

URDAD as Quality-Driven Process ¹

Fritz Solms, Stefan Gruner and Cuen Edwards

URDAD-MDE subgroup of SSFM
Department of Computer Science
University Of Pretoria

fritz@solms.co.za

June 9, 2011

¹Accepted as a regular paper at SOMET 2011, the 10 International Conference on Intelligent Software Methodologies, Tools, and Techniques, St Petersburg, 28 - 30 Sept 2011

Problem Specification

URDAD as
Quality-Driven
Process

Solms, Gruner,
Edwards

- Inferior requirements
 - Core contributor to poor software quality & high cost.

Problem
Specification

Definitions

Abstract

URDAD

Quality drivers
embedded in
URDAD

Internal
consistency

Summary

Problem Specification

URDAD as
Quality-Driven
Process

Solms, Gruner,
Edwards

Problem
Specification

Definitions

Abstract

URDAD

Quality drivers
embedded in
URDAD

Internal
consistency

Summary

- Inferior requirements
 - Core contributor to poor software quality & high cost.
- Formal methods
 - Use mathematical modeling & formal logic to specify & verify requirements.
 - Incur high cost & skills requirements.

Problem Specification

URDAD as
Quality-Driven
Process

Solms, Gruner,
Edwards

Problem
Specification

Definitions

Abstract

URDAD

Quality drivers
embedded in
URDAD

Internal
consistency

Summary

- Inferior requirements
 - Core contributor to poor software quality & high cost.
- Formal methods
 - Use mathematical modeling & formal logic to specify & verify requirements.
 - Incur high cost & skills requirements.
- Semi-formal methods
 - Constrain cost & skills requirements.
 - Degree of formalization of process & inputs/outputs.

Problem Specification

URDAD as
Quality-Driven
Process

Solms, Gruner,
Edwards

Problem
Specification

Definitions

Abstract

URDAD

Quality drivers
embedded in
URDAD

Internal
consistency

Summary

- Inferior requirements
 - Core contributor to poor software quality & high cost.
- Formal methods
 - Use mathematical modeling & formal logic to specify & verify requirements.
 - Incur high cost & skills requirements.
- Semi-formal methods
 - Constrain cost & skills requirements.
 - Degree of formalization of process & inputs/outputs.
- Model Driven Engineering (MDE)
 - Fall into class of semi-formal methods
 - Often no defined engineering process.
 - Ad-hoc processes often inefficient and non-predictable (estimation)
 - Design structures, quality & semantics often vary.
 - Increases cost & complexity of model validation, code generation, documentation generation, ...

Definitions

URDAD as
Quality-Driven
Process

Solms, Gruner,
Edwards

Definition

Quality is the degree to which a set of inherent characteristics fulfills requirements.²

Problem
Specification

Definitions

Abstract

URDAD

Quality drivers
embedded in
URDAD

Internal
consistency

Summary

²David Hoyle, *ISO 9000: 2000 Quality Systems Handbook*. 4th ed, 2000.

³P. G Petersen, et al., Software quality drivers and indicators. *System Sciences*, p210 –218 vol.2, 1989.

Definitions

URDAD as
Quality-Driven
Process

Solms, Gruner,
Edwards

Problem
Specification

Definitions

Abstract

URDAD

Quality drivers
embedded in
URDAD

Internal
consistency

Summary

Definition

Quality is the degree to which a set of inherent characteristics fulfills requirements.²

Definition

A *quality criterion* is an observable quality characteristic of the solution.

²David Hoyle, *ISO 9000: 2000 Quality Systems Handbook*. 4th ed, 2000.

³P. G Petersen, et al., Software quality drivers and indicators. *System Sciences*, p210 –218 vol.2, 1989.

Definitions

URDAD as
Quality-Driven
Process

Solms, Gruner,
Edwards

Definition

Quality is the degree to which a set of inherent characteristics fulfills requirements.²

Definition

A *quality criterion* is an observable quality characteristic of the solution.

Definition

A *quality measure* is a quantitative metric for a quality criterion.

²David Hoyle, *ISO 9000: 2000 Quality Systems Handbook*. 4th ed, 2000.

³P. G Petersen, et al., Software quality drivers and indicators. *System Sciences*, p210 –218 vol.2, 1989.

Definitions

URDAD as
Quality-Driven
Process

Solms, Gruner,
Edwards

Definition

Quality is the degree to which a set of inherent characteristics fulfills requirements.²

Definition

A *quality criterion* is an observable quality characteristic of the solution.

Definition

A *quality measure* is a quantitative metric for a quality criterion.

Definition

A *quality driver* is an activity which improves one or more process or model quality criteria.³

²David Hoyle, *ISO 9000: 2000 Quality Systems Handbook*. 4th ed, 2000.

³P. G Petersen, et al., Software quality drivers and indicators. *System Sciences*, p210 –218 vol.2, 1989.

