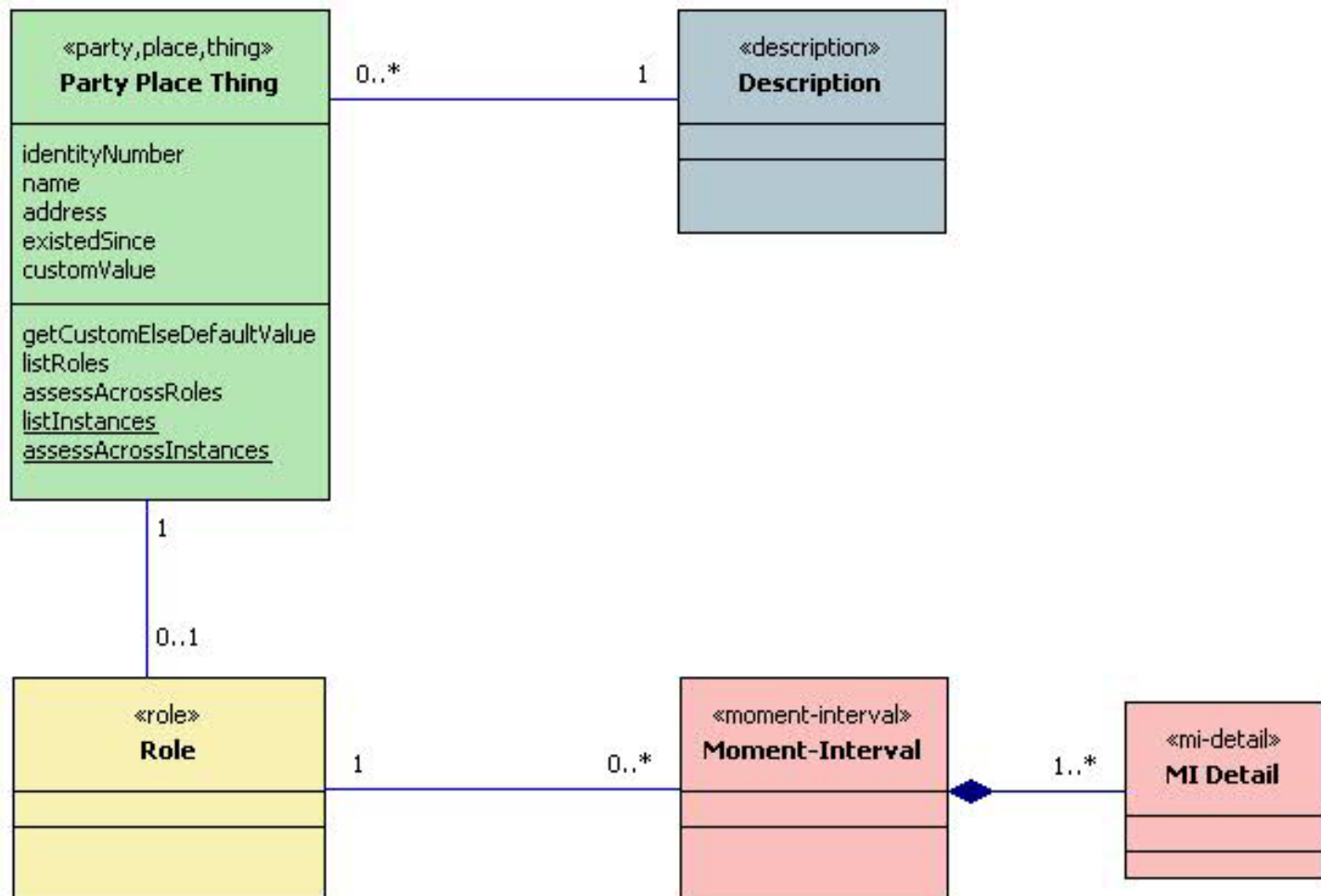
Stereotype

- 1. An unvarying model, as though cast from a mold*
- 2. A text tag for annotating a UML diagram element*
- 3. A broad categorization of classes*

