Paper presented at the 14th Monterey Workshop,
"Workshop on Innovations for Requirements Analysis:
From Stakeholders Needs to Formal Designs,"
September 10-13, 2007, Monterey, CA, USA

Discussion of Some Past and Current Approaches to Specification and to Requirements Analyses

Martin S. Feather[1]

[1] Jet Propulsion Laboratory, California Institute of Technology, 4800 Oak Grove Dr, Pasadena CA 91109-8099
Martin.S.Feather@jpl.nasa.gov

Abstract. Several decades ago Balzer at al aimed to aid users compose precise and correct specifications from informal natural language descriptions. How they approached this problem may be of interest to this workshop. The workshop, however, has a focus on requirements (as distinct from specifications). This suggests a significant difference in the issues that need to be dealt with, even though the end points – natural language, and implementations, might be the same. Some ongoing themes of research are mentioned, leading to a focus on what is clamed as an understudied area, support for requirements decision making on the basis of partial information. Goal graph work is suggested as exemplifying work in this area, and lastly the author's own work in a quantitatively-based goal graph like approach is briefly illustrated with respect to the case study blog.

Keywords:.Specification freedoms, requirements, goals, objectives, quantitative reasoning, risk-informed decision making

1 Early Work on Natural Language to *Specification* to Implementation

In the 1970s Bob Balzer was Project Leader of the Specification Acquisition From Experts (SAFE) Project at the Information Sciences Institute, University of Southern California. Some of this work is described in [1]; quoting from Balzer's biography at the end, "This project is attempting to aid users to compose precise and correct program specifications from informal natural language descriptions by resolving the ambiguity present through context provided by the system's knowledge of programs, programming, and the application domain". One of the key decisions Balzer et al made was to *design a formal specification language that was as close as possible to the way people tended to express problem specifications in natural language*. They began by establishing the principles that such a language should exhibit, documented in [2]. They then developed the language Gist to fulfill these principles [3]. Gist thus served as a stepping stone – a problem could, it was hoped, readily be converted from an informal, natural language, formulation into its formal equivalent in Gist; thereafter, program synthesis and program transformation techniques [4] could be applied to convert that specification into an efficient implementation. These ambitions

led to much work by Balzer's group, described in a retrospective paper [5]. Much of the effort was expended on the step from formal specification to implementation, an area of investigation by several research groups at that time. Challenges remain in that area, as outlined in a more recent retrospective [6]. The step from natural language to formal specification received relatively little attention once the specification language Gist had been developed. Balzer's group did some work in the reverse direction, generating natural language as a means to explain the formal specification, and to explain the results of symbolic evaluation of the formal specification.
With this history in mind, how does this relate to the workshop's focus on *requirements* analysis?

2 Requirements Analysis … Needs to Designs

The workshop's title and statement of objectives include mention of, among other things:

- Requirements analysis: From Stakeholders Needs to Formal Designs
- Precise specification (for deliberation both for "whether the specification is right" and "whether it has been properly implemented")
- Challenging requirements to express formally ("user-friendly", "easily maintainable")
- Inferring contextual information, and aiding in requirements/specifications elicitation

These, together with the blog case study, indicate that while the ultimate goal might be to arrive at a formal specification (or even a design) which will thereafter be implemented, *there is much to do to arrive at what that formal specification should be*. It is this that I interpret as the realm of "requirements analysis". A while back Steve Fickas and I argued that, just as specifications exhibit freedom from implementation concerns (a theme of the work I was involved in as a member of Balzer's group [6]), requirements exhibit freedom from specification concerns [7]. Briefly, we argued that requirements may well exhibit incompleteness, inconsistency, redundancy, ambiguity, non-uniformity, and heterogeneity, and that it is these "freedoms" that have to be removed to arrive at a specification (which will ideally be complete, consistent, non-redundant, unambiguous, uniform, and homogeneous). NOTE: there may be some different interpretation to the term "requirements" – some would argue that they are specification clauses. In the spirit of the work I did with Fickas, and (I think) with the intent of this workshop, I am treating requirements and requirements analysis as encompassing what some might term "goals", "objectives", "desiderata", etc., and fully expect that there will be the need for negotiation, compromise, iteration, decision-making, etc., to deal with them.