Problem
Specification

Definitions

Abstract

URDAD

Quality drivers
embedded in
URDAD

Internal
consistency

Summary

Abstract

URDAD as
Quality-Driven
Process

Solms, Gruner,
Edwards

Problem
Specification

Definitions

Abstract

URDAD

Quality drivers
embedded in
URDAD

Internal
consistency

Summary

- URDAD is a semi-formal, service-oriented A&D methodology.
 - Generates technology neutral requirements model (PIM).
 - Methodology supported by metamodel & DSL.

Abstract

URDAD as
Quality-Driven
Process

Solms, Gruner,
Edwards

Problem
Specification

Definitions

Abstract

URDAD

Quality drivers
embedded in
URDAD

Internal
consistency

Summary

- URDAD is a semi-formal, service-oriented A&D methodology.
 - Generates technology neutral requirements model (PIM).
 - Methodology supported by metamodel & DSL.
- Contributions of this paper
 - We identify for each quality criterion
 - Set of quality drivers.
 - Show quality drivers used in URDAD.

URDAD

URDAD as
Quality-Driven
Process

Solms, Gruner,
Edwards

Problem
Specification

Definitions

Abstract

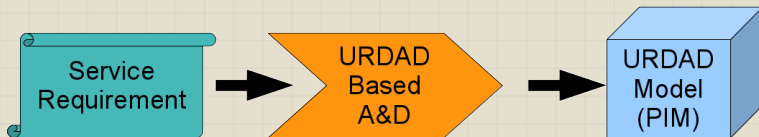
URDAD

Quality drivers
embedded in
URDAD

Internal
consistency

Summary

- Systematic methodology for technology-neutral A&D
 - service-oriented approach
 - generates MDA's PIM



URDAD as recursive analysis & design algorithm

URDAD as
Quality-Driven
Process

Solms, Gruner,
Edwards

Problem
Specification

Definitions

Abstract

URDAD

Quality drivers
embedded in
URDAD

Internal
consistency

Summary

```
1 class Urdad
2 {
3     provideService(serviceRequirement):Service
4     {
5         serviceContract = negotiateContract(serviceRequirement)
6
7         try
8         {
9             return serviceRegistry.getService(serviceContract)
10        }
11        catch (noRealizingServiceException)
12        {
13            service = designService(serviceContract)
14
15            for (lowerLevelServiceRequirement : service.requiredServices)
16                provideService(lowerLevelServiceRequirement)
17        }
18    }
19 }
```

URDAD analysis phase

URDAD as
Quality-Driven
Process

Solms, Gruner,
Edwards

Problem
Specification

Definitions

Abstract

URDAD

Quality drivers
embedded in
URDAD

Internal
consistency

Summary

```
1 class UrdadAnalysis
2 {
3     negotiateContract(serviceRequirement):ServiceContract
4     {
5         for (stakeholder:identifyStakeHolders(serviceRequirement))
6         {
7             functionalRequirements = sourceFunctionalRequirements(
2             stakeholder, serviceRequirement)
8             qualityRequirements = sourceFunctionalRequirements(stakeholder,
2             serviceRequirement)
9         }
10        negotiateConsistentRequirements()
11        groupFunctionalRequirementsIntoServiceRequirements(
2            functionalRequirements)
12        for (functionalRequirement:functionalRequirements)
13            defineCondition(functionalRequirement)
14            // includes test & associated exception
15        specifyDatastructuresForRequestAndResultClasses()
16        assembleServiceContract()
17        assignServiceContractToResponsibilityDomain()
18        return serviceContract
19    }
20 }
```

Semantic quality

Model quality impacted by quality of modeling language.

- Define semantics via metamodel or ontology.

URDAD as
Quality-Driven
Process

Solms, Gruner,
Edwards

Problem
Specification

Definitions

Abstract

URDAD

Quality drivers
embedded in
URDAD

Internal
consistency

Summary

Semantic quality

Model quality impacted by quality of modeling language.

- Define semantics via metamodel or ontology.

Qualities of modeling language:

- *Completeness*

- Formal lang: power to express statements needed for URDAD.
 - All meaning to be conveyed can be conveyed.
- Informally verified through
 - Analyze URDAD process & models for required semantics.
 - Empirically tested via example models.

URDAD as
Quality-Driven
Process

Solms, Gruner,
Edwards

Problem
Specification

Definitions

Abstract

URDAD

Quality drivers
embedded in
URDAD

Internal
consistency

Summary

Semantic quality

URDAD as
Quality-Driven
Process

Solms, Gruner,
Edwards

Model quality impacted by quality of modeling language.

- Define semantics via metamodel or ontology.

Qualities of modeling language:

■ *Completeness*

- Formal lang: power to express statements needed for URDAD.
 - All meaning to be conveyed can be conveyed.
- Informally verified through
 - Analyze URDAD process & models for required semantics.
 - Empirically tested via example models.

■ *Consistency*

- Metamodel/ontology is instantiable
- Verified: transform to ontology & assessed consistency using logical reasoner.

Problem
Specification

Definitions

Abstract

URDAD

Quality drivers
embedded in
URDAD

Internal
consistency

Summary

Semantic quality

URDAD as
Quality-Driven
Process

Solms, Gruner,
Edwards

Model quality impacted by quality of modeling language.