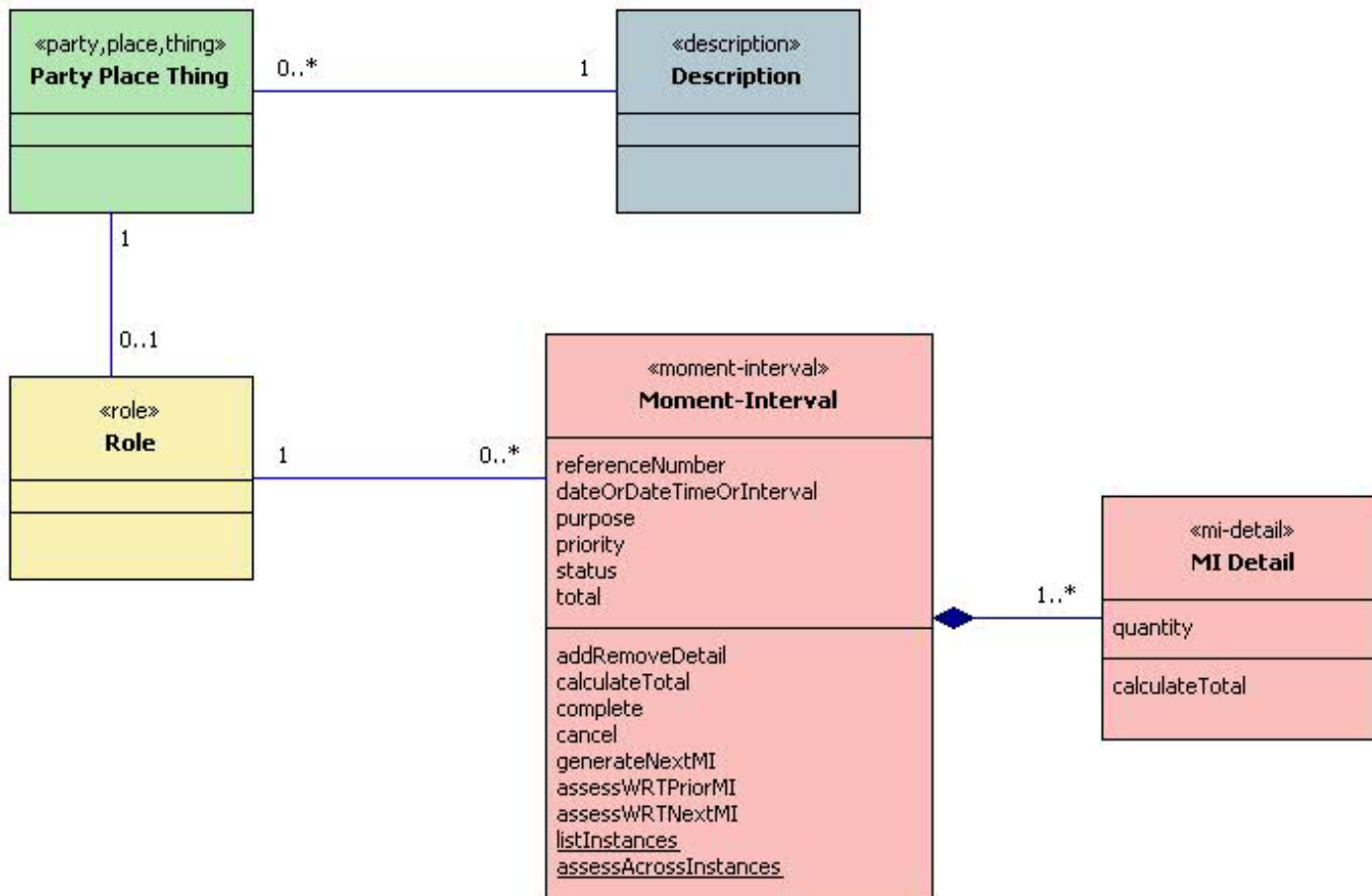
Archetype

A model from which all things of the same kind more or less follow.