With this in as the focus, what does the history of Balzer's specification-centered work suggest? One possibility is to take the same tack: look at how people naturally state requirements in natural language, strive to construct a *requirements* notation that provides a no less convenient yet rigorous equivalent, and develop the tools and techniques that will operate on that language to help perform the analyses,

translations, feedback, etc. Yet scrutiny of the outcome of the history of Balzer et al's efforts suggests that while they inspired a good deal of interesting work, they did not, in the end, traverse the (in retrospect, highly ambitious) pathway all the way from natural language, through specification, to implementation. So a plausible alternative way to proceed would be to focus on the intermediate steps of going from natural language through requirements to specification: examine these steps to clarify their purpose, understand the current state of the art, and extend existing capabilities, or adapt them, or unify them etc, as the case may be, to lead to *incremental* progress.

On the formal or semi-formal representation and analysis aspects, there has been, and continues to be, many examples of work that fits this pattern (I am much less familiar with related work that focuses on the natural language aspects). For example, in the 1990's the "viewpoints" concept was being actively pursued by a number of researchers as a means to represent and deal with multiple stakeholders' differing perspectives on the software system they were trying to ultimately produce (for an overview, see [8]; for a workshop covering this area, [9]). Various studies have focused on activities in what Robinson et al termed "Requirements Interaction Management" [10] ("… the set of activities directed toward the discovery, management, and disposition of critical relationships among sets of requirements"). Much related work continues under the rubric "Goal-Oriented Requirements Engineering", as surveyed in [11] and in a follow-on [12].

So I think the workshop could yield:

- An understanding of how natural language could/should influence the ongoing body of requirements engineering research and application,
- A roadmap to improving the ongoing areas of requirements engineering research – identifying how they should be combined, validated, etc.
- A "gap analysis" to reveal which areas are relatively understudied yet worthwhile, and so in most need of attention. I believe that the work I have been involved in recently at JPL fits into one of these areas.

3 An Understudied Area – Support for Requirements Decision Making on the Basis of Partial Information

At requirements time there is often the need to make influential decisions on the basis of partial information. I assert that support for this is a relatively understudied area. Approaches that assume anything remotely close to complete capture of all relevant information to decision making will nearly always be impractical. Instead, the goal should be to make generally good decisions (albeit at the risk of missing superior solutions, or of needing to perform backtracking to revise previous decisions that turn out to lead to undesirable results). As a consequence, the need to have a "perfect" representation – one that can capture all the nuances and interactions of a problem domain – is much less of a driver than it would be for, say, a representation of a formal specification that will lead to implementation.

The goal graph representations and the uses to which they are put, as seen in the NFR[13] / i*[14] / Tropos [15] lines of work, exemplify a judicious choice of

representation and analysis capabilities. I believe the workshop's case study blog would be interesting to see represented in this style.

During recent years I have been working on an approach (called DDP [16]) that bears some similarity to the above-referenced goal graphs. It has been successfully utilized predominantly to help make technology development decisions relatively early in their lifecycle [17]. In contrast to the goal graphs work, it adopts a more restrictive graph structure, but extensively utilizes numerical calculations to help perform analyses over the accumulated data. My JPL colleague Steve Cornford who invented the DDP approach envisioned from the start the utility of a numerical basis for DDP's calculations. Despite the facts that there are known pitfalls with trying to ascribe and utilize numerical values, our positive experiences suggest that it has indeed helped teams of stakeholders arrive at superior solutions to requirements problems. The subsections that follow introduce the core concepts of DDP, and show (the start of) application of DDP to the workshop case study.