- Define semantics via metamodel or ontology.

Qualities of modeling language:

■ *Completeness*

- Formal lang: power to express statements needed for URDAD.
 - All meaning to be conveyed can be conveyed.
- Informally verified through
 - Analyze URDAD process & models for required semantics.
 - Empirically tested via example models.

■ *Consistency*

- Metamodel/ontology is instantiable
- Verified: transform to ontology & assessed consistency using logical reasoner.

■ *Complexity*

- Assessed by counting classes, relationships & constraints.
- Much lower than for UML (generic language).
 - UML: 16x more classes, 7x more relationships.

Problem
Specification

Definitions

Abstract

URDAD

Quality drivers
embedded in
URDAD

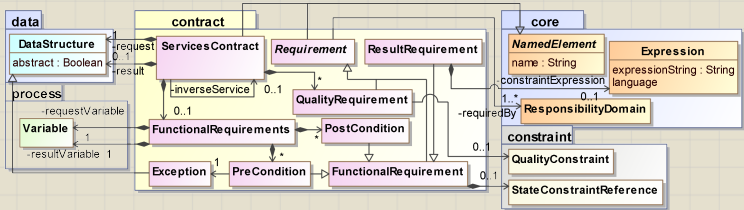
Internal
consistency

Summary

Example: Language elements for contract specification

URDAD as
Quality-Driven
Process

Solms, Gruner,
Edwards



Syntactic quality

Ensure statements made in model comply to syntax rules of metamodel.

- Important for model validation, code, test & documentation generation

URDAD as
Quality-Driven
Process

Solms, Gruner,
Edwards

Problem
Specification

Definitions

Abstract

URDAD

Quality drivers
embedded in
URDAD

Internal
consistency

Summary

Syntactic quality

Ensure statements made in model comply to syntax rules of metamodel.

- Important for model validation, code, test & documentation generation

Syntactic quality drivers

- Define concrete syntax for encoding of models.
 - Text-based or diagrammatic.
 - Bi-directional mapping between syntax & metamodel.
 - Enforces URDAD semantics & model structure.

URDAD as
Quality-Driven
Process

Solms, Gruner,
Edwards

Problem
Specification

Definitions

Abstract

URDAD

Quality drivers
embedded in
URDAD

Internal
consistency

Summary

Syntactic quality

URDAD as
Quality-Driven
Process

Solms, Gruner,
Edwards

Problem
Specification

Definitions

Abstract

URDAD

Quality drivers
embedded in
URDAD

Internal
consistency

Summary

Ensure statements made in model comply to syntax rules of metamodel.

- Important for model validation, code, test & documentation generation

Syntactic quality drivers

- Define concrete syntax for encoding of models.
 - Text-based or diagrammatic.
 - Bi-directional mapping between syntax & metamodel.
 - Enforces URDAD semantics & model structure.
- Generate validating editor for concrete syntax.
 - Done using MDA tool suite.

Syntactic quality

Ensure statements made in model comply to syntax rules of metamodel.

- Important for model validation, code, test & documentation generation

Syntactic quality drivers

- Define concrete syntax for encoding of models.
 - Text-based or diagrammatic.
 - Bi-directional mapping between syntax & metamodel.
 - Enforces URDAD semantics & model structure.
- Generate validating editor for concrete syntax.
 - Done using MDA tool suite.
- Use model validators
 - Compliance to metamodel structure.
 - Adherence to metamodel constraints.

Example: Service contract specification (1/2)

URDAD as
Quality-Driven
Process

Solms, Gruner,
Edwards

Problem
Specification

Definitions

Abstract

URDAD

Quality drivers
embedded in
URDAD

Internal
consistency

Summary

```
1 ServiceContract enrollForPresentation
2 {
3   FunctionalRequirements receiving Variable
     enrollForPresentationRequest ofType
     EnrollForPresentationRequest
4 {
5   PreCondition enrollmentPrerequisitesMet requiredBy (
     TrainingRegulator Student) raises
     EnrollmentPrerequisitesNotSatisfiedException checks constraint
     enrollmentPrerequisitesForPresentationMet with ValueOf
     enrollForPresentationRequest
6   PostCondition enrollmentProcessPerformed requiredBy (Student
     Client TrainingRegulator) ensures constraint
     studentEnrolledForPresentation with ValueOf
     studentEnrolledRequest constructedUsing doSequential
7 {
8   create Variable studentEnrolledRequest ofType
     StudentEnrolledRequest
9   set Query OCL:"studentEnrolledRequest.personIdentifier" equalTo
     Query OCL:"enrollForPresentationRequest.personIdentifier"
10  set Query OCL:"studentEnrolledRequest.presentationIdentifier"
     equalTo Query OCL:"enrollForPresentationRequest.
     presentationIdentifier"
```


Example: Service contract specification (2/2)

URDAD as
Quality-Driven
Process

Solms, Gruner,
Edwards

Problem
Specification

Definitions

Abstract

URDAD

Quality drivers
embedded in
URDAD

Internal
consistency

Summary

```
1   ...
2   PostCondition invoiceIssued ...
3   }
4   Request DataStructure EnrollForPresentationRequest
5   {
6     has identification presentationIdentifier identifying Presentation
7     has identification studentIdentifier identifying Person
8     has identification clientIdentifier identifying LegalEntity
9   }
10  Result DataStructure EnrollForPresentationResult { ... }
11 }
```

Simplicity

URDAD as
Quality-Driven
Process

Solms, Gruner,
Edwards

Inverse measure of complexity.

