

Architecture and Design of Adaptive Object-Models

Joseph W. Yoder
Federico Balaguer
Ralph Johnson

University of Illinois

The Refactory, Inc.

yoder@refactory.com

[balaguer|johnson]@cs.uiuc.edu

<http://www.adaptiveobjectmodel.com>

Metadata and Adaptive Object-Models

"Anything you can do, I can do Meta"

Metadata: If something is going to vary in a predictable way, store the *description* of the variation in a database so that it is easy to change....Ralph Johnson

"Meta is Beta"

General Problem

- Requirements change within applications' domain.
- Business Rules are changing rapidly.
- Applications have to quickly adapt to new business requirements.
- Changing the application is costly, it generally includes code and data-storage.
- There are cycles of: build-compile-release.

Architecture and Design of Adaptive Object-Models – Intriguing Technology Session, OOPSLA'2001 – October 2001, Tampa FL. Copyright 2001, The Refactory, Inc.

General Solution

- Create an object design (meta-model) that describes the domain objects which includes attributes, relationships, and business rules as instances rather than classes.
- The domain objects are instantiated through a description given by the user or domain expert.
- Each new requirement is satisfied by creating a new description and a new instantiation.

Architecture and Design of Adaptive Object-Models – Intriguing Technology Session, OOPSLA'2001 – October 2001, Tampa FL. Copyright 2001, The Refactory, Inc.

Adaptive Object-Model

(Dynamic Object-Model)

- An ADAPTIVE OBJECT-MODEL is an object model that provides “meta” information about the domain so that it can be changed at runtime
 - | explicit object model that it interprets at run-time
 - | change the object model, system changes its behavior
- ADAPTIVE OBJECT-MODELS usually arise as domain-specific frameworks
- Business rules can be stored in ADAPTIVE OBJECT-MODELS

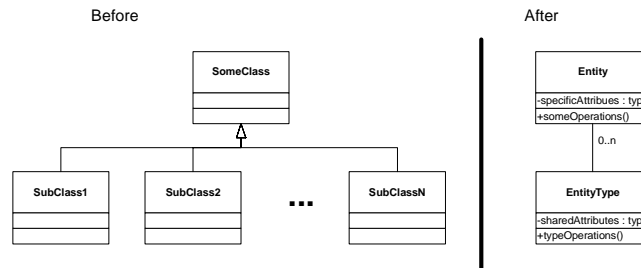
Architecture and Design of Adaptive Object-Models – Intriguing Technology Session, OOPSLA'2001 – October 2001, Tampa FL. Copyright 2001, The Refactory, Inc.

Architectural Elements of AOM

- Metadata
- TypeObject
- Properties
- Type Square
- Entity-Relationship
- Strategy/RuleObjects
- Interpreters/Builders
- Editors/GUIs

Architecture and Design of Adaptive Object-Models – Intriguing Technology Session, OOPSLA'2001 – October 2001, Tampa FL. Copyright 2001, The Refactory, Inc.

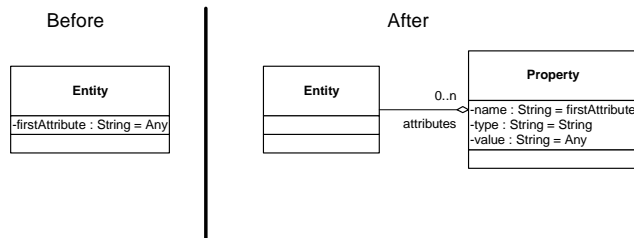
Type-Object



PLoPD3 - Johnson and Woolf

Architecture and Design of Adaptive Object-Models – Intriguing Technology Session, OOPSLA'2001 – October 2001, Tampa FL. Copyright 2001, The Refactory, Inc.

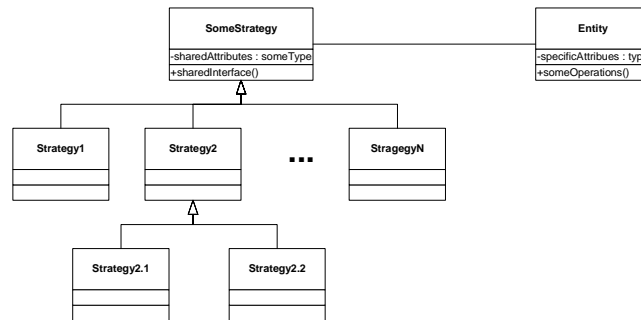
Properties



PLoP98 - Foote and Yoder

Architecture and Design of Adaptive Object-Models – Intriguing Technology Session, OOPSLA'2001 – October 2001, Tampa FL. Copyright 2001, The Refactory, Inc.

