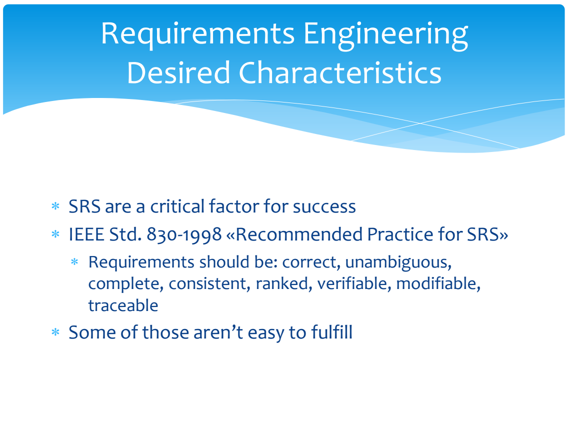
Aspect Oriented Software Development

Early Aspects – Aspect-Oriented Requirement Engineering

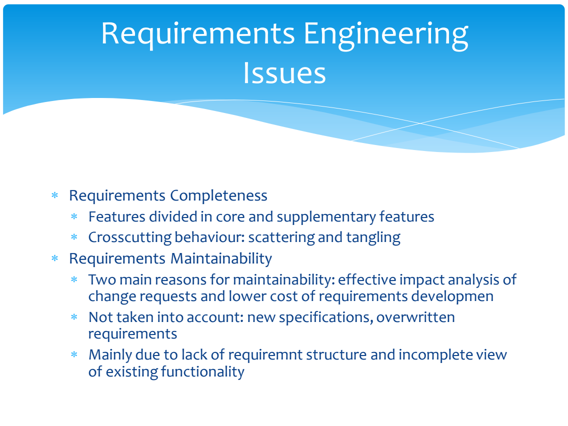
Agenda

- Requirements Engineering
- Methodologies
- Aspect-Oriented Requirement Engineering
  - Requirement Process
- Adopting AORE
  - Analysis
  - Specification



Requirements Engineering Desired Characteristics

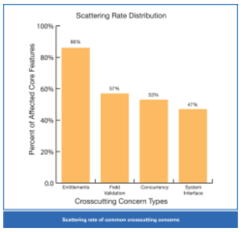
- SRS are a critical factor for success
- IEEE Std. 830-1998 «Recommended Practice for SRS»
  - Requirements should be: correct, unambiguous, complete, consistent, ranked, verifiable, modifiable, traceable
- Some of those aren't easy to fulfill



Requirements Engineering Issues

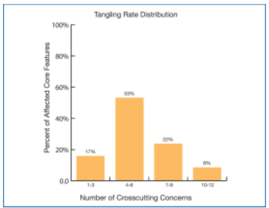
- Requirements Completeness
  - Features divided in core and supplementary features
  - Crosscutting behaviour: scattering and tangling
- Requirements Maintainability
  - Two main reasons for maintainability: effective impact analysis of change requests and lower cost of requirements developmen
  - Not taken into account: new specifications, overwritten requirements
  - Mainly due to lack of requiremnt structure and incomplete view of existing functionality

Requirements Engineering Issues



Scattering rate of common crosscutting concerns

Requirements Engineering Issues

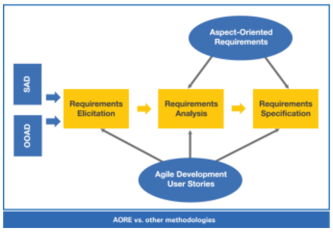


Tangling rate of core features

Methodologies

- SAD: Structured analysis and design
- OOAD: Object-oriented analysis and design
- ASD: Agile software development
- AOSD: Aspect-oriented software development

Methodologies
Requirement Process

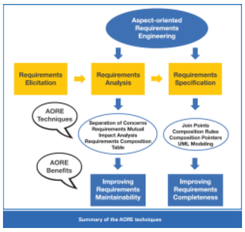

Aspect-Oriented Requirements
SAD
OOAD
Requirements Elicitation
Requirements Analysis
Requirements Specification
Agile Development User Stories