- Important because reduces cost, risk & improves maintainability.

Problem
Specification

Definitions

Abstract

URDAD

Quality drivers
embedded in
URDAD

Internal
consistency

Summary

Simplicity

URDAD as
Quality-Driven
Process

Solms, Gruner,
Edwards

Inverse measure of complexity.

- Important because reduces cost, risk & improves maintainability.

Simplicity drivers

- Use DSL to provide compact, precise language.
 - Reduce model size & improves understandability.

Problem
Specification

Definitions

Abstract

URDAD

Quality drivers
embedded in
URDAD

Internal
consistency

Summary

Simplicity

URDAD as
Quality-Driven
Process

Solms, Gruner,
Edwards

Inverse measure of complexity.

- Important because reduces cost, risk & improves maintainability.

Simplicity drivers

- Use DSL to provide compact, precise language.
 - Reduce model size & improves understandability.
- Ensure all process activities address functional requirements.
 - Enforced through metamodel.

Problem
Specification

Definitions

Abstract

URDAD

Quality drivers
embedded in
URDAD

Internal
consistency

Summary

Simplicity

URDAD as
Quality-Driven
Process

Solms, Gruner,
Edwards

Inverse measure of complexity.

- Important because reduces cost, risk & improves maintainability.

Simplicity drivers

- Use DSL to provide compact, precise language.
 - Reduce model size & improves understandability.
- Ensure all process activities address functional requirements.
 - Enforced through metamodel.
- Enforce single responsibility principle
 - Assignment of services to responsibility domains.

Problem
Specification

Definitions

Abstract

URDAD

Quality drivers
embedded in
URDAD

Internal
consistency

Summary

Simplicity

URDAD as
Quality-Driven
Process

Solms, Gruner,
Edwards

Inverse measure of complexity.

- Important because reduces cost, risk & improves maintainability.

Simplicity drivers

- Use DSL to provide compact, precise language.
 - Reduce model size & improves understandability.
- Ensure all process activities address functional requirements.
 - Enforced through metamodel.
- Enforce single responsibility principle
 - Assignment of services to responsibility domains.
- No duplication of statements
 - Only one way to specify things.

Problem
Specification

Definitions

Abstract

URDAD

Quality drivers
embedded in
URDAD

Internal
consistency

Summary

Model completeness

The extend to which the model has all elements required for the model use cases

- e.g. code, test & documentation generation.

URDAD as
Quality-Driven
Process

Solms, Gruner,
Edwards

Problem
Specification

Definitions

Abstract

URDAD

Quality drivers
embedded in
URDAD

Internal
consistency

Summary

Model completeness

The extend to which the model has all elements required for the model use cases

- e.g. code, test & documentation generation.

Model completeness drivers

- **Structural completeness criteria**
 - Certain minimal structure enforced through metamodel.

URDAD as
Quality-Driven
Process

Solms, Gruner,
Edwards

Problem
Specification

Definitions

Abstract

URDAD

Quality drivers
embedded in
URDAD

Internal
consistency

Summary

Model completeness

The extend to which the model has all elements required for the model use cases

- e.g. code, test & documentation generation.

Model completeness drivers

- Structural completeness criteria
 - Certain minimal structure enforced through metamodel.
- Process completeness
 - All functional requirements addressed.
 - Enforced through metaodel constraint.

URDAD as
Quality-Driven
Process

Solms, Gruner,
Edwards

Problem
Specification

Definitions

Abstract

URDAD

Quality drivers
embedded in
URDAD

Internal
consistency

Summary

Model completeness

The extend to which the model has all elements required for the model use cases

- e.g. code, test & documentation generation.

Model completeness drivers

- Structural completeness criteria
 - Certain minimal structure enforced through metamodel.
- Process completeness
 - All functional requirements addressed.
 - Enforced through metaodel constraint.
- No enforced completeness on levels of granularity.
 - Decoupled via services contracts.
 - Service provider need not be designed - could be plugged in.

Model completeness

The extend to which the model has all elements required for the model use cases

- e.g. code, test & documentation generation.

Model completeness drivers

- Structural completeness criteria
 - Certain minimal structure enforced through metamodel.
- Process completeness
 - All functional requirements addressed.
 - Enforced through metaodel constraint.
- No enforced completeness on levels of granularity.
 - Decoupled via services contracts.
 - Service provider need not be designed - could be plugged in.
- Process assistance for completeness via process steps with
 - defined inputs & outputs, and
 - defined process tasks.

Model Consistency

URDAD as
Quality-Driven
Process

Solms, Gruner,
Edwards

Consistency often problematic in UML models

- Different UML models structurally and even semantically very different.
- Consistency issues across diagrams (e.g. sequence, activity diagrams & state charts).

