SFWR ENG 3RA3 Summary

Author: Kemal Ahmed
Instructor: Dr. Ryszard Janicki
Date: Fall 2014

Math objects made using MathType; graphs made using Winplot.

Please join GitHub and contribute to this document. There is a guide on how to do this on my GitHub.

Table of Contents

Lecture 2 – Types of Statements

Descriptive Statement: facts about the system, such as natural laws and physical constraints

- Domain Property (DOM): affecting environmental phenomena, such as physics

Prescriptive Statement: desired behavioural properties of a system; can be negotiated
Types of prescriptive statements:

- **System Requirement** (SYSREQ): when the software interacts with the other system components, i.e. environment
  - vocabulary understandable by all parties
  - Types of SYSREQ:
    - Assumptions (ASM): how the environment should be, usually through sensors and stuff
  - SOFREQ, ASM, DOM |= SYSREQ
    - When the SOFREQ, ASM, and DOM are satisfied, SYSREQ is satisfied
- **Software Requirement** (SOFREQ): relationship between a set of input variables, *I*, and *O*, the set of output variables
  - vocabulary understandable by software developers

Lecture 3

Non-functional requirements

- Look and Feel Requirements:
  - Appearance Requirements
  - Style Requirements
- Usability and Humanity Requirements:
  - Ease of Use Requirements
  - Personalization and Internationalization Requirements
  - Learning Requirements
  - Understandability and Politeness Requirements
  - Accessibility Requirements
- Performance Requirements:
  - Speed and Latency Requirements
  - Safety-Critical Requirements
  - Precision or Accuracy Requirements
  - Reliability and Availability Requirements
  - Robustness or Fault-Tolerance Requirements
  - Capacity Requirements
  - Scalability or Extensibility Requirements
  - Longevity Requirements
- Operational and Environmental Requirements:
  - Expected Physical Environment
  - Requirements for Interfacing with Adjacent Systems
  - Productization Requirements
  - Release Requirements
- Maintainability and Support Requirements:
  - Maintenance Requirements
  - Supportability Requirements
  - Adaptability Requirements
- Security Requirements:
  - Access Requirements
  - Integrity Requirements
  - Privacy Requirements

    - Audit Requirements
    - Immunity Requirements
- Cultural and Political Requirements
    - Cultural Requirements
    - Political Requirements
- Legal Requirements:
    - Compliance Requirements
    - Standards Requirements
- Open Issues: Issues that have been raised and do not yet have a conclusion
- Off-the-Shelf Solutions: is there anything that is ready made (components or full product) or even something you can copy

Lecture 5

Defining Requirements

Types of projects:

- Rabbit:
    - Agile
    - Short life
- Horse:
    - Fast, strong, dependable
    - Most common in corporate
    - Medium longevity
- Elephant:
    - Solid, strong, long life

Artifact-driven: basing the requirements on data collection, questionnaires, etc.

- You can often collect too much data
- Only keep what you need to know
- *prune* the document space, so you only keep the useful data.

Scenario: similar to *storyboards*…

Positive Scenario: behaviour system should cover

- **Normal Scenario**: everything proceeds as expected
- **Abnormal Scenario**: an unexpected behaviour

Negative Scenario: behaviour system should exclude

Knowledge Acquisition

Stakeholders: important to identify when determining who to customize the project towards

- Who is responsible for funding/using/managing the project?
- Caution: interactions with them must be done carefully

Domain expertise: what does the domain know / qualifications? Domain is who the project is directed towards

Lecture 6

Stakeholders-driven Elicitation Techniques: methods of knowledge acquisition

- Interviews
  - Single interview for multiple stakeholders: faster, but less involving
  - Steps:
    - Select stakeholders
    - Background study
    - Predesign sequence of questions, focused on concerns of present stakeholder(s)
    - Begin by asking easy questions
    - Keep focus during interview
    - Ask ‘why’-questions
    - Record answers and reactions
    - Write report from transcripts
    - Confirm report with stakeholders interviewed
  - Types:
    - **Structured**: predetermined set of questions
    - **Unstructured**: free discussion of current system
    - Optimal: start with *structure*, then do *unstructured*

Observation:

- people behave differently when observed
- slow & expensive

Group sessions: more than 4 people

Lecture 7

Inconsistencies: conflicting views or incorrect

Boundary Condition: the sample of instances where conditions conflict
Divergence: when two viewpoints have boundary conditions; they must be clarified

Strong conflict: non-satisfiable to the point of being logically inconsistent
Weak conflict: satisfiable without boundary condition

Lecture 8

Defect Detection Prevention (DDP): quantitative approach to risk analysis

Risk Trees

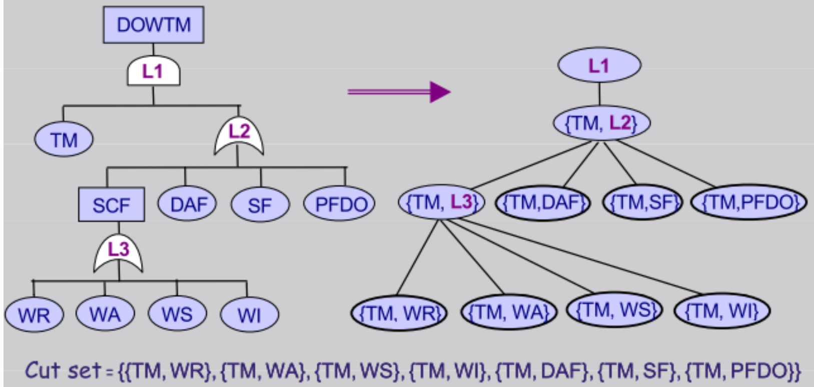
Risk Trees: a visual way of breaking down the causes of potential risks to identify where special attention needs to be placed in the design process

