ER
Enginyeria de Requisits

REQUIREMENTS VALIDATION

FACULTAT D'INFORMÀTICA DE BARCELONA FIB UNIVERSITAT POLITÈCNICA DE CATALUNYA BARCELONATECH

Maria Ribera Sancho, Ernest Teniente, Sergio Morales

Professors

Maria-Ribera Sancho
Enginyeria de Serveis i Sistemes d'Informació
Edifici: Omega, despatx 130
Email: maria.ribera.sancho@upc.edu

Sergio Morales
Enginyeria de Serveis i Sistemes d'Informació
Edifici: Omega, despatx 120
Email: sergio.morales@upc.edu

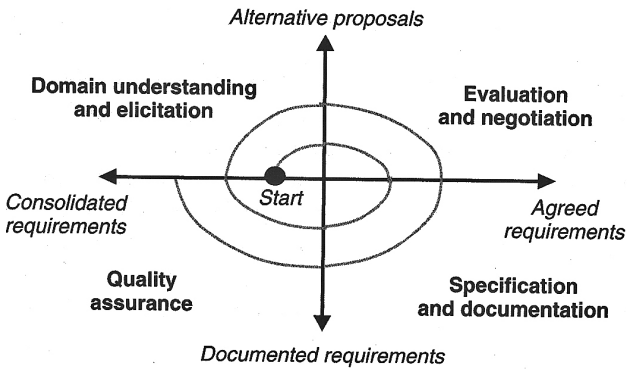
Ernest Teniente
Enginyeria de Serveis i Sistemes d'Informació
Edifici : Omega, despatx 132
Email: teniente@essi.upc.edu

Sergio Paredes
Enginyeria de Serveis i Sistemes d'Informació
Edifici : Omega, despatx 120
Email: sergio.paredes@upc.edu

Outline

- Fundamentals
- Quality of Conceptual Schemas
  - Motivation
  - Conceptual Schema Verification
  - Conceptual Schema Validation
- Exercise

Requirements Specification



*Axel van Lamsweerde, **Requirements Engineering**, 2009*

Validation: Definition

Requirements validation is the activity from requirements engineering aimed at **checking** that:

- the inputs of requirements engineering
- the activities performed during requirements engineering
- the artifacts created during requirements engineering

fulfil defined quality criteria.

Requirements Engineering. Pohl. Springer. 2010, chapter 27-29. http://cataleg.upc.edu/record=b1379140~S1*cat

Validation in the Content Dimension

1. **Completeness** (artifact/document)
   - *Does the artifact or document contains all relevant information and requirements?*
2. **Consistency**
   - *Can the system meet all requirements together or there are contradictions among them?*
3. **Correctness** (requirement) **and necessity**
   - *Does each requirement correspond to stakeholders needs and wishes?*
   - *Is each documented requirement needed to satisfy an agreed goal?*
4. **No premature design decisions**
   - *Is there any requirement biased towards a specific implementation?*
5. **Testability**
   - *Have acceptance criteria been defined for each requirement?*
6. **Traceability**
   - *Is it possible to know from where each requirement comes from?*

Requirements Engineering. Pohl. Springer. 2010, chapter 27.3.1. http://cataleg.upc.edu/record=b1379140~S1*cat

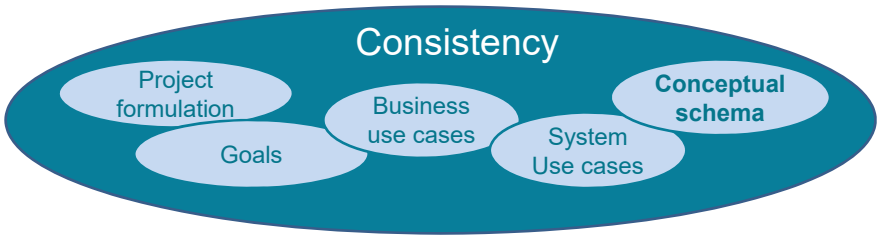
Validation principles

1. **Involving the right stakeholders**
   - *They depend on the artifact and the goal of validation*
   - *Independence of reviewers (i.e. they cannot be the authors)*
   - *Internal vs external validation (from the organization)*
2. **Separate defect detection from defect correction**
   - *Correction does not start until having a complete vision of validation and when all defects have been confirmed*
3. **Leveraging multiple independent views**
   - *Reviewers act independently ones from the others*
   - *Reviewers validate different parts of the artifact*
   - *Reviewers validate from different points of view (designer, user, etc.)*

