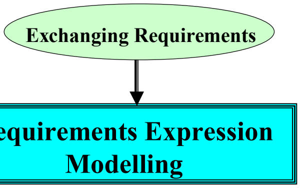
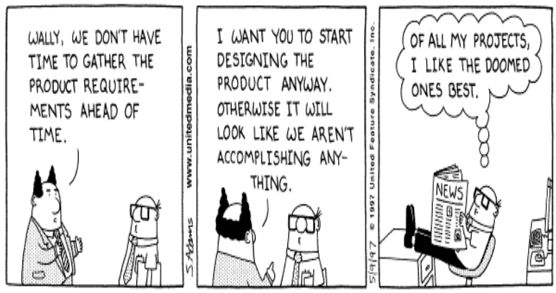
Requirements Expression and Modelling



Exchanging Requirements

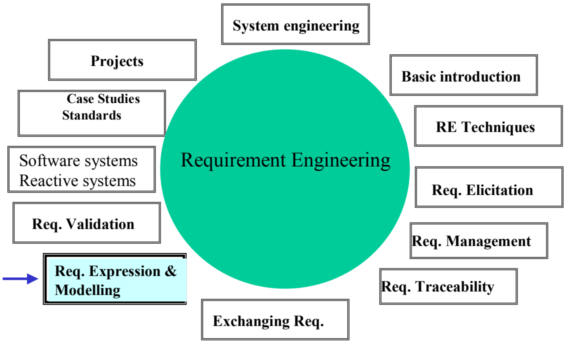
Requirements Expression Modelling

RE According To Dilbert


WALLY, WE DON'T HAVE TIME TO GATHER THE PRODUCT REQUIREMENTS AHEAD OF TIME.
www.unitedmedia.com
S.Adams
I WANT YOU TO START DESIGNING THE PRODUCT ANYWAY. OTHERWISE IT WILL LOOK LIKE WE AREN'T ACCOMPLISHING ANYTHING.
5/9/97 © 1997 United Feature Syndicate, Inc.
OF ALL MY PROJECTS, I LIKE THE DOOMED ONES BEST.
NEWS

Copyright © 1997 United Feature Syndicate, Inc.
Redistribution in whole or in part prohibited

Requirement expression and Modelling



System engineering

Projects

Basic introduction

Case Studies
Standards

RE Techniques

Software systems
Reactive systems

Requirement Engineering

Req. Elicitation

Req. Validation

Req. Management

Req. Expression & Modelling

Req. Traceability

Exchanging Req.

Introduction

Most confused notion in requirement engineering :

It concerns the notation in view for <u>exchange</u>, communication and <u>validation</u> for further development

The natural way

- Use natural language : many systems were developped and still being developped using natural language
- Everyday life and social activity is made through natural language
- **BUT** .....
- No precise semantics
- No structuring
- No abstraction
- No Validation ... Except some exceptions

Content

- Requirements for a requirements notation
- Classes of languages, models, tools, methods, techniques
- Review of basic related aspects (*seen in other courses*)
- Main methods used in industry
- Formal notations and associated methods
- Case study and lab : Statecharts

Requirements for a requirements notation

- Let‘s recall the main processes
  - Elicitation
  - Expression
  - Validation
  - Generation of specification
    - Hardware
    - Software requirements
    - Others systems specification

Requirements for such notation

- Express Behaviour
- Data specification
- Functions (data transformation)
- Supports abstraction
- Executable (ideally)
- Associated method/methodology

What to express

- Requirements that can be understood
- That can have a single meaning
- That can be refined when needed
- That can strcturerd for managing inconsistencies and changes

The modelling Issue

- Modelling can guide elicitation
- Modelling can provide a measure of progess
- Modelling can help to uncover problems (Inconsistencies)
- Central concepts
  - Abstraction
  - Process modelling
  - Data Modelling
  - Data Flow
  - Behaviour
  - Etc ...

Example

- The system must be reliable
  - 1. Not precise
  - 2. Qualitative attribute
  - 3. Add a quantitative attribute
  - Context in which le system must be reliable

  (the context means system internal context and environment)

Writing good requirements : syntax and semantics as any statement

Classes of languages, models, tools, methodes, techniques

- Consider Software systems
- Any language can be a requirement language
- A programming language is a requirement language for the computer to execute what is required
  - Example : While {..} do ....
  - It is well specified and no ambiguity
- **However in RE**
  - There many stakeholders
  - Different cultures (not necessarily computer scientists or familiar with programming languages)
  - Requirements are of many types