Brief introduction to DDP

A DDP model is populated by instances of three kinds of concepts: *Objectives* – what it is that the system or technology under scrutiny is to achieve, *Risks* – what could occur to impede the attainment of the Objectives, and *Mitigations* – what could be done to reduce the likelihood and/or impact of Risks[1]. In the DDP model these instances have quantitative attributes: each Objective has a *weight*, its relative importance; each Risk has a likelihood, its probability of occurrence, and each Mitigation has a cost, the cost of performing it – usually a financial cost, but other resources can also be considered, such as schedule, power, mass. Quantitative relationships connect these instances: Objectives are related to Risks, and Risks to Mitigations. Specifically, Objectives are related to Risks to indicate how much each Risk, should it occur, *impacts* (i.e., detracts from the attainment of) each Objective. Risks are quantitatively related to Mitigations, to indicate how much of a Risk-reducing *effect* a Mitigation, should it be applied, has on reducing each Risk, either by decreasing the Risk's likelihood, or by reducing the magnitude of the Risk's impacts on Objectives; the nature of the Mitigation dictates which kind of reduction takes place.

The primary purpose of constructing a DDP model is to help decide which Mitigations to perform; Mitigations incur costs and, by their Risk-reducing effects, achieve benefits (measured in terms of attainment of Objectives). DDP helps users balance these concerns when choosing Mitigations. A DDP model can also help users decide which Objectives to abandon, should it prove too costly to attain all of them.

The usual way of developing and using a DDP model is to assemble a group of relevant stakeholders in a series of facilitated sessions. The information that goes into the DDP model is elicited from those stakeholders, and thereafter they make decisions on its basis. Custom software has been developed to support application of DDP in

[1] On occasion our publications use alternate terminology such as "Requirements" in place of "Objectives", "Failure Modes" in place of "Risks", and "PACTs" – an acronym for Preventative measures, Analysis, process Controls, and Tests – in place of "Mitigations".

these sessions. A user familiar with DDP "drives" the software, and the DDP screen is projected so as to be visible to all stakeholders. As they proffer information, it is entered into DDP on-the-fly, and the status of the information is visible to all. DDP performs the calculations of cost and benefit as various selections of Mitigations are studied, and offers a form of heuristic search (using simulated annealing) to help locate near-optimal selections of Mitigations (e.g., maximize attainment of Objectives while remaining within some cost limit). DDP employs a variety of visualizations to present the accumulated information [18].

Initial application of DDP to the case study

A hasty first-cut attempt applying DDP to represent some of the information found in the blog is shown next.

DDP's *Objectives* are used to represent desiderata expressed by the "Blogger Participants". At a minimum, DDP requires for each a short title string to serve as a pithy description, e.g.,

"Invulnerable to terrorist plot"
"High passenger throughput"
"Few false positives"
"Avoid baggage mishandling"

Further information may be associated a DDP Objective, such as a lengthier textual description, and the source (i.e., indication of which one of the Bloggers provided the Objective). By default, Objectives are initialized with the same relative weight (1). These weights can be adjusted to represent priorities, e.g., if it is twice as important to achieve the Objective "Invulnerable to terrorist plot" as "High passenger throughput", then the former would be ascribed double the weight of the latter.

DDP's *Risks* are used to represent conditions and events that potentially detract from attainment of Objectives, e.g.,

"Known attack modes"
"New attack modes"
"Screeners inattentiveness"
"False alarms"
"Passengers confused about regulations"

Lengthier textual descriptions, and their sources, may be associated with Risks. By default, each Risk is initialized with an "a-priori likelihood" of 1, meaning the Risk is sure to occur unless Mitigations are applied to inhibit it. Also by default, DDP's Risks are each "atomic" objects, i.e., lack any further structure. It is possible to compose DDP's Risks into fault trees using "And", ""Or" and "Not" gates. For example, false alarms may occur in current practice, and, as suggested in the second posting, can also be caused by new technologies that are "not quite ready for deployment". Thus the "False alarms" risk could be structured into an "Or" of false alarms from conventional screening, and false alarms from new screening technologies.