Problem
Specification

Definitions

Abstract

URDAD

Quality drivers
embedded in
URDAD

Internal
consistency

Summary

Model Consistency

URDAD as
Quality-Driven
Process

Solms, Gruner,
Edwards

Consistency often problematic in UML models

- Different UML models structurally and even semantically very different.
- Consistency issues across diagrams (e.g. sequence, activity diagrams & state charts).

Model consistency drivers

- Repeatable process with defined inputs, outputs & tasks for each process step.

Problem
Specification

Definitions

Abstract

URDAD

Quality drivers
embedded in
URDAD

Internal
consistency

Summary

Model Consistency

URDAD as
Quality-Driven
Process

Solms, Gruner,
Edwards

Consistency often problematic in UML models

- Different UML models structurally and even semantically very different.
- Consistency issues across diagrams (e.g. sequence, activity diagrams & state charts).

Model consistency drivers

- Repeatable process with defined inputs, outputs & tasks for each process step.
- Enforced model structure & semantics through metamodel.
 - Does not allow duplicate specifications

Problem
Specification

Definitions

Abstract

URDAD

Quality drivers
embedded in
URDAD

Internal
consistency

Summary

Model Cohesion

URDAD as
Quality-Driven
Process

Solms, Gruner,
Edwards

Problem
Specification

Definitions

Abstract

URDAD

Quality drivers
embedded in
URDAD

Internal
consistency

Summary

Cohesion refers the extend to which structural realtionships map onto conceptual and functional relationships.

- Important for localized maintenance, easy finding of model elements, testability reusability and understandability.
- High model cohesion results in code with high cohesion.

Model Cohesion

URDAD as
Quality-Driven
Process

Solms, Gruner,
Edwards

Problem
Specification

Definitions

Abstract

URDAD

Quality drivers
embedded in
URDAD

Internal
consistency

Summary

Cohesion refers the extend to which structural reallationships map onto conceptual and functional relationships.

- Important for localized maintenance, easy finding of model elements, testability reusability and understandability.
- High model cohesion results in code with high cohesion.

Model cohesion drivers

- Responsibility localization
 - Contracts contain only services from same responsibility domain.
 - “Encouraged” by process.

Model Cohesion

URDAD as
Quality-Driven
Process

Solms, Gruner,
Edwards

Problem
Specification

Definitions

Abstract

URDAD

Quality drivers
embedded in
URDAD

Internal
consistency

Summary

Cohesion refers the extend to which structural relationships map onto conceptual and functional relationships.

- Important for localized maintenance, easy finding of model elements, testability reusability and understandability.
- High model cohesion results in code with high cohesion.

Model cohesion drivers

- Responsibility localization
 - Contracts contain only services from same responsibility domain.
 - “Encouraged” by process.
- Services as cohesive, self-contained units
 - Statelessness enforced by metamodel.
 - Each service must address complete functional requirement at some level of granularity.

Traceability

URDAD as
Quality-Driven
Process

Solms, Gruner,
Edwards

Traceability refers to the ability to trace rationale, satisfaction, dependency and evolution.

- Important for estimation, design validation and maintenance.
- Validation for both, sufficiency and necessity.

Problem
Specification

Definitions

Abstract

URDAD

Quality drivers
embedded in
URDAD

Internal
consistency

Summary

Traceability

Traceability refers to the ability to trace rationale, satisfaction, dependency and evolution.

- Important for estimation, design validation and maintenance.
- Validation for both, sufficiency and necessity.

Traceability drivers

- Include satisfaction, dependency and rationale links in model.

URDAD as
Quality-Driven
Process

Solms, Gruner,
Edwards

Problem
Specification

Definitions

Abstract

URDAD

Quality drivers
embedded in
URDAD

Internal
consistency

Summary

Traceability

Traceability refers to the ability to trace rationale, satisfaction, dependency and evolution.

- Important for estimation, design validation and maintenance.
- Validation for both, sufficiency and necessity.

Traceability drivers

- Include satisfaction, dependency and rationale links in model.
- **Rationale links**
 - only indirectly through link of process activity to functional requirement & functional requirement to stakeholder.

URDAD as
Quality-Driven
Process

Solms, Gruner,
Edwards

Problem
Specification

Definitions

Abstract

URDAD

Quality drivers
embedded in
URDAD

Internal
consistency

Summary

Traceability

Traceability refers to the ability to trace rationale, satisfaction, dependency and evolution.

- Important for estimation, design validation and maintenance.
- Validation for both, sufficiency and necessity.

Traceability drivers

- Include satisfaction, dependency and rationale links in model.
- Rationale links
 - only indirectly through link of process activity to functional requirement & functional requirement to stakeholder.
- Satisfaction links
 - between services and service contracts.

URDAD as
Quality-Driven
Process

Solms, Gruner,
Edwards

Problem
Specification

Definitions

Abstract

URDAD

Quality drivers
embedded in
URDAD

Internal
consistency

Summary

Traceability

URDAD as
Quality-Driven
Process

Solms, Gruner,
Edwards

Traceability refers to the ability to trace rationale, satisfaction, dependency and evolution.