Strategies

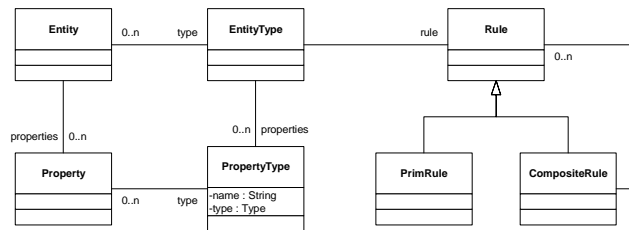


Design Patterns - GOF95

Architecture and Design of Adaptive Object-Models – Intriguing Technology Session, OOPSLA'2001 – October 2001, Tampa FL. Copyright 2001, The Refactory, Inc.

Adaptive Object-Model (Very Common Structure)

ECOOP & OOPSLA 2001 Yoder, Balaguer, Johnson



Type Square

Architecture and Design of Adaptive Object-Models – Intriguing Technology Session, OOPSLA'2001 – October 2001, Tampa FL. Copyright 2001, The Refactory, Inc.

Adaptive Object-Model

(Advantages)

- Can more easily adapt to new business requirements
- Smaller in terms of classes so possibly easier to maintain by experts
- Time to market can be reduced
- More closely

Architecture and Design of Adaptive Object-Models – Intriguing Technology Session, OOPSLA'2001 – October 2001, Tampa FL. Copyright 2001, The Refactory, Inc.

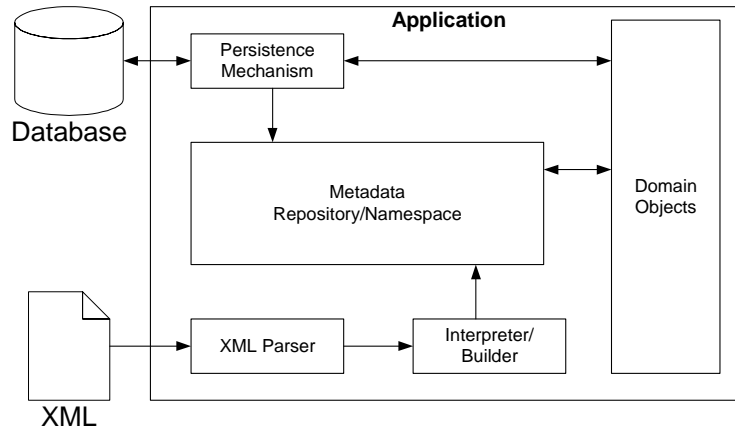
Adaptive Object-Model

(Disadvantages)

- Can be hard to understand and maintain
- Can have higher start up costs
- Can have poor performance
- Requires support tools

Architecture and Design of Adaptive Object-Models – Intriguing Technology Session, OOPSLA'2001 – October 2001, Tampa FL. Copyright 2001, The Refactory, Inc.

Interpreters / Builders: Solution



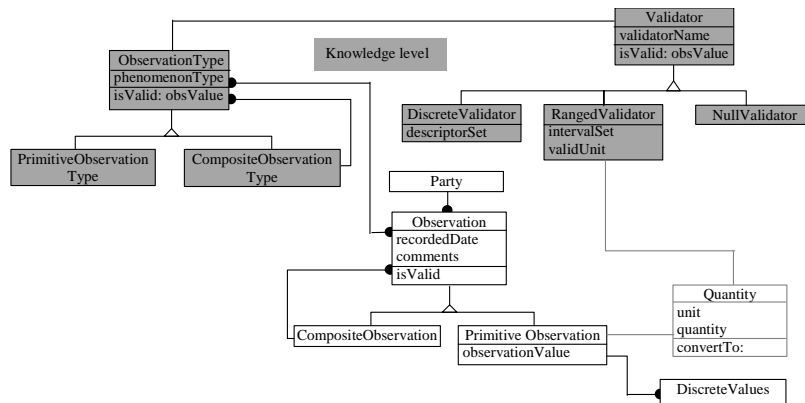
Architecture and Design of Adaptive Object-Models – Intriguing Technology Session, OOPSLA'2001 – October 2001, Tampa FL. Copyright 2001, The Refactory, Inc.

Medical Observations



Architecture and Design of Adaptive Object-Models – Intriguing Technology Session, OOPSLA'2001 – October 2001, Tampa FL. Copyright 2001, The Refactory, Inc.