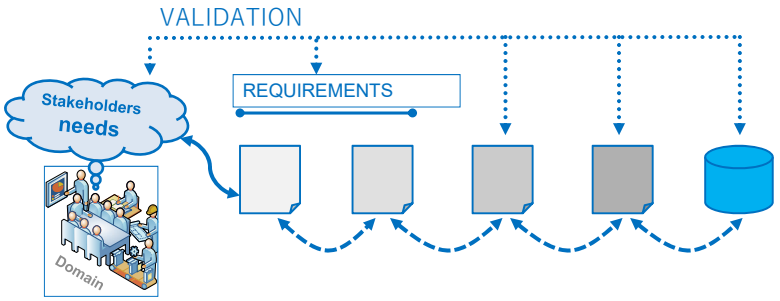
Requirements Engineering. Pohl. Springer. 2010, chapter 27.8. http://cataleg.upc.edu/record=b1379140~S1*cat

Artifacts Consistency

Consistency between requirements specifications
=
lack of contradictions between artifacts



Why early validation?



Requirement Artifacts

Requirements

Functional Requirements

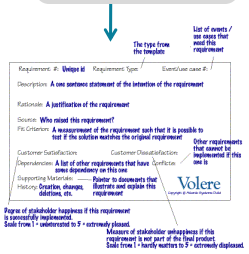
Quality Requirements

User-system interaction

Use case: Create customer
Scope: EU-Rent system
Primary Actor: Customer
Stakeholders and interests:
Precondition: The customer does not exist.
Trigger: A new customer wants to make a rental car reservation.
Main Success Scenario:
1. The customer provides his personal information (name, address, date of birth).
2. The customer provides the details of his driving license.
3. The system records the information about the new customer.

+

Conceptual schema specification

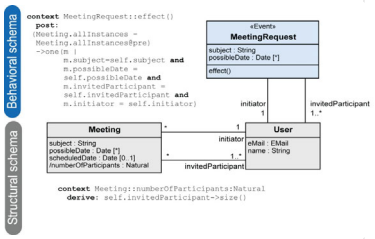


Conceptual Schemas

How do we build conceptual schemas ?

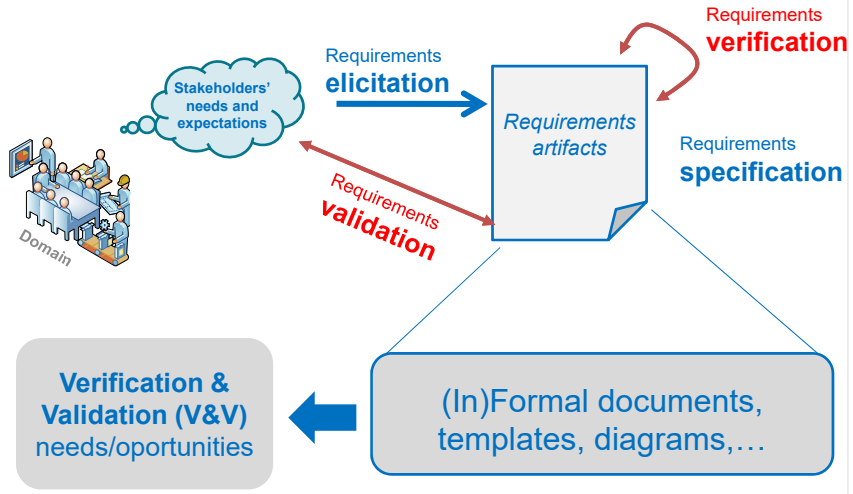
A user can request a meeting specifying the invited participants, its subject and its possible dates. The number of participants is….

...



Formal conceptual modeling languages (UML / OCL / BPMN)

Assessing the quality of requirements' artifacts



Conceptual schemas should be

Valid

Correct: The knowledge it defines is true for the domain.