Components:

- Rectangles: can have children
- Ellipses: leaf nodes; may not have children
- AND / OR gates: you know how they work…

Cut Set

Cut set: the set of causes that result in the risk occurring



Cut set = {{TM, WR}, {TM, WA}, {TM, WS}, {TM, WI}, {TM, DAF}, {TM, SF}, {TM, PFDO}}

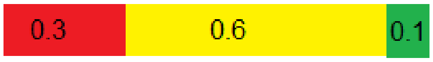
Qualitative Risk Assessment

Consequences	**Risk Likelihood** (*probability*)		
	Likely	Possible	Unlikely
risk 1	*Outcome*	*Outcome*	*Outcome*

Outcome can be Low, Moderate, High, Severe, or Catastrophic

Quantitative Risk Assessment

Consequences	**Risk Likelihood**					
	Likelihood levels 0.3		0.6		0.1	
	Likely		Possible		Unlikely	
risk 1	*Outcome*		*Outcome*		*Outcome*	

Likelihood levels: the total must equal one for the 

Analytic Hierarchy Process (AHP):
AHP Comparison Matrix

Pairwise Comparisons

This is a way of seeing if your values for your AHP matrix are consistent.
Weights: measure of importance from 0 to 1

$$w_i = \sqrt[n]{\prod_{j=1}^{n} a_{ij}}$$

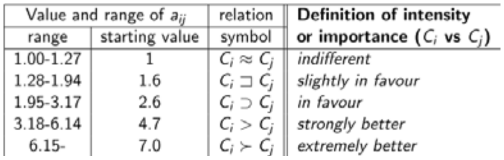
Although the sum of your weights, should equal 1, don’t worry if it doesn’t. Instead, normalize them by dividing them all by the sum of your weights.

$$x = \sum_{i=1}^{n} w_i$$

a_{xy}, where x is columns and y is rows
i, j, and k are index variables with a range of the number of elements

a: a_{ij}; so $i = x$, $j = y$
b: a_{ik}
c: a_{kj}

Inconsistency coefficient [cm$_A$]: $cm_A = \max\limits_{i,j,k}\left(\min\left(\left| 1 - \frac{a_{ij}}{a_{ik}a_{kj}} \right|, \left| 1 - \frac{a_{ik}a_{kj}}{a_{ij}} \right| \right) \right)$



Value and range of a_{ij}		relation	Definition of intensity
range	starting value	symbol	**or importance (C_i vs C_j)**
1.00-1.27	1	$C_i \approx C_j$	*indifferent*
1.28-1.94	1.6	$C_i \sqsupset C_j$	*slightly in favour*
1.95-3.17	2.6	$C_i \supset C_j$	*in favour*
3.18-6.14	4.7	$C_i > C_j$	*strongly better*
6.15-	7.0	$C_i \succ C_j$	*extremely better*

If the inconsistency coefficient is > 0.3, then you need to tweak your values.

Entity Relationship (ER) Diagram

Entity: class of concept instances
Attribute 1
…
Attribute *n*: intrinsic feature of an entity (regardless of other entities); public variables stored in the class, like hasHair or eyeColour for an Animal class

|
| relationshipName
|

Entity 2

arity: range of entities that contribute to the relationship

e.g.)

participant
Name
Address
e-mail

arity↓ |
1..* | invitedTo
| Invitation
0..* | invites

Data Flow Diagrams

Rectangles: actors outside of system who either input to or receive output from the system
Arrows: direction of flow of information, the description of the information is usually described along the length of the arrow
Circles: actions by system

State Machine Diagram

Arrow:

- [constraint]: necessary input to get to next state
- flow: what the machine is doing

Circles: description of state

All states must go to a termination state!

Lecture 17

Before-After Predicates

Before:
attribute : entity → {set of potential values of attribute}
After:
Processing based on values of attributes
e.g.
$\mathrm{hasAuthorization}(p) \land \mathrm{carriesPassport}(p) \land \neg\mathrm{inBuilding}(p) \Rightarrow$
$\mathrm{peopleInBuilding}' = \mathrm{peopleInBuilding} \cup \{p\} \land$
$\mathrm{passportsAtDesk}' = \mathrm{passportsAtDesk} \cup \{\mathrm{passportOf}\ (p)\} \land$
$\mathrm{inBuilding}(p) \land \neg\mathrm{carriesPassport}(p)$

If you (p) *have authorization and a passport and you're not in the building, then peopleInBuilding becomes peopleInBuilding + you. Also, your passport is added to the list of passports on the desk. Also, you enter the building and you're no longer carrying your passport because you handed it into the front desk.*