- Important for estimation, design validation and maintenance.
- Validation for both, sufficiency and necessity.

Traceability drivers

- Include satisfaction, dependency and rationale links in model.
- Rationale links
 - only indirectly through link of process activity to functional requirement & functional requirement to stakeholder.
- Satisfaction links
 - between services and service contracts.
- Dependency links throughout.
 - e.g. dependencies between services across levels of granularity.

Problem
Specification

Definitions

Abstract

URDAD

Quality drivers
embedded in
URDAD

Internal
consistency

Summary

Traceability

Traceability refers to the ability to trace rationale, satisfaction, dependency and evolution.

- Important for estimation, design validation and maintenance.
- Validation for both, sufficiency and necessity.

Traceability drivers

- Include satisfaction, dependency and rationale links in model.
- Rationale links
 - only indirectly through link of process activity to functional requirement & functional requirement to stakeholder.
- Satisfaction links
 - between services and service contracts.
- Dependency links throughout.
 - e.g. dependencies between services across levels of granularity.
- Evolutionary links through version control.

URDAD as
Quality-Driven
Process

Solms, Gruner,
Edwards

Problem
Specification

Definitions

Abstract

URDAD

Quality drivers
embedded in
URDAD

Internal
consistency

Summary

Example: Service specification (1/2)

URDAD as
Quality-Driven
Process

Solms, Gruner,
Edwards

Problem
Specification

Definitions

Abstract

URDAD

Quality drivers
embedded in
URDAD

Internal
consistency

Summary

```
1 Service enrollForPresentationImpl realizes enrollForPresentation
   receiving Variable enrollForPresentationRequest ofType
   EnrollForPresentationRequest
2 {
3   use checkStudentSatisfiesEnrollmentPrerequisites toAddress (
       enrollmentPrerequisitesMet)
4   use issueInvoice toAddress (financialPrerequisitesSatisfied
       invoiceIssued)
5   ...
6   Process doSequential
7   {
8     create Variable
       checkStudentSatisfiesEnrollmentPrerequisitesRequest ofType
       CheckStudentSatisfiesEnrollmentPrerequisitesRequest
9     set Query OCL:"enrollForPresentationRequest.studentIdentifier"
       equalTo Query OCL:"checkEnrollmentPrerequisitesRequest.
       studentIdentifier"
10    set Query OCL:"enrollForPresentationRequest.presentationIdentifier"
       equalTo Query OCL:"checkEnrollmentPrerequisitesRequest.
       presentationIdentifier"
11    ...
```


Example: Service specification (2/2)

URDAD as
Quality-Driven
Process

Solms, Gruner,
Edwards

Problem
Specification

Definitions

Abstract

URDAD

Quality drivers
embedded in
URDAD

Internal
consistency

Summary

```
1  requestService checkStudentSatisfiesEnrollmentPrerequisites with  
    checkStudentSatisfiesEnrollmentPrerequisitesRequest yielding  
    Variable checkStudentSatisfiesEnrollmentPrerequisitesResult  
    ofType CheckStudentSatisfiesEnrollmentPrerequisitesResult  
2  choice  
3  {  
4      if Constraint enrollmentMeetsPrerequisitesMet OCL:"  
        checkStudentSatisfiesEnrollmentPrerequisitesResult.  
        enrollmentPrerequisitesMet = true" doSequential  
5      {  
6          ...  
7          requestService issueInvoice with issueInvoiceRequest yielding  
            Variable issueInvoiceResult ofType IssueInvoiceResult  
8          {  
9              on FinancialPrerequisitesNotSatisfiedException raiseException  
                FinancialPrerequisitesNotSatisfiedException  
10         }  
11         ...  
12         returnResult enrollForPresentationResult  
13     }  
14     else raiseException EnrollmentPrerequisitesNotSatisfiedException  
15 }  
16 }
```

Model modifiability

Modifiability refers to the ease with which the model can be modified.

- Important for maintenance in context of change requests and refactorization.

URDAD as
Quality-Driven
Process

Solms, Gruner,
Edwards

Problem
Specification

Definitions

Abstract

URDAD

Quality drivers
embedded in
URDAD

Internal
consistency

Summary

Model modifiability

Modifiability refers to the ease with which the model can be modified.

- Important for maintenance in context of change requests and refactorization.

Modifiability drivers

- Decoupling via services contracts
 - Modifiability through decoupling.
 - “Enforced” by process & metamodel.

URDAD as
Quality-Driven
Process

Solms, Gruner,
Edwards

Problem
Specification

Definitions

Abstract

URDAD

Quality drivers
embedded in
URDAD

Internal
consistency

Summary

Model modifiability

Modifiability refers to the ease with which the model can be modified.

- Important for maintenance in context of change requests and refactorization.

Modifiability drivers

- Decoupling via services contracts
 - Modifiability through decoupling.
 - “Enforced” by process & metamodel.
- Defined levels of granularity
 - Process includes step to check whether additional levels of granularity should be defined.
 - Requirements engineer verifies whether any services at any level of granularity can be combined into single, cohesive, higher-level service.