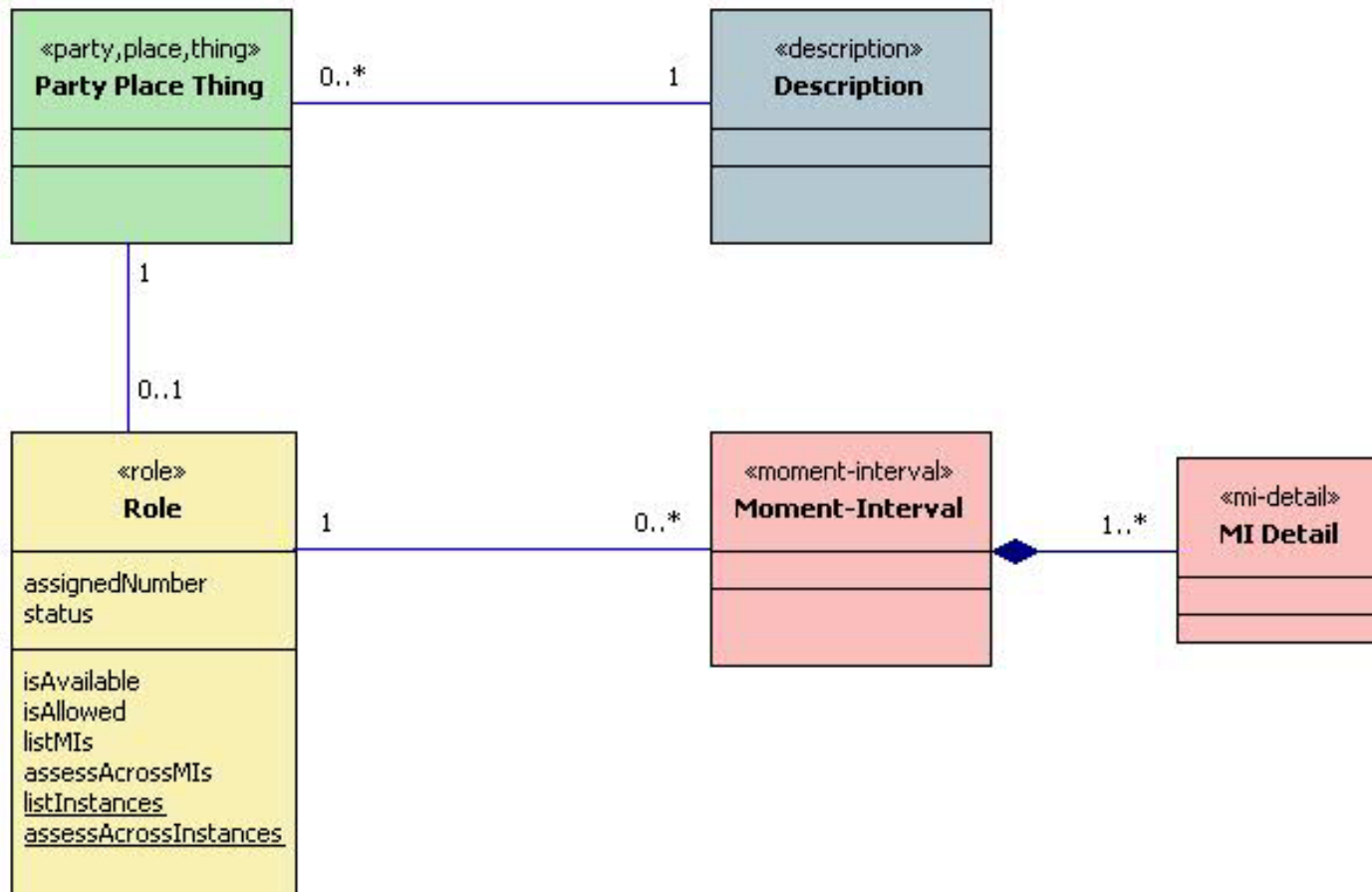
People, Places and Things



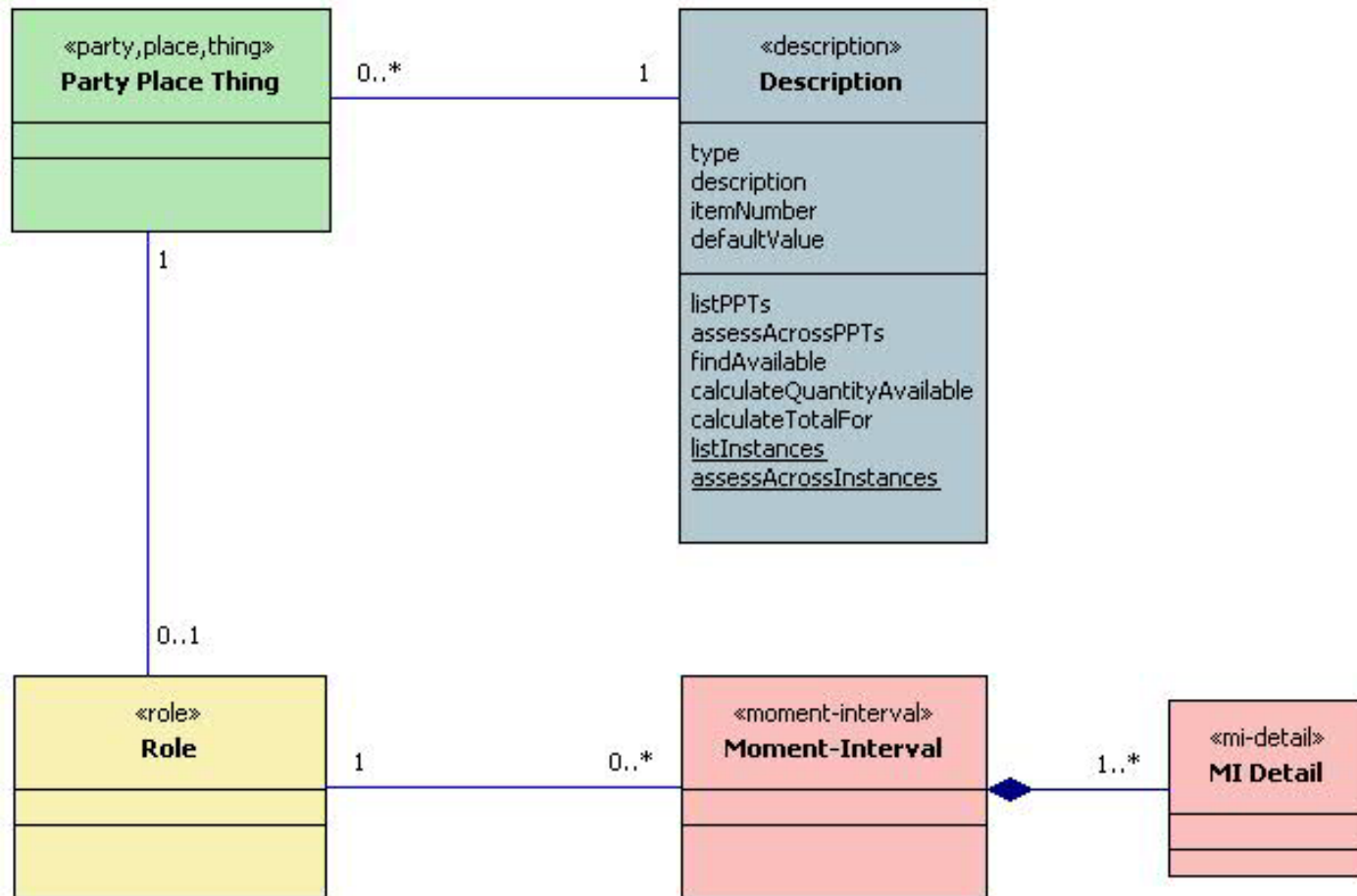
Moment-Intervals



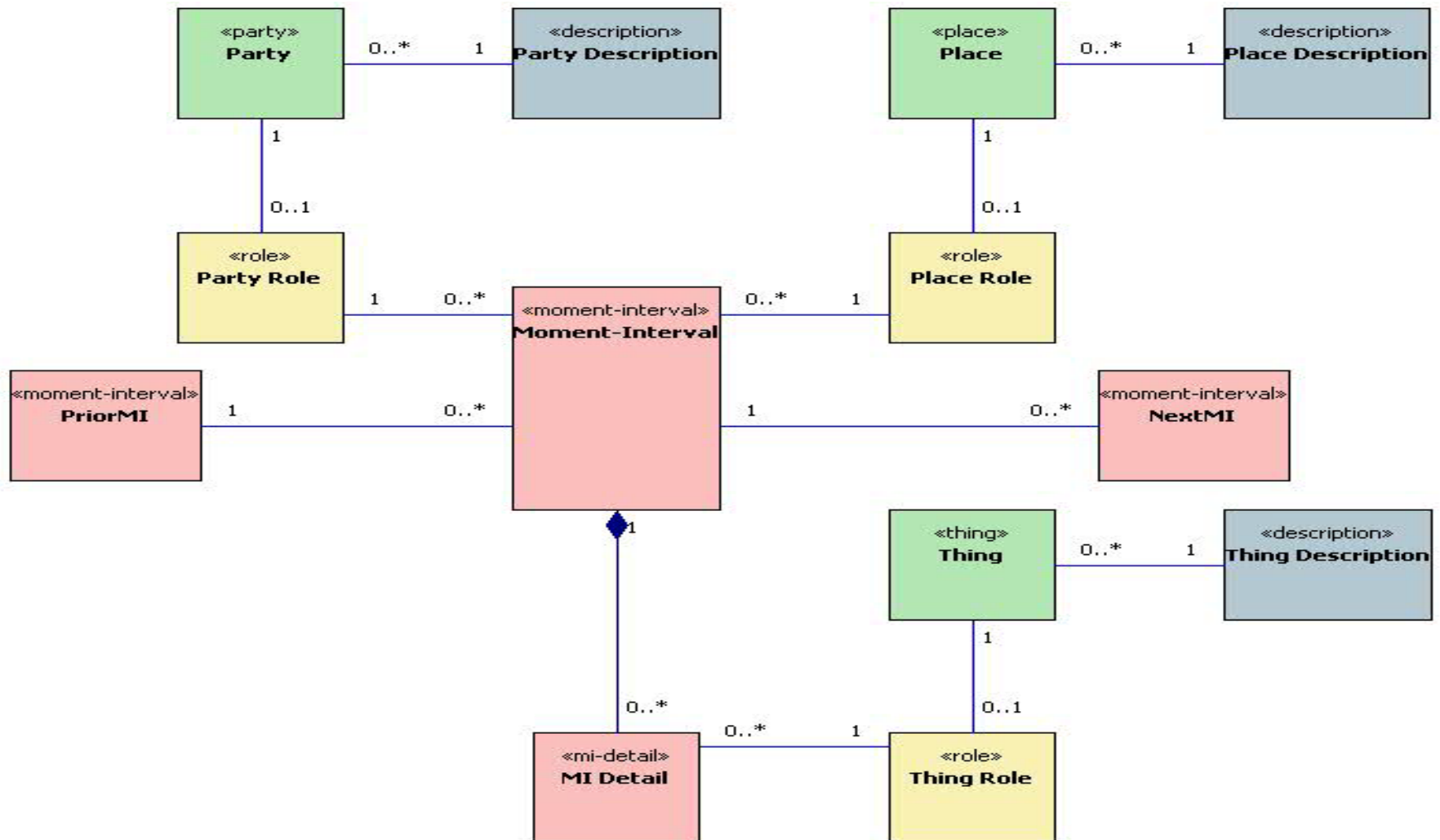
Roles



Catalogue-entry-like Descriptions



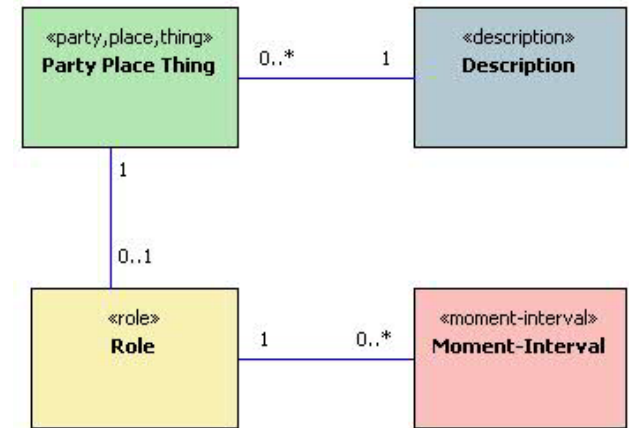
Domain Neutral Component



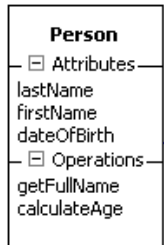


Colour

- New Layers of information
- Colour schemes
 - How many colours?
 - What colours?
 - Objections



Robust ...



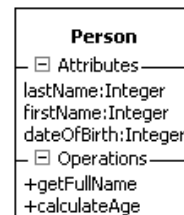
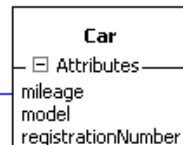
*

fares

*

+driver

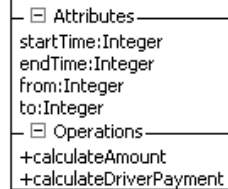
+taxi



*

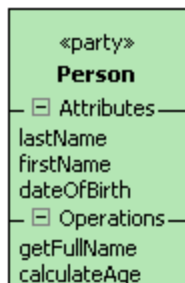
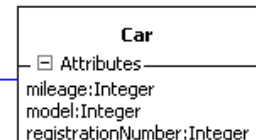
driver

