

The Curious Case of the Missing Communication

Towards a Normative Account of
Requirements Engineering

Amit K. Chopra
University of Trento



Communications in RE

- Much of the literature (that concerns us) ignores communication between customers (stakeholders) and engineers
- If we take a communication-oriented view, RE is a social application just the same as healthcare, business processes, and so on



The Missing Stakeholder

- The customer itself is often missing or is in the background
 - Begin with the requirements, so effectively the stakeholder has no more role to play
 - Zave and Jackson's KSR work gives the impression that the engineer *solves* for the requirements by studying the environment
 - Selection of requirements (a variant) via goal-modeling is better in that respect
 - However, using goal-models as runtime artifacts relinquishes that advantage

Goals and communications

- $\text{Req}(p) \Rightarrow \text{goal}(p)$
- Idea: $R(\text{customer}, \text{engineer}, \text{goal}(p))$
 - Setup by the communication $\text{CreateReq}(c, e, p)$
 - But I didn't like the predicate $\text{goal}(p)$
 - Because I could just as well have written $\text{goal}(c, e, p)$
- So just $R(c, e, p)$
- More generally, $R(c, e, p, q)$

Requirements and Communications

- CreateReq(c,e,p,q)
 - Performer is c
- ReleaseReq(c,e,p,q)
 - Performer is c
- CancelReq(c,e,p,q)
 - Performer is e
- Example: R(c,e,temp>15,alarmRaised)

Normative Nature of Requirements

- Requirements are *satisfied* and *violated*
- Responsibility of engineer if violated (but not automatically, only under certain conditions)
- Expressing a requirement entails a certain dialectical commitment of the customer in the sense of nonrepudiation

Domain assumptions and communication

- Dialectical commitments of the customer
 - $C^d(c, e, T, \text{alarmOperational})$

Engineer's commitments

- $C^p(e, c,$
not(violatedAss(c,e,T,alarmOperationa)),
not(violatedReq(c,e,temp >15, alarmRaised))
)
- In other words, engineer makes practical commitments
 - which he is responsible for,
 - but only to the extent that the relevant domain assumptions hold

Business Contracts



- The foregoing relationships could lead to a principled basis for devising business contracts between software development companies and customers



Requirements Evolution

- Change in requirements
 - Can be tracked by monitoring the communication
- Would imply change in the business contract
- Could form the basis of requirements management systems



Requirements negotiation

- But why limit ourselves to requirements evolution? Let's set in the broader framework of *requirements negotiation*
- Any party may want to *negotiate* in general anything in the contract, including domain assumptions and the engineer's practical commitments
- Cost, time, and other such constraints could also be take into account



Zave and Jackson's KSR

- Becomes a module that an engineer may (or may not) use before practically committing to meeting the requirements
- Specifications, unlike requirements and domain assumptions, have no privileged place in the communication-oriented view
 - They represent merely “how” an engineer would meet his commitments, but the customer need not know of them



Requirements and goals

- Which is better?

- $R(x,y,p,q)$,

- $\text{Goal}(p)$

- $\text{Goal}(x,y,p,q)$

- following the idea of goal dependencies, (although I am conceding to it R's technical advantage of conditionality)