URDAD as
Quality-Driven
Process

Solms, Gruner,
Edwards

Problem
Specification

Definitions

Abstract

URDAD

Quality drivers
embedded in
URDAD

Internal
consistency

Summary

Model modifiability

Modifiability refers to the ease with which the model can be modified.

- Important for maintenance in context of change requests and refactorization.

Modifiability drivers

- Decoupling via services contracts
 - Modifiability through decoupling.
 - “Enforced” by process & metamodel.
 - Defined levels of granularity
 - Process includes step to check whether additional levels of granularity should be defined.
 - Requirements engineer verifies whether any services at any level of granularity can be combined into single, cohesive, higher-level service.
- Simplicity and hence its quality drivers also improve modifiability.

URDAD as
Quality-Driven
Process

Solms, Gruner,
Edwards

Problem
Specification

Definitions

Abstract

URDAD

Quality drivers
embedded in
URDAD

Internal
consistency

Summary

Reusability

Reusability refers to the ease with which model elements can be reused.

- Important for reducing development & maintenance cost & risk, as well as consistency.

URDAD as
Quality-Driven
Process

Solms, Gruner,
Edwards

Problem
Specification

Definitions

Abstract

URDAD

Quality drivers
embedded in
URDAD

Internal
consistency

Summary

Reusability

Reusability refers to the ease with which model elements can be reused.

- Important for reducing development & maintenance cost & risk, as well as consistency.

Reusability drivers

- All services realize services contracts
 - Modifiability through decoupling.
 - “Enforced” by process & metamodel.

URDAD as
Quality-Driven
Process

Solms, Gruner,
Edwards

Problem
Specification

Definitions

Abstract

URDAD

Quality drivers
embedded in
URDAD

Internal
consistency

Summary

Reusability

Reusability refers to the ease with which model elements can be reused.

- Important for reducing development & maintenance cost & risk, as well as consistency.

Reusability drivers

- All services realize services contracts
 - Modifiability through decoupling.
 - “Enforced” by process & metamodel.
- Optimize levels of granularity

URDAD as
Quality-Driven
Process

Solms, Gruner,
Edwards

Problem
Specification

Definitions

Abstract

URDAD

Quality drivers
embedded in
URDAD

Internal
consistency

Summary

Reusability

Reusability refers to the ease with which model elements can be reused.

- Important for reducing development & maintenance cost & risk, as well as consistency.

Reusability drivers

- All services realize services contracts
 - Modifiability through decoupling.
 - “Enforced” by process & metamodel.
- Optimize levels of granularity
- **Stateless, self-contained services.**

URDAD as
Quality-Driven
Process

Solms, Gruner,
Edwards

Problem
Specification

Definitions

Abstract

URDAD

Quality drivers
embedded in
URDAD

Internal
consistency

Summary

Reusability

Reusability refers to the ease with which model elements can be reused.

- Important for reducing development & maintenance cost & risk, as well as consistency.

Reusability drivers

- All services realize services contracts
 - Modifiability through decoupling.
 - “Enforced” by process & metamodel.
- Optimize levels of granularity
- Stateless, self-contained services.
- Cohesion and hence its quality drivers.

URDAD as
Quality-Driven
Process

Solms, Gruner,
Edwards

Problem
Specification

Definitions

Abstract

URDAD

Quality drivers
embedded in
URDAD

Internal
consistency

Summary

Reusability

Reusability refers to the ease with which model elements can be reused.

- Important for reducing development & maintenance cost & risk, as well as consistency.

Reusability drivers

- All services realize services contracts
 - Modifiability through decoupling.
 - “Enforced” by process & metamodel.
- Optimize levels of granularity
- Stateless, self-contained services.
- Cohesion and hence its quality drivers.
- Linkage btw service & contract aids service provider discoverability.

URDAD as
Quality-Driven
Process

Solms, Gruner,
Edwards

Problem
Specification

Definitions

Abstract

URDAD

Quality drivers
embedded in
URDAD

Internal
consistency

Summary

Reusability

Reusability refers to the ease with which model elements can be reused.

- Important for reducing development & maintenance cost & risk, as well as consistency.

Reusability drivers

- All services realize services contracts
 - Modifiability through decoupling.
 - “Enforced” by process & metamodel.
- Optimize levels of granularity
- Stateless, self-contained services.
- Cohesion and hence its quality drivers.
- Linkage btw service & contract aids service provider discoverability.
- Enforced adapter layer.

URDAD as
Quality-Driven
Process

Solms, Gruner,
Edwards

Problem
Specification

Definitions

Abstract

URDAD

Quality drivers
embedded in
URDAD

Internal
consistency

Summary

Testability

URDAD as
Quality-Driven
Process

Solms, Gruner,
Edwards

Testability refers to the ease with which the model or its implementation mapping can be tested.

- Important for model & code validation.