Fare



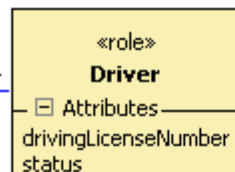
taxi

*



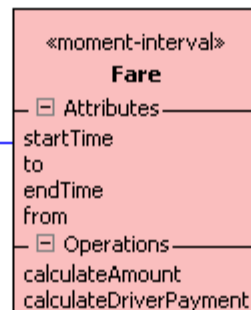
1

0..1



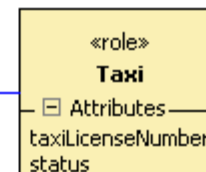
1

*



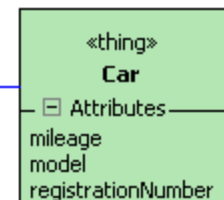
*

1



0..1

1





General Approach

- Do:
 - Go after pinks first
 - Fill out the rest of the DNC as prescribed by the pattern
- Don't:
 - Model first then color later
 - Argue about color



More reading...

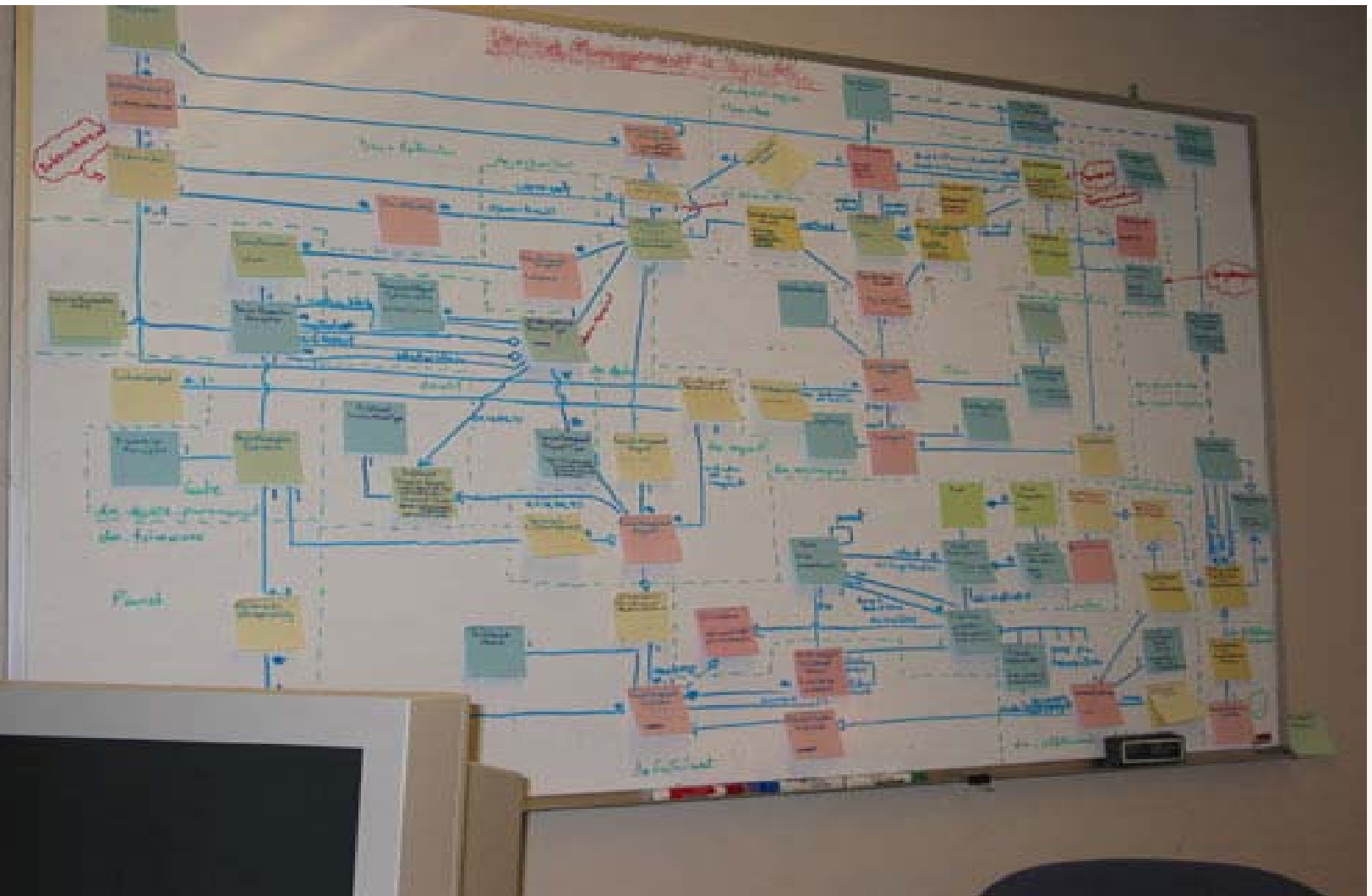
- Web articles introducing modelling in colour
 - knol.google.com/k/stephen-palmer
- Old Coad Letter issues
 - edn.embarcadero.com/search?q=Coad+color
- Compare and contrast:
 - Domain-Driven Design (domaindrivendesign.org)
 - Streamlined Object Modeling (www.streamlinedmodeling.com)