DDP's *Mitigations* are used to represent the practices that reduce risks, both those in current use, and those suggested as potential improvements, e.g.,

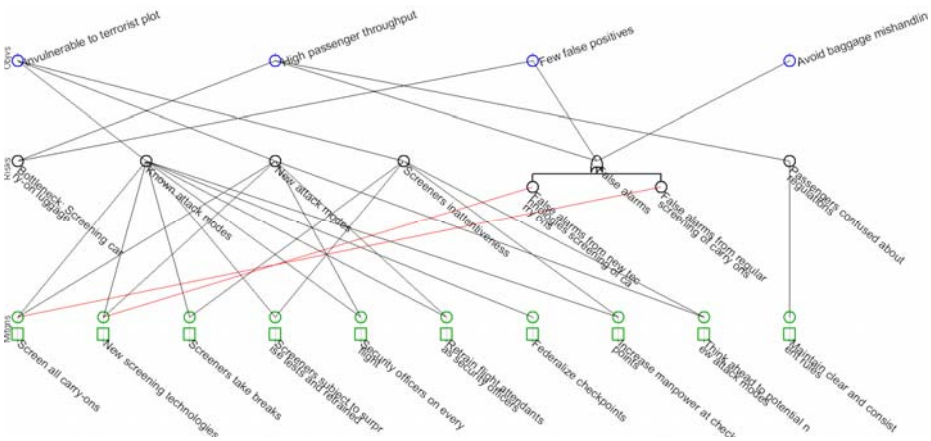
"Ban taking liquids on board"
"Screen all carry-on luggage"

"New screening technologies"
"Screeners take breaks"
"Screeners subject to surprise tests and retraining"
"Security officers on every flight"
"Retrain flight attendants as security officers"
"Federalize checkpoints"

These too can have associated textual descriptions, and sources. As the information becomes available, costs can be associated with Mitigations.

DDP then requires that these concepts be linked – Risks are linked to the Objectives whose attainment they threaten, and Mitigations are linked to the Risks they quell. Furthermore, in order to employ DDP's quantitative reasoning, these links have to be given values; a Risk-Objective link is given an impact value, the proportion of the Objective's attainment that would be lost were the Risk to occur; a Mitigation-Risk link is given an effect value, the proportion by which the Risk is reduced if the Mitigation is applied.

For purposes of illustration, I have entered into DDP some of the Objectives, Risks and Mitigations that I saw within the case study, and made some (as yet unquantified) connections among them. The figure below shows this connectivity (this is a screenshot taken from one of DDP's visualizations).



The top row, of small blue circles, represent Objectives (labeled with their title strings). The bottom row of small green circles and squares represent Mitigations (the squares are actually checkboxes on the DDP screen for displaying and controlling which of the Mitigations are chosen). The middle row of black circles and a small fault tree structure with an "Or" gate represent Risks. The lines connecting the Risks to the Objectives indicate which Objectives are impacted by which Risks (absence of a line denotes zero impact), but as yet no numerical values have been ascribed to these connections. The lines connecting the Mitigations to the Risks indicate which Risks are reduced by which Mitigations (absence of a line denotes no effect). Red colored lines indicate Mitigation-Risk connections where the Mitigation makes the Risk *worse*.

It is noteworthy that the case study blog does not contain indications of connectivity among these concepts, and provides no indication as to the magnitude of the various effects. E.g., how much improvement (reduction of risk, and thereby increase in attainment of objectives) will be gained if checkpoints are federalized? What is the cost of doing so? Without such information I assert that it is very hard to make any meaningful decisions. In the style of the goal graph work, *qualitative* valuations would be utilized – I would be interested to see the extent to which this would enable decision making. In the DDP style, *quantitative* valuations are expected, and enable various forms of decision making (e.g., identification of the biggest risks, of the objectives most under threat, of the cost-benefit tradeoffs of different selections of actions). On this basis I assert that the natural language descriptions found in the case study blog are a good start at establishing the concepts and how they interrelate, but the final comment in the blog "come up with some concrete measures" matches the need, in DDP, to ascribe quantitative valuations in order to support decision making.

Acknowledgments. Some of the work described in this paper was performed at Information Sciences Institute, University of Southern California, sponsored by DARPA, and some at the Jet Propulsion Laboratory, California Institute of Technology, under a contract with the National Aeronautics and Space Administration.