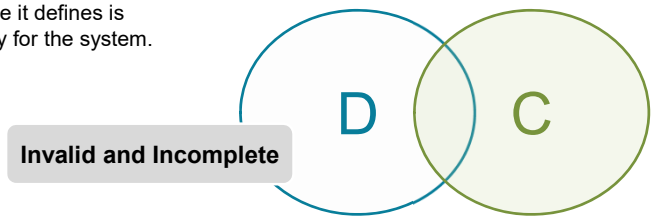
Relevant: The knowledge it defines is necessary for the system.

Complete

The conceptual schema includes all relevant knowledge

D = Required Knowledge

C = Knowledge defined in the CS



Invalid and Incomplete

Conceptual schemas should be

Valid

Correct: The knowledge it defines is true for the domain.

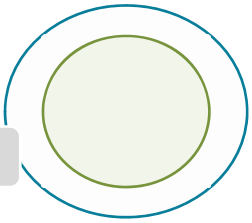
Relevant: The knowledge it defines is necessary for the system.

Complete

The conceptual schema includes all relevant knowledge

D = Required Knowledge

C = Knowledge defined in the CS



Valid but incomplete CS

Conceptual schemas should be

Valid

Correct: The knowledge it defines is true for the domain.

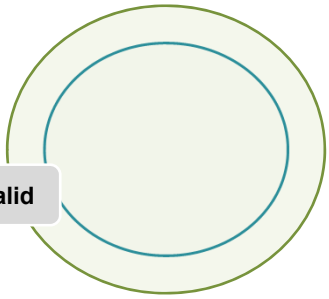
Relevant: The knowledge it defines is necessary for the system.

Complete

The conceptual schema includes all relevant knowledge

D = Required Knowledge

C = Knowledge defined in the CS



Complete but invalid

Conceptual schemas should be

Valid

Correct: The knowledge it defines is true for the domain.

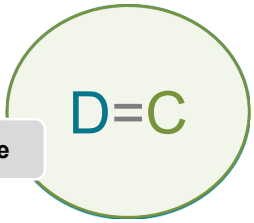
Relevant: The knowledge it defines is necessary for the system.

Complete

The conceptual schema includes all relevant knowledge

D = Required Knowledge

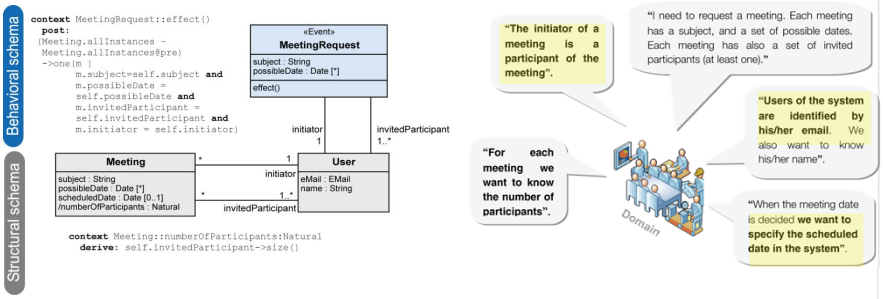
C = Knowledge defined in the CS



Valid and complete

Quality of a Conceptual Schema

Syntactically correct … **but invalid and incomplete**



Conceptual schema **verification & validation**

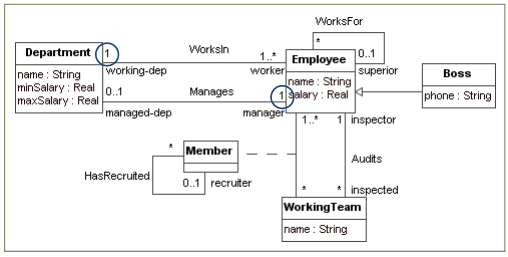
A critical (and non-trivial) problem in requirements engineering

Conceptual Schema Verification

Is the Conceptual Schema **Right**?