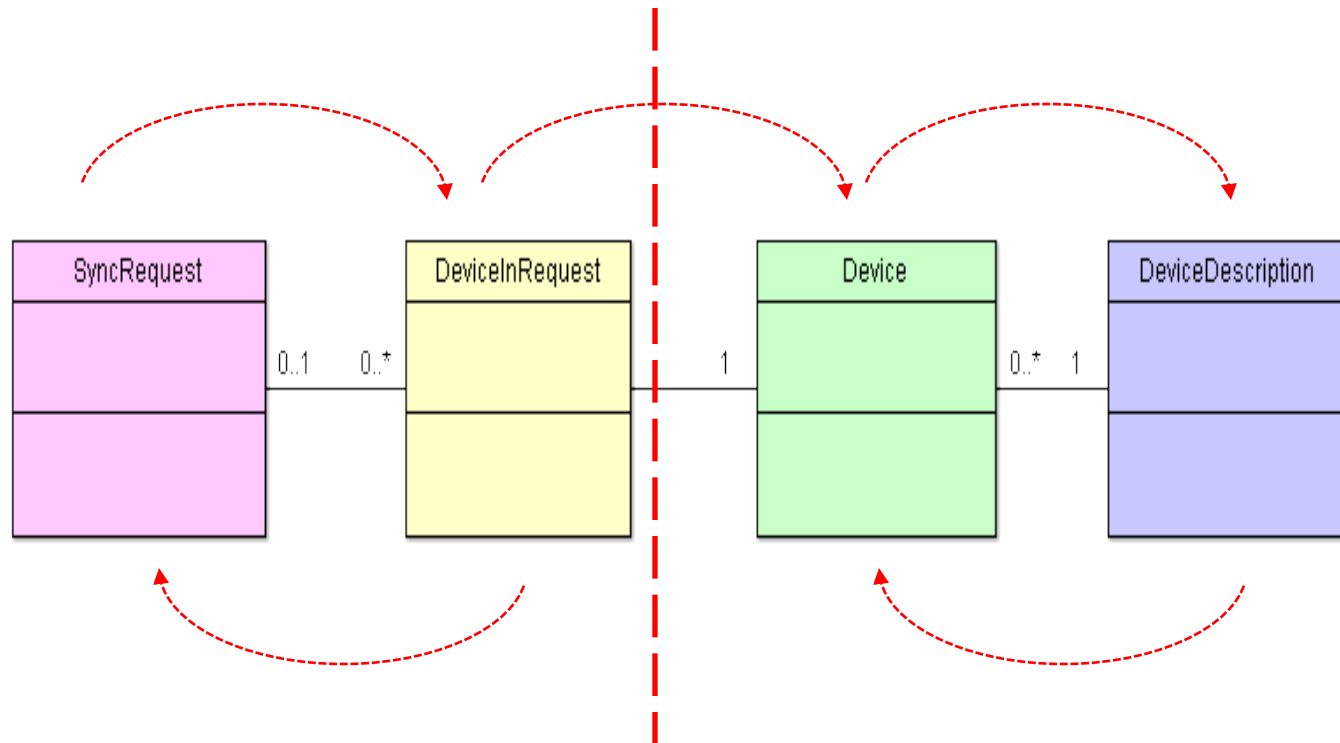


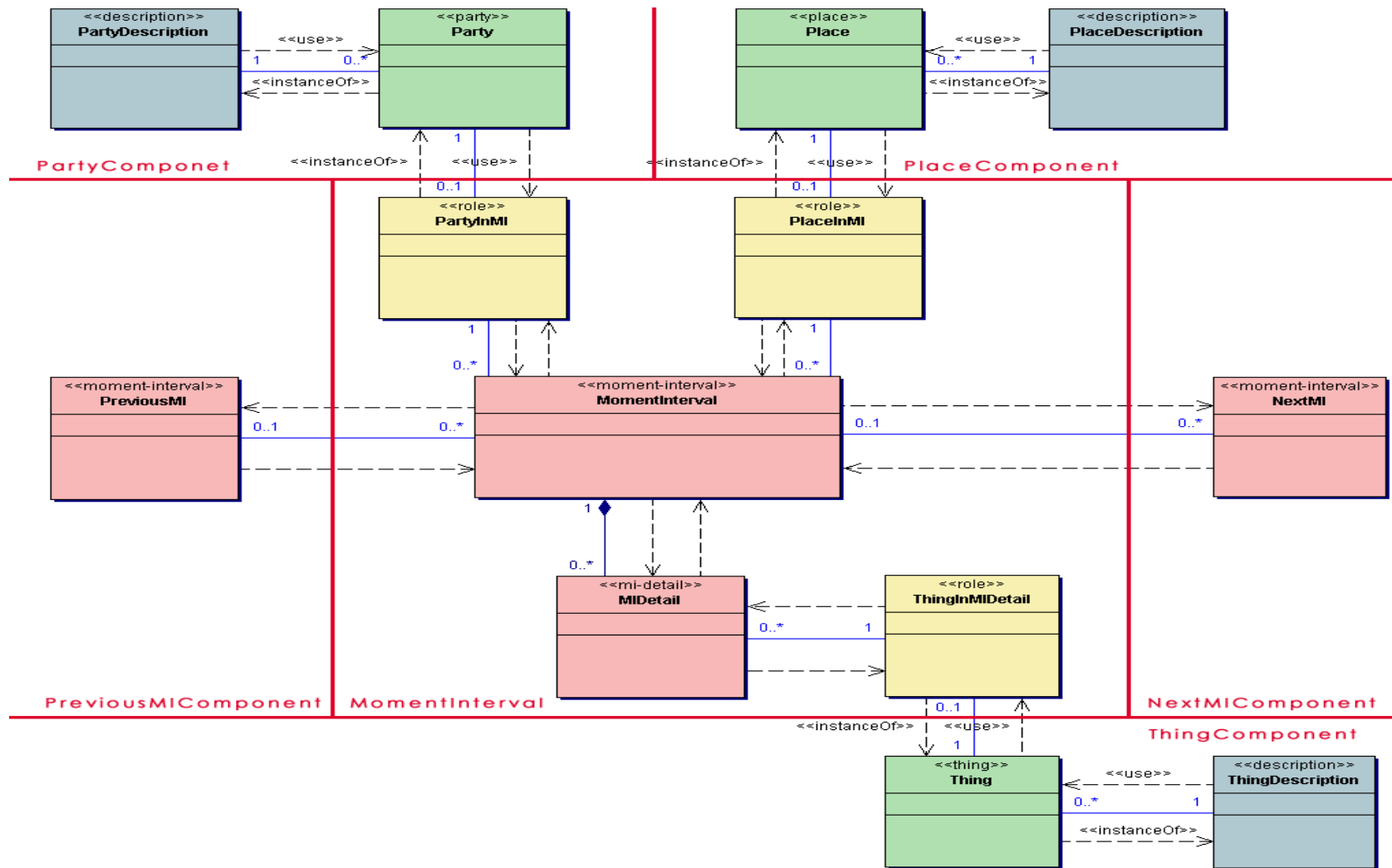
Extracting Services from a Model



A Brief History of...



