

Requirements Engineering Domain Dimensions

Alistair G. Sutcliffe

University of Manchester

This doc gives my initial ideas on the dimensions/criteria for different genres of applications (or domains if you prefer), following my summary presentation at the Dagstuhl workshop.

The application genres were

- Engineering – control applications
 - large scale systems engineering- aircraft, chemical plants...
- Administrative systems
 - large scale social systems (engineering ??)
- User driven systems
 - unpredictable and emergent needs, individual and social scale
 - tools rather than applications
- Product-style systems- transactional
 - ERPs, market driven requirements and satisfying
- Software products discretionary
 - life style, social-ware, entertainment, education, user experience

As a taxonomy the categories are not orthogonal, e.g. office apps like Excel and be administrative and end user driven systems; however, the genres are just tools for thought about the criteria/ dimension which might usefully characterise different types of applications. My first shot follows:

Ownership- personal, organisational, governmental, single party, multi party. This begs the stakeholder question, the owner might be the purchaser/ procurer, but for social software who owns it the community (viz Facebook, Open source)

Time scale- short /long. Probably useful as it intersects with scale. Definitely a dimension but has a confound in versions, also reusable software that iterative augmented, when does one life morph into another.

Constraints from the Physical/Social world. This is motivated by Michael J's problem frames. Some systems are more bound by constraints (domain properties) emanating from the physical world whereas others suffer from human made constraints (laws and rules). Probably useful dimension Human-Physical and degree of constraint (hard-soft).

Requirements freedoms- May not be dimensions or criteria, but more like 'typical process pathways' by which products set developed: market driven, contractual bespoke, legacy, incremental design, ERP composition, design by reuse- CBSE, mash ups-EUD. Really a space of choice for developers, users and market influences. COTS, procurement literature has some input, I did a pathways article partially on this theme a long time ago Sutcliffe, A. G. (1996). A conceptual framework for requirements engineering. *Requirements Engineering*, 1(3), 170-189.

Locus of design- related to the requirements freedoms, expresses the dimension of involvement from professional to end users. Solo user, multi party communities,

2 Alistair G. Sutcliffe

Open source, SNS, web 2.0 mash ups, 2nd order design of toolkits, meta design (Fischer)

Solution dimensions- this is the intersection between requirements and architecture, i.e. the mix of development strategies: code from scratch (nobody these days), code with libraries (common), CBSE by composability, adaptability, development by application generators, intelligence in the machine- adaptation, learning also nature of the relationship between the machine and world- context awareness.

Knowledge maturity- level of maturity of knowledge in the area, in existing products, rule based or heuristic creative cognition, related to radical or incremental design, development of new IT markets (services people go on about this)

The way forward ??

At the workshop we discussed driving discovery and refinement of the criteria from examples of real systems, so we could

1. Volunteer and gather a corpus well documented application case studies- would need to be web accessible
2. Discuss and refine the initial criteria by email
3. Apply refined criteria to examples gathered in (1). This stage could be distributed among a group, we all analyse a case study we don't own
4. Bring the results together to write a journal article on the application genre criteria and their utility in guiding the RE process. Maybe a RE methods /process management framework. I would go for an REJ article

Of course it is never easy to keep the momentum going remotely, so if anyone has better ideas on collaboration, now is the time

AGS

Oct 2009