It must satisfy a set of general properties, independently of the requirements

Is it possible to have at least one instance of *Employee* (liveliness)?




```
Integrity constraints
…
6.context Department inv ManagerIsWorker:
self.worker->includes(self.manager)
…
8.context Boss inv BossIsManager:
self.managed-dep->notEmpty()
9.context Boss inv BossHasNoSuperior:
self.superior->isEmpty()
10.context Boss inv SuperiorOfAllWorkers:
self.employee->includesAll(self.managed-
dep.worker)
```

Employee	employee(#e1,john)
WorksIn	worksIn(#e1,#s1)
Department	department(#s1,sales)
Manages	manages(#e1,#s1)

Employee **is lively**

The class is well-defined

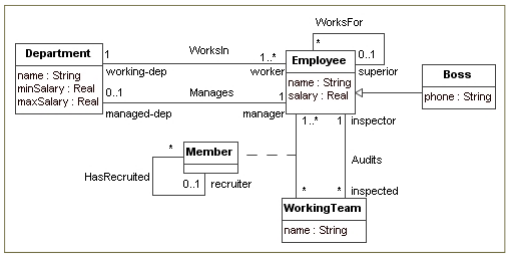
Verification tests:
- **automatically generated**
- their **result** determines the (un)correctness of the schema

Conceptual Schema Validation

Is it the Right Conceptual Schema?

The knowledge it represents corresponds to the requirements

May an Employee work for himself?




```
Integrity constraints

…
6.context Department inv ManagerIsWorker:
self.worker->includes(self.manager)
7.context Department inv MgrHasNoSuperior:
self.manager.superior->isEmpty()
…
```

WorksFor	worksFor(#e1, #e1)
Employee	employee(#e1,mary)
WorksIn	worksIn(#e1, #s1)
Department	department(#s1,sales)
Manages	manages(#e1, #s1)

An employee can work for himself

Is the schema correct?

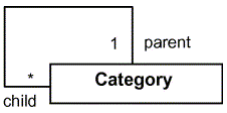
Validation tests:
- can be automatically generated
- the **designer** must determine the (un)correctness of the schema

V&V in UML/OCL Conceptual Schemas

UML/OCL allow formal definitions of conceptual schemas

↓

V&V



UML/OCL is based on logics

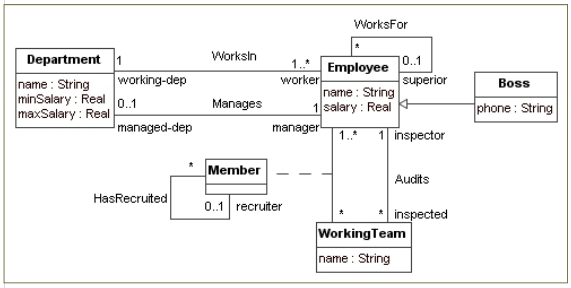
↓

Some properties are automatically verifiable

Exercise: Find a valid instantiation of this conceptual schema fragment
(assume that there is a constraint that does not allow cycles)

An Exercise …

Given the following UML conceptual schema:



Restriccions textuals:

1. Claus externes: (Department,name); (Employee,name); (WorkingTeam,name)
2. El sou mínim d'un departament és de 1000€
3. El sou mínim d'un departament ha de ser inferior al seu sou màxim
4. El director (manager) d'un departament ha de ser un dels seus empleats (worker)
5. El director d'un departament no té cap superior
6. Si un empleat és Boss, aleshores ha de dirigir (managedDep) algun departament
7. Els Boss no tenen cap superior
8. El Boss és el superior de tots els empleats que treballen al departament que ell dirigeix
9. L'inspector d'un WorkingTeam no pot ser membre d'aquell WorkingTeam
10. Un membre no es pot reclutar (recruiter) a ell mateix
11. Algun dels membres d'un WorkingTeam ha d'haver sigut reclutat per un altre membre del mateix Working Team

An Exercise …

Which is the answer to the following questions?

1. Do all classes have an external identifier (key)?
2. Is it possible that the manager of a department does not work in it?
3. Is it possible to have an employee earning less than the minimum salary of his/her department?
4. Is it possible to have an employee who is not a boss?
5. Is there any class or association that does not allow any instance?
6. Is there any redundant integrity constraint (so that it can never be violated)?
7. May an employee be recruited by himself?