Classes

- Programming languages
- Specific notation
- General purpose Methods
  - Informal
  - Semi-formal : Used in industry
  - Formal : Developped by academia
  - Abstract : limited to specific issues (pure academia work)
- Paradigms
  - Function oriented
  - Object

Review of basic related aspects (*seen in other courses*)

- Control structure (behaviour) : seq; //, if .. Else
- Communication (shared data, synch, async, ..)
- Abstraction
- Encapsulation
- Properties
- Invariants

CRITERIA *(CMU-DoD_SEI Taxonomy)*

- Representation

  → Concepts and techniques described using the technique
- Derivation

  → Methods to produce a specification from another
- Validation-Verification

  → Properties that can be determined using the specification technique

REPRESENTATION

- Style
- Concurrency
- Communication
- Non-Determinism
- Fairness
- Modularity
- Time
- Data

DERIVATION

- Transformation

  → Transformation rules from a specification technique to another (e.g multiformalism approach)
- Elaboration

  → Same as above with a refinement process
- Composition

  → Combination of various methods for a complex system

VERIFICATION-VALIDATION

- Equivalence
- Consistency
- Safety and liveness

Methods Criteria	VDM	RDP	Statemate	OMT	SART	LOTOS	SDL	Z	B	Estelle
Rigor	3	3	2	1	1	3	2	3	3	3
Data modeling	3	1	2	3	2	3	2	3	2	2
Function modeling	3	0	2	2	2	1	2	3	3	3
Control structures	2	3	3	2	2	3	2	0	2	3
TC expression	0	2	2	0	1	2	2	0	0	1
Exception handling	2	1	3	0	0	3	2	0	0	0
Verifiability	2	3	1	0	0	3	2	0	0	1
Validability	3	3	3	0	2	3	2	1	2	2
Modularity	2	1	3	3	2	2	2	2	2	1
Level of abstraction	3	1	3	2	2	2	2	3	3	1
Reusability	2	1	3	2	1	2	2	1	3	1
Implementability	2	2	2	2	1	2	3	1	2	3
Friendliness	1	2	3	3	3	1	3	2	1	2
Tool maturity	3	3	3	2	3	2	3	1	1	2
	0	3	3	2	1	3	3	1	1	2

Specific Notation

- Most are in house methods
- Often not available tools
- Do correspond to the needs
- Difficulty to interface witth other notations

Informal

- Mostly based on natural language
- Template
- Simple to use by everybody
- Problem with validation
- Example : Volere template (Natural language)

Main methods used in industry

- Semi-formal
- General purpose and dedicated (example : SDL for communication)
- Validation
  - By Simulation
  - Inspection
- Often Graphical notation
- A semantic (very rarely formal, but precise)

Formal methods

- Formal semantics (*still polemics on the issue mathematical equation and formal notation*)
- Two types
  - Model oriented : VDM, Z, B, SCR, OBJ
  - Behaviour : Petri nets, ‘‘statecharts“, Lotos, ..
- Formal validation
- Automated tool
- Not well established in industry

Abstract oriented modelling

- These are based on agebra, logic
  - Logic
  - Temporal logic and extensions for reactive systems
  - Process algebra

Semi-formal methods

- SADT
- SA-RT
- Statemate
- OOA
- OMT
- UML

Formal notations and associated methods

- See lectures
  - – Petri nets
  - – Statecharts and statemate (the statemate method)
  - – VDM
  - – Formal methods

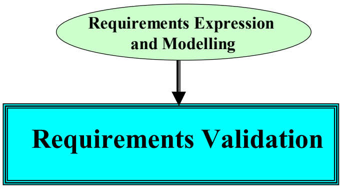
Case study and lab : Statecharts

- Consider the case study on
  - aeronautic application
  - Manufacturing
  - Communication protocols
  - Transportation

Conclusions

- Many methods and tools
- A need for taxonomy for such methods
- A need for a methdology (as UML) for using a number of methods to covers all needs for requirement specification.

Next lecture


Requirements Expression and Modelling

Requirements Validation

Paper Reading and assignments

- **Each student read at least one paper in section 8 and paper 8.D**
-
-
- M. Glinz : An integrated formal model of scenarios based on statecharts. ESEC conf., 1995
- M. Glinz : Improving the quality of requirements with scenarios. World cong on quality, sept 2000
- M. Glinz : Problems and deficiencies of UML as a requirements specification language. Proceedings of the 10th workshop on software spec and design, Nov 2000.
- C. Heitmeyer : SCR , a practical method for requirements specification. NRL report.
- C. Heitmeyer et al : Applying SCR requirement method to a weapons control panel : an experience report.