AORE vs. other methodologies

Aspect-Oriented Requirement Engineering

- Addresses problems caused by crosscutting concerns during requirement stages
- Provides means to identify, modularize, compose and specify crosscutting concerns
- Extension of Separation of Concerns, providing a two-dimensional decomposition:
  - core features
  - crosscutting concerns
- Improvement of requirements modularization, enhancing completeness and maintainability of requirements

AORE
Requirement Process

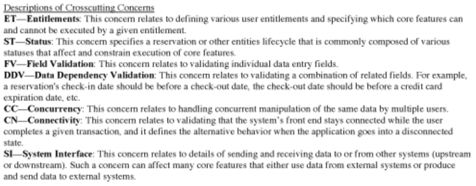

Aspect-oriented Requirements Engineering
Requirements Elicitation
Requirements Analysis
Requirements Specification
AORE Techniques
Separation of Concerns Requirements Mutual Impact Analysis Requirements Composition Table
Join Points Composition Rules Composition Pointers UML Modeling
AORE Benefits
Improving Requirements Maintainability
Improving Requirements Completeness

Summary of the AORE techniques

Adopting AORE - Analysis Identifying CCC

- Identification of EA by means of a manual inspection, based on experience or catalogs of recurrent CCC, or by using a semiautomatic tool
- Specification of general behavior of each early aspect

Adopting AORE - Analysis Identifying CCC



Descriptions of Crosscutting Concerns

ET—Entitlements: This concern relates to defining various user entitlements and specifying which core features can and cannot be executed by a given entitlement.

ST—Status: This concern specifies a reservation or other entities lifecycle that is commonly composed of various statuses that affect and constrain execution of core features.

FV—Field Validation: This concern relates to validating individual data entry fields.

DDV—Data Dependency Validation: This concern relates to validating a combination of related fields. For example, a reservation's check-in date should be before a check-out date, the check-out date should be before a credit card expiration date, etc.

CC—Concurrency: This concern relates to handling concurrent manipulation of the same data by multiple users.

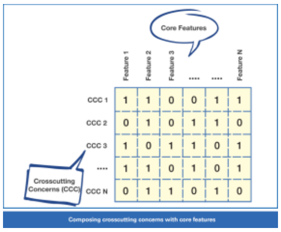
CN—Connectivity: This concern relates to validating that the system's front end stays connected while the user completes a given transaction, and it defines the alternative behavior when the application goes into a disconnected state.

SI—System Interface: This concern relates to details of sending and receiving data to or from other systems (upstream or downstream). Such a concern can affect many core features that either use data from external systems or produce and send data to external systems.

Adopting AORE - Analysis Requirements Composition Table

- It is a structure to link the two-dimensional model (core vs. crosscutting features)
- “means that for each core feature context we indicate which crosscutting concerns impact this core feature and become a part of its context”
- Provides a holistic view of the application functionality

Adopting AORE - Analysis Requirements Composition Table


Core Features
Crosscutting Concerns (CCC)

	Feature 1	Feature 2	Feature 3	....	....	Feature N
CCC 1	1	1	0	0	1	1
CCC 2	0	1	0	1	1	0
CCC 3	1	0	1	1	0	1
....	1	1	0	1	0	1
CCC N	0	1	1	0	1	0

Composing crosscutting concerns with core features

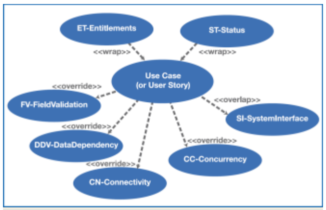
Adopting AORE - Specification Composition Rules

- Establishes how CCC affect other features
- Three types of composition rules:
  - Wrap: a CCC imposes a constraint and controls the entire context of a core feature
  - Override: a CCC can interrupt the core feature flow and change its outcome
  - Overlap: a CCC adds detail to a core feature