References

1. Balzer, R., Goldman, N., Wile, D.: Informality in Program Specifications. IEEE Transactions on Software Engineering, Vol. SE-4 No 2 (March 1978) 94–103
2. Balzer, R., Goldman, N.: Principles of Good Software Specification and their Implications for Specification Languages. Proc. Specifications of Reliable Software Conference, Boston, MA, April 1979 IEEE Computer Society (1979) 58–67, reprinted in Software Specification Techniques, Gehani, N., McGettrick, A.D., eds., Addison Wesley (1986) 25–39
3. Balzer, R.: Transformational Implementation: An Example. IEEE Transactions on Software Engineering, Vol. SE-7 No 1 (Jan 1981) 3–14
4. Balzer, R.: A 15 Year Perspective on Automatic Programming. IEEE Transactions on Software Engineering, Vol. SE-11 No 11 (November 1985) 1257–1268
5. Wile, D.S.: Panel Topic and Whitepaper: The Lessons of the 80's, Proceedings of a Workshop on Transformation Technology, ICSE '99 (May 1999)
6. London, P.E., Feather, M.S.: Implementing Specification Freedoms. Science of Computer Programming 2 North-Holland (1982) 91–131, reprinted in Readings in Artificial Intelligence and Software Engineering, Morgan Kaufmann (1986) 285–305
7. Feather, M.S., Fickas, S: Coping with Requirement Freedoms. Position paper for the Workshop on Intelligent Information Systems, Niagara-on-the-lake, Ontario, Canada, April 1991 University of Toronto Press, Toronto (1991) 42–46
8. Finkelstein, A., Sommerville, I.: The Viewpoints FAQ. Software Engineering Journal Vol 11 No 1 (1996) 2–4
9. Vidal L., Finkelstein, A., Spanoudakis, G., Wolf, A.L.: Proceedings of the International Workshop on Multiple Perspectives in Software Development, Part II of the Joint Proceedings of the SIGSOFT '96 Workshops, ACM (1996) 155–297

10. Robinson, W.M., Pawlowski, S.D., Volkov, V.: Requirements Interaction Management, ACM Computing Surveys, Vol 35 No 2 (June 2003) 132–190
11. van Lamsweerde, A.: Goal-Oriented Requirements Engineering: A Roundtrip from Research to Practice, 2004 IEEE International Requirements Engineering Conference, (2004), 4–7
12. Mylopoulos, J.: Goal-Oriented Requirements Engineering: Part II, Keynote, 14th IEEE Requirements Engineering Conference (2006), http://www.ifi.unizh.ch/req/events/RE06/ConferenceProgram/RE06_slides_Mylopoulos.pdf
13. Mylopoulos, J., Chung, L., Nixon, B.: Representing and Using Non-Functional Requirements : A Process-Oriented Approach, IEEE Transactions on Software Engineering Vol 18 No 6, (June 1992) 483–497
14. Yu, E.: Towards Modelling and Reasoning Support for Early-Phase Requirements Engineering, 3rd IEEE International Symposium on Requirements Engineering (Jan 1997), 226–235
15. Giogini, P., Mylopoulos, J., Sebastiani, R.: Goal-Oriented Requirements Analysis and Reasoning in the Tropos Methodology, Engineering Applications of Artificial Intelligence, Vol 18 No 2, (March 2005) 159–171
16. Feather, M.S. and Cornford, S.L. "Quantitative Risk-Based Requirements Reasoning" Requirements Engineering (2003) 8: pp. 248-263.
17. Feather, M.S., Cornford, S.L., Hicks, K.A., Johnson, K.R.: Applications of tool support for risk-informed requirements reasoning, Computer Systems Science and Engineering, Vol 20 No 1 (January 2005) 5–17
18. Feather, M.S., Cornford, Menzies, T., Kiper, J.D.: Experiences using Visualization Techniques to Present Requirements, Risks to Them, and Options for Risk Mitigation, International Workshop on Requirements Engineering Visualization (REV 2006) Minneapolis / St. Paul, MN, Sep 2006