Observation Example



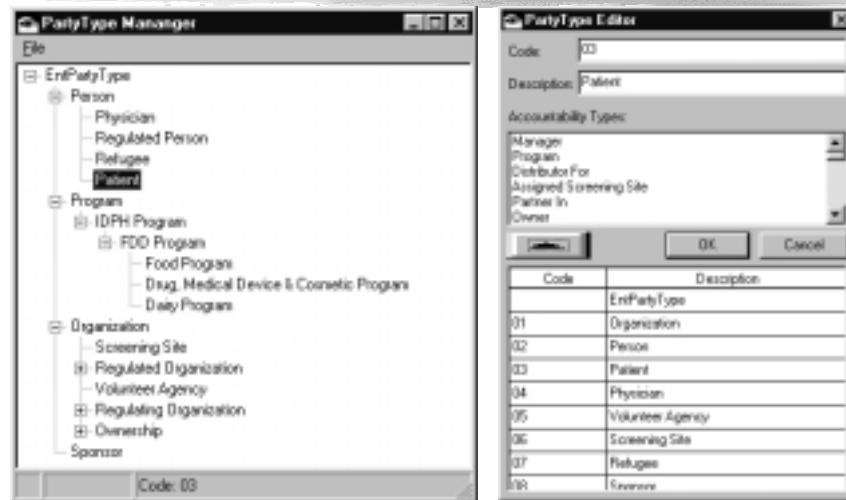
Architecture and Design of Adaptive Object-Models – Intriguing Technology Session, OOPSLA'2001 – October 2001, Tampa FL. Copyright 2001, The Refactory, Inc.

Metamodel and GUI

- The metadata can simplify building user interfaces. Special GUI components can be developed for using the metadata.
- Example: The Observation model includes widgets that display list of values from the DiscreteValidators and also EntryBoxes that use RangeValidator.
- A Mediator and Adaptor layer was developed for managing the interactions between the domain objects and the GUIs.

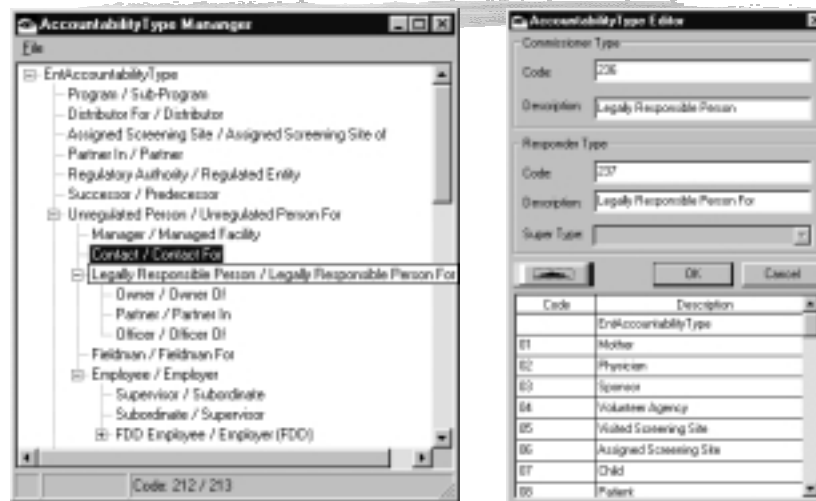
Architecture and Design of Adaptive Object-Models – Intriguing Technology Session, OOPSLA'2001 – October 2001, Tampa FL. Copyright 2001, The Refactory, Inc.

PartyType: Metadata-Editors



Architecture and Design of Adaptive Object-Models – Intriguing Technology Session, OOPSLA'2001 – October 2001, Tampa FL. Copyright 2001, The Refactory, Inc.

Accountability: Metadata-Editors



Architecture and Design of Adaptive Object-Models – Intriguing Technology Session, OOPSLA'2001 – October 2001, Tampa FL. Copyright 2001, The Refactory, Inc.

Observation: Metadata-Editors

Phenomenon	Type	validatorName	validatorType
17-Hydroxycoprosterol	Composite	DefaultValidator	
17OHHPDTY	Primitive	17OHHPDTY	Ranged
17OHPRELST	Primitive	POSNEGBOFDER	Discrete
ABDOMEN	Primitive	CONDITIONTYPE	Discrete
ALCOHOLABUSE	Primitive	BOOLEAN	Discrete
ALLERGIES	Primitive	BOOLEAN	Discrete
ANTRHIC	Primitive	BOOLEAN	Discrete
ARTHRITIS	Primitive	BOOLEAN	Discrete
ASTHMA	Primitive	BOOLEAN	Discrete
Bloodbase	Primitive	POSNEG	Discrete
BLOODPRESSURE	Composite	DefaultValidator	
CANCER	Primitive	BOOLEAN	Discrete
CBC	Primitive	CONDITIONTYPE	Discrete
CHANCESID	Primitive	BOOLEAN	Discrete
CHESTXRAYRESULT	Primitive	CONDITIONTYPE	Discrete
CHRONICALCOHOL	Primitive	BOOLEAN	Discrete
DIABETES	Primitive	BOOLEAN	Discrete
DIASTOLIC	Primitive	DIASTOLIC	Ranged

Architecture and Design of Adaptive Object-Models – Intriguing Technology Session, OOPSLA'2001 – October 2001, Tampa FL. Copyright 2001, The Refactory, Inc.