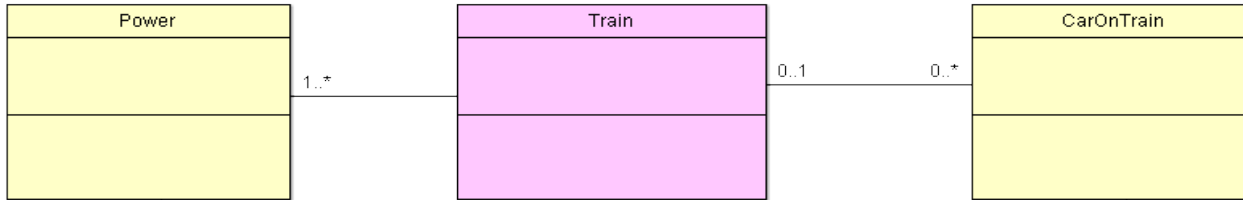




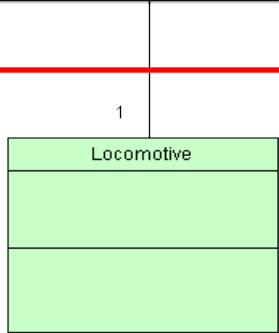


Train Example

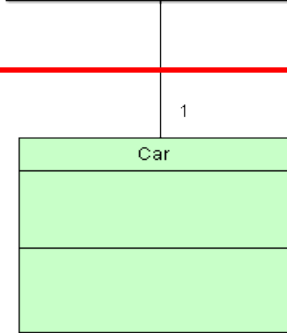
Train Service

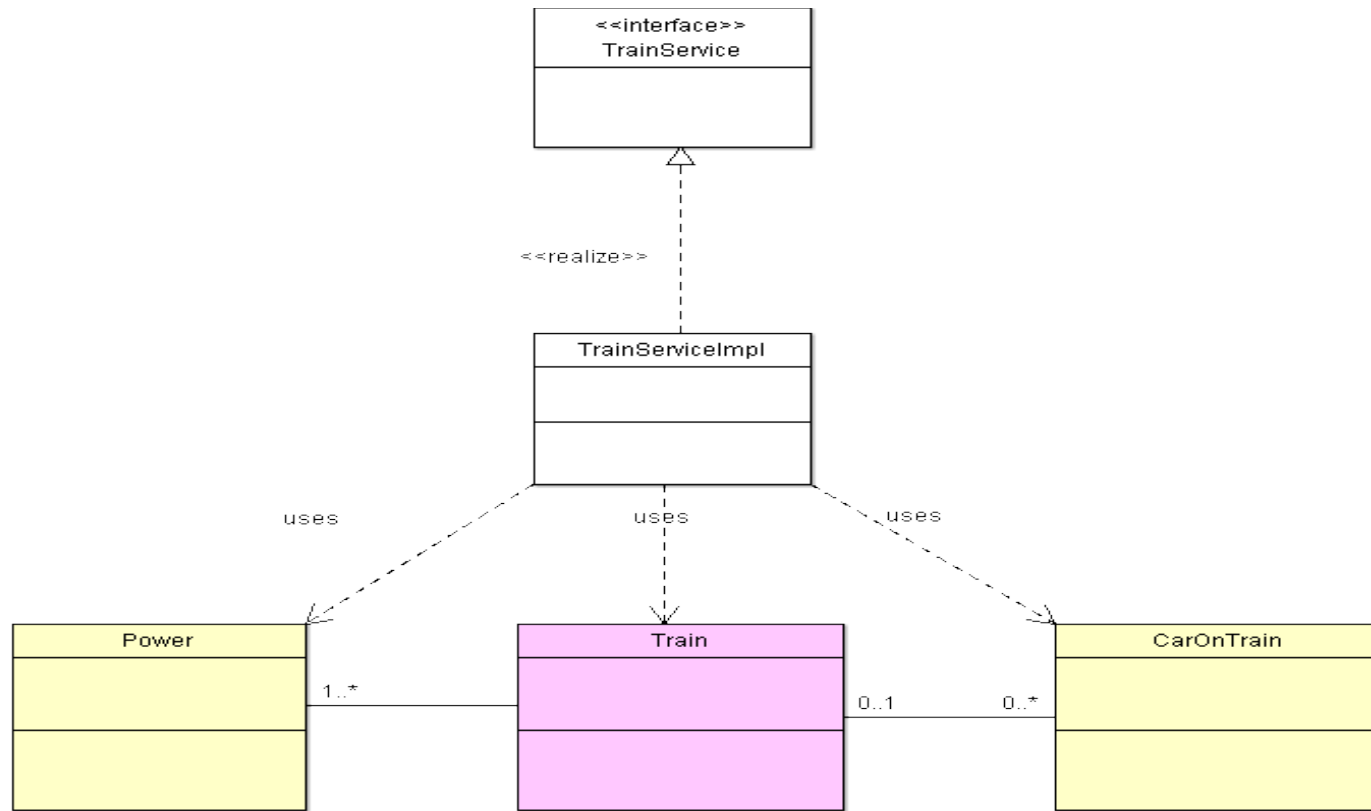


Locomotive Service



Car Service







Sample Schema Directory Structure

train-service

- get-cars-on-train
- get-power
- get-train
 - 1.0
 - get-train-request.xsd
 - get-train-reply.xsd



Sample Train Schema

```
<?xml version="1.0" encoding="UTF-8" ?>  
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified">
```

```
<xsd:include schemaLocation="power.xsd"/>
```

```
<xsd:include schemaLocation="car-on-train.xsd"/>
```

```
<xsd:complexType name="trainServiceType">  
  <xsd:sequence>  
    <xsd:element name="train-attributes" type="trainAttributesType" minOccurs="0" />  
    <xsd:element name="scheduled-train" type="scheduledTrainType" minOccurs="0" />  
    <xsd:element name="current-crew" type="crewType" minOccurs="0" />  
    <xsd:element name="air-brake-inspection-list" type="airBrakeInspectionListType" minOccurs="0" />  
    <xsd:element name="EOT-inspection-list" type="EOTInspectionListType" minOccurs="0" />  
    <xsd:element name="timetable-instructions" type="xsd:string" minOccurs="0" />  
    <xsd:element name="car-placement-error-list" type="carPlacementErrorListType" minOccurs="0" />  
    <xsd:element name="geographic-placement-error-list" type="geographicPlacementErrorListType" minOccurs="0" />  
    <xsd:element name="coupler-limit-list" type="couplerLimitListType" minOccurs="0" />  
    <xsd:element name="front-train-unit-list" type="trainUnitListType" minOccurs="0" />  
    <xsd:element name="EOT-list" type="trainUnitListType" minOccurs="0" />  
    <xsd:element name="bad-order-car-list" type="badOrderCarListType" minOccurs="0" />  
    <xsd:element name="power-consist" type="powerConsistType" minOccurs="0" />  
    <xsd:element name="equivalent-powered-axles" type="equivalentPoweredAxlesType" minOccurs="0" />  
    <xsd:element name="equivalent-dynamic-brake-axles" type="equivalentDynamicBrakeAxlesType" minOccurs="0" />  