Adopting AORE - Specification Composition Rules in UML

- A CCC is shown a use case using the <<aspect>> stereotype
- Each composition is represented by a relationship with an according stereotype:
  - <<wrap>>: dashed arrow pointing from CCC to UC
  - <<override>> and <<overlap>>: dashed arrow pointing from UC to CCC

Adopting AORE - Specification Composition Rules in UML

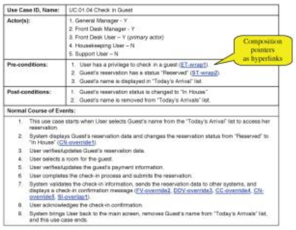


Composition modeling using UML

Adopting AORE - Specification Join Points

- Are steps or portions of the use case specification which are impacted by one or more crosscutting concern
- Indicates the composition of a use case with the affected CCC
- Don't confuse with extension points
- It may imply some specification refactorings

Adopting AORE - Specification Join Points

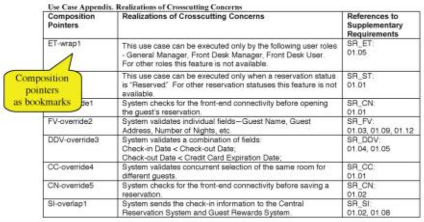


Use Case ID, Name:	UC.01.04 Check in Guest
Actor(s):	1. General Manager - Y 2. Front Desk Manager - Y 3. Front Desk User – Y (primary actor) 4. Housekeeping User – N 5. Support User – N
Pre-conditions:	1. User has a privilege to check in a guest (ET-wrap1) 2. Guest's reservation has a status "Reserved" (ST-wrap2) 3. Guest's name is displayed in "Today's Arrival" list.
Post-conditions:	1. Guest's reservation status is changed to "In House" 2. Guest's name is removed from "Today's Arrivals" list.
Normal Course of Events:	
1. This use case starts when User selects Guest's name from the "Today's Arrival" list to access her reservation. 2. System displays Guest's reservation data and changes the reservation status from "Reserved" to "In House" (CN-override1). 3. User verifies/updates Guest's reservation data. 4. User selects a room for the guest. 5. User verifies/updates the guest's payment information. 6. User completes the check-in process and submits the reservation. 7. System validates the check-in information, sends the reservation data to other systems, and displays a check-in confirmation message (FV-override2, DDV-override3, CC-override4, CN-override5, SI-overlap1). 8. User acknowledges the check-in confirmation. 9. System brings User back to the main screen, removes Guest's name from "Today's Arrivals" list, and this use case ends.	

Adopting AORE - Specification Realization of CCC

- Means capturing high-level ideas about how these concerns affect a given requirement (such as a use case)
- The complete details about the CCC are encapsulated in a specific document per CCC, while realizations represent the influence of a CCC over a core feature

Adopting AORE - Specification Realization of CCC



Use Case Appendix. Realizations of Crosscutting Concerns

Composition Pointers	Realizations of Crosscutting Concerns	References to Supplementary Requirements
ET-wrap1	This use case can be executed only by the following user roles - General Manager, Front Desk Manager, Front Desk User. For other roles this feature is not available.	SR_ET: 01.05
	This use case can be executed only when a reservation status is "Reserved." For other reservation statuses this feature is not available.	SR_ST: 01.01
de1	System checks for the front-end connectivity before opening the guest's reservation.	SR_CN: 01.01
FV-override2	System validates individual fields—Guest Name, Guest Address, Number of Nights, etc.	SR_FV: 01.03, 01.09, 01.12

DDV-override3	System validates a combination of fields:
	Check-in Date < Check-out Date;
	Check-out Date < Credit Card Expiration Date;
		SR_DDV:
		01.04, 01.05

CC-override4	System validates concurrent selection of the same room for different guests.	SR_CC: 01.01
CN-override5	System checks for the front-end connectivity before saving a reservation.	SR_CN: 01.02
SI-overlap1	System sends the check-in information to the Central Reservation System and Guest Rewards System.	SR_SI: 01.02, 01.08

Composition pointers as bookmarks

Questions