Observation: Metadata-Editors

Primitive Observation Type Editor

Phenomenon: FeedingType

Validator: IntegerValidator [Edit]

[Save] [Done]

Composite Observation Type Editor

Phenomenon: Galactosemia

Validation Policy: NullValidator [Edit]

Available... Included...

- AGE-RangedValidator
- FEEDING-DiscreteValidator
- HEG-NullValidator
- TESTRESULT-NullValidator
- TECH-NullValidator
- WEIGHT-NullValidator
- GAL-NullValidator
- PHC-NullValidator
- PHI-DiscreteValidator

[OK] [Close] [Cancel]

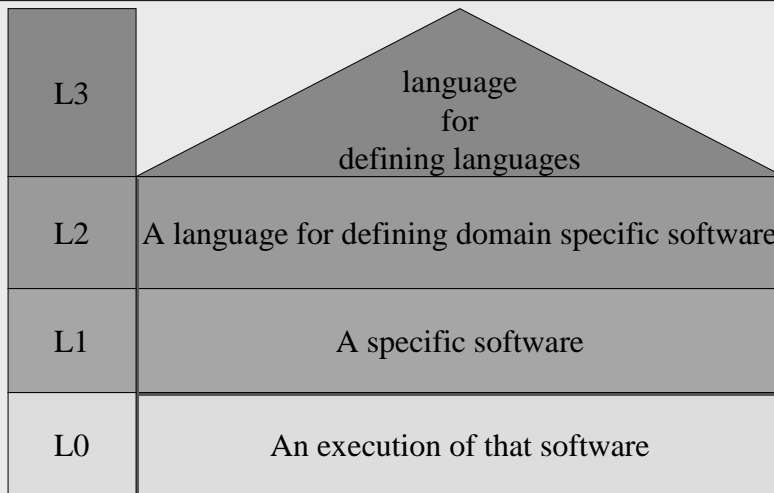
Architecture and Design of Adaptive Object-Models – Intriguing Technology Session, OOPSLA'2001 – October 2001, Tampa FL. Copyright 2001, The Refactory, Inc.

Observation: Metadata-Editors



Architecture and Design of Adaptive Object-Models – Intriguing Technology Session, OOPSLA'2001 – October 2001, Tampa FL. Copyright 2001, The Refactory, Inc.

MOF



Dimensions of abstraction in Adaptive Object-Models, Reflection and OMG 's metamodeling Architecture

Copyright by ECOOP' 2000 workshop on Adaptive Object-Model. --- <http://www.joeyoder.com/Research/metadata/ECOOP2000/description.html>

Adaptive Object-Model

■ When to Build AOMs

- Need for flexibility
- High pace of business change
- Need for experimentation
- Need to empower user

Architecture and Design of Adaptive Object-Models – Intriguing Technology Session, OOPSLA'2001 – October 2001, Tampa FL. Copyright 2001, The Refactory, Inc.

Advantages of AOM

- Systems can more easily be adapted to domain changes.
- Changes do not require recompiling the system.
- Power Users can change the business rules.
- Shorter time-to-market.
- Smaller in terms of classes so can be easier to maintain by experts.

Architecture and Design of Adaptive Object-Models – Intriguing Technology Session, OOPSLA'2001 – October 2001, Tampa FL. Copyright 2001, The Refactory, Inc.

Disadvantages of AOM

- Developing AOM is expensive.
(higher startup costs)
- Can be hard to understand and maintain.
(user-model and meta-model)
- It requires skilled human resources.
- Can have poor performance.
- It demands having infrastructure for storing, building, interpreting metadata (special support tools, editors, etc).

Architecture and Design of Adaptive Object-Models – Intriguing Technology Session, OOPSLA'2001 – October 2001, Tampa FL. Copyright 2001, The Refactory, Inc.

Other Approaches and Technologies

- Black-box Frameworks
- Code Generators
- Metamodeling Techniques
- Table-driven Systems
- Generative Techniques

Architecture and Design of Adaptive Object-Models – Intriguing Technology Session, OOPSLA'2001 – October 2001, Tampa FL. Copyright 2001, The Refactory, Inc.