    <xsd:element name="completed-train-event-list" type="trainEventListType" minOccurs="0" />  
    <xsd:element name="scheduled-train-event-list" type="trainEventListType" minOccurs="0" />  
    <xsd:element name="train-activity" type="trainActivityType" minOccurs="0" />  
    <xsd:element name="train-block-list" type="trainBlockListType" minOccurs="0" />  
    <xsd:element name="hazardous-instructions" type="xsd:string" minOccurs="0" />  
    <xsd:element name="high-value-loads-message" type="xsd:string" minOccurs="0" />  
  </xsd:sequence>  
  <xsd:attribute name="symbol" type="xsd:string" use="required" />  
  <xsd:attribute name="day" type="xsd:unsignedByte" use="optional" />  
</xsd:complexType>
```



Sample Train Reply Schema

```
<?xml version="1.0" encoding="UTF-8" ?>
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  targetNamespace="http://services.www.up.com/train/get-train/1.0"
  xmlns:tns="http://services.www.up.com/train/get-train/1.0"
  elementFormDefault="qualified">

  <xsd:include schemaLocation="../../train-1.0.xsd" />

  <xsd:element name="get-train-reply" type="tns:getTrainReplyRootType" />

  <xsd:complexType name="getTrainReplyRootType">
    <xsd:sequence>
      <xsd:element name="train-reply-list" type="tns:trainReplyListType" maxOccurs="unbounded"/>
    </xsd:sequence>
  </xsd:complexType>

  <xsd:complexType name="trainReplyListType">
    <xsd:sequence>
      <xsd:element name="train-reply-by-id" type="tns:trainReplyByIdType" />
    </xsd:sequence>
  </xsd:complexType>

  <xsd:complexType name="trainReplyByIdType">
    <xsd:sequence>
      <xsd:element name="train" type="tns:trainServiceType" />
    </xsd:sequence>
  </xsd:complexType>

</xsd:schema>
```



Other topics for discussion

- Schema Design
 - Appropriate use of namespaces
 - Xsd Import vs. Xsd Include
- Service Lifecycle
 - Versioning
 - Independent deployment
 - Deprecation/deletion
- Database Design



Disclosure

- Not Perfect—but good enough
 - Quickly overcome project inertia
 - Immediately identify candidate services
- Tooling still weak
- Successfully used across multiple technologies
 - Java: POJOs/EJBs
 - Web Services (WSDL)
 - .NET C#/WCF