Problem
Specification

Definitions

Abstract

URDAD

Quality drivers
embedded in
URDAD

Internal
consistency

Summary

Testability

URDAD as
Quality-Driven
Process

Solms, Gruner,
Edwards

Testability refers to the ease with which the model or its implementation mapping can be tested.

- Important for model & code validation.

Testability drivers

- Fully specified services contracts
 - In service-oriented paradigm, services can only be tested by
 - Extracting information about environment using other services.
 - Assessing constraints on obtained information.

Problem
Specification

Definitions

Abstract

URDAD

Quality drivers
embedded in
URDAD

Internal
consistency

Summary

Testability

URDAD as
Quality-Driven
Process

Solms, Gruner,
Edwards

Problem
Specification

Definitions

Abstract

URDAD

Quality drivers
embedded in
URDAD

Internal
consistency

Summary

Testability refers to the ease with which the model or its implementation mapping can be tested.

- Important for model & code validation.

Testability drivers

- Fully specified services contracts
 - In service-oriented paradigm, services can only be tested by
 - Extracting information about environment using other services.
 - Assessing constraints on obtained information.
- Metamodel
 - Contract has constraint as either pre- or post-condition.
 - Same state constraint can be pre- and post- condition for different services.

Example: State constraint specification

URDAD as
Quality-Driven
Process

Solms, Gruner,
Edwards

Problem
Specification

Definitions

Abstract

URDAD

Quality drivers
embedded in
URDAD

Internal
consistency

Summary

```
1 StateConstraint studentEnrolledForPresentation receiving Variable  
   enrollForPresentationRequest ofType  
   EnrollForPresentationRequest  
2 {  
3   stateAssessmentProcess doSequential  
4   {  
5     create Variable getEnrollmentsRequest ofType GetEnrollmentsRequest  
6     set Query OCL:"getEnrollmentsRequest.presentationIdentifier"  
       equalTo Query OCL:"enrollForPresentationRequest.  
       presentationIdentifier"  
7     requestService getEnrollments with getEnrollmentsRequest yielding  
       Variable getEnrollmentsResult ofType GetEnrollmentsResult  
8   }  
9   Constraint OCL:"getEnrollmentsResult.enrollments.includes (  
       enrollForPresentationRequest.personIdentifier)"  
10 }
```


Summary of quality drivers in URDAD

URDAD as
Quality-Driven
Process

Solms, Gruner,
Edwards

Quality-driver	Model qualities									
	Semantic	Syntactic	Pragmatic model qualities							
			Simplicity	Completeness	Modifiability	Consistency	Decoupling	Cohesion	Reusability	Traceability
Problem Specification	✓	✓	✓	✓	✓	✓	✓			✓
Definitions		✓	✓		✓					
Abstract			✓		✓					
URDAD			✓		✓				✓	✓
Quality drivers embedded in URDAD			✓		✓		✓		✓	✓
			✓		✓			✓	✓	✓
Internal consistency			✓	✓	✓	✓		✓		✓
Summary			✓	✓	✓	✓	✓	✓	✓	✓

Internal consistency of methodology

URDAD as
Quality-Driven
Process

Solms, Gruner,
Edwards

Problem
Specification

Definitions

Abstract

URDAD

Quality drivers
embedded in
URDAD

Internal
consistency

Summary

- URDAD = analysis & design methodology used to design services
 - Apply process to design service of performing analysis & design for service.
 - Should result in service with process, semantics & model structure which represents URDAD itself.
 - If it doesn't, then not internally consistent.
 - If it does, it does not show that URDAD is a good methodology, but only that it is consistent.

Internal consistency of methodology

URDAD as
Quality-Driven
Process

Solms, Gruner,
Edwards

Problem
Specification

Definitions

Abstract

URDAD

Quality drivers
embedded in
URDAD

Internal
consistency

Summary

- URDAD = analysis & design methodology used to design services
 - Apply process to design service of performing analysis & design for service.
 - Should result in service with process, semantics & model structure which represents URDAD itself.
 - If it doesn't, then not internally consistent.
 - If it does, it does not show that URDAD is a good methodology, but only that it is consistent.
- Applying URDAD to design service-oriented A&D methodology regenerates
 - process, and
 - metamodel.

Summary

URDAD as
Quality-Driven
Process

Solms, Gruner,
Edwards

Problem
Specification

Definitions

Abstract

URDAD

Quality drivers
embedded in
URDAD

Internal
consistency

Summary

■ Linked quality drivers to quality criteria.

Summary

URDAD as
Quality-Driven
Process

Solms, Gruner,
Edwards

Problem
Specification

Definitions

Abstract

URDAD

Quality drivers
embedded in
URDAD

Internal
consistency

Summary

- Linked quality drivers to quality criteria.
- Demonstrated how quality drivers used in URDAD process.

Summary

URDAD as
Quality-Driven
Process

Solms, Gruner,
Edwards

Problem
Specification

Definitions

Abstract

URDAD

Quality drivers
embedded in
URDAD

Internal
consistency

Summary

- Linked quality drivers to quality criteria.
- Demonstrated how quality drivers used in URDAD process.
- When using URDAD to design A&D process, one can regenerate URDAD with its metamodel.