How to Think about Customer Value in Requirements Engineering

Xinwei Zhang[1,2], Guillaume Auriol[1,2], Claude Baron[1,2], Vikas Shukla [1,2]
1 CNRS ; LAAS ; 7 avenue du colonel Roche, F-31077 Toulouse Cedex 4, France
2 Université de Toulouse ; UPS, INSA, INP, ISAE ; UT1, UTM, LAAS ; F-31077 Toulouse Cedex 4, France
{xwzhang, guillaume.auriol, claude.baron}@insa-toulouse.fr and vshukla@laas.fr

***Abstract*—Value is important for customer decisions and software design decisions. Understanding customer needs using value-focused thinking contributing to connecting customer needs and customer values and finally developing an approach of value-based Requirements Engineering. The main question of such approach is: how customer value can be reasonably quantified or measured? The ideas underlying our research are to qualify and quantify customer values on basis of the input of initial customer statements by introducing a set of techniques, e.g. multiple attributes preference theory and means-ends objectives network. In this paper, we give a preview on our proposed approach of qualitative and quantitative thinking that will enable value measurable and help make rational decision-makings.**

Keywords-customer values, requirements engineering, multiple attributes preference theory, weights

I. INTRODUCTION

It is attractive to develop and provide software products that have high levels of value to customers, and that conform to the customer needs. Then there may be certain intrinsic relationships between customer needs and customer values. There is a substantive opportunity to clarify understanding of customer values and their relationships with customer needs, which finally contributes to an approach of value-based Requirements Engineering (RE). However, many discussions on RE are about deriving software requirements from customer needs and are value-neutral in nature [1][2]. Some discussions on value in RE literature are too subjective and qualitative to be quantified reasonably [3][4]. It is also easy to confusing value with weightings or rankings of needs or requirements [5][6] that are part of elements to quantify value.

Values are what customers *fundamentally* care about in decision-making [7]. There are lots of discussions about the concept and definition about “value”, and no consensus has been achieved. We think of value in a broad sense, including preference under certainty (value in a narrow sense) and preference under uncertainty (utility). All customer statements, such as needs and expectations, function and performance requirements, constraints, goals, indicate value. But some are important to customers because they are fundamentally important to customers themselves (value) while the others are means to influence the achievement of value. Researches in decision analysis have make distinctions between value models and consequences models [7]. Value model incorporates the value or value tradeoffs and risk tolerances to evaluate consequences. Consequence models, such as performance models, model the influence relationships between design parameters to software performance.

Thus, discussions about customer values should be separated from software design solutions. The distinction is similar to the separation of *what* to do from *how* to do, or separation of *world* from *machine* in Jackson’s term [1]. We believe that understanding customer values is not so trivial and should be explored firstly in depth and width before used for software design, although iterations between customer values and means implementing customer values are always necessary.

In our approach, a set of techniques is introduced to clarify the understanding of customer needs and values. As customer statements are always expressed in different levels and granularities and some statements are even too vague to be appropriately understood, means-ends objectives network and fundamental objectives hierarchy are utilized to structure them reasonably. Then customer needs are quantified in term of value with multiple attributes preference theory. With these techniques and theory, it is then possible, for example, to model customer value and evaluate value contribution of various software solutions.

The rest of paper is structured as follows. Sections 2 gives an introduction to value-focused thinking approach. In Section 3, two techniques are introduced to structure the initially identified customer statements and to identify real customer needs. In Section 4, preference theory is utilized to construct value model with discussions about its implications for relevant problems. Section 5 outlines other related work. Finally, a conclusion is made with an outlook.

II. VALUE-FOCUSED THINKING APPROACH

Traditionally, general problem-solving approach is the alternative-focused thinking as the decision maker first focuses on alternatives, then on evaluating criteria. Value-focused thinking approach is different and proactive in nature. It first focuses on value and later on alternatives that might achieve it.

Qualitative and quantitative thinking of value is implied in value-focused thinking as shown in Fig. 1. It provides methodological basis of thinking about customer value in requirements engineering stage.

ISBN: 978-1-61208-165-6

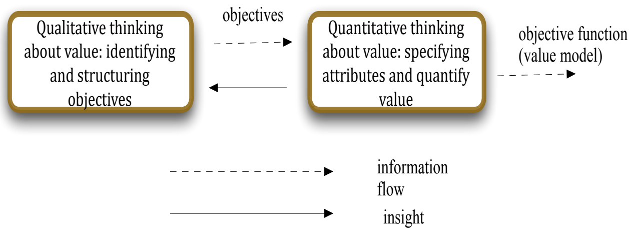
Qualitative thinking about value: identifying and structuring objectives
objectives
Quantitative thinking about value: specifying attributes and quantify value
objective function (value model)
information flow
insight

Figure 2. Qualitative and quantitative thinking of value [7]

III. UNDERSTANDING CUSTOMER NEEDS USING QUALITATIVE THINKING ABOUT VALUE

The initially elicited customer statements are usually in different levels and granularities, e.g. maximize security, access of database, sharing of information and maximize usability. “Access of database” influences negatively the security of software and is a possible means to influence the ends “maximum security”. It also is a part of “sharing of information” as others, such as “access other staffs’ files”, are also include as parts. “Sharing of information” then influences positively the usability of software and acts as one possible means of the ends “maximize usability”.

The intrinsic abstractions underlying these statements are means-ends and part-whole relationships, respectively. It is then useful to introduce some techniques encompassing these abstractions for structuring customer statements. Means-ends objectives network is utilized to trace statements in different levels and identify real customer needs hidden in statements. Fundamental objectives hierarchy is used to organize and expand the understanding of the real customer needs in different granularities. These two abstractions are also used to structure the intent specification of the software system [8].

The structuring process firstly performs means-ends analysis with customers on the initially identified customer statements. Typically, two kinds of questions are asked. One kind of questions, for example, “why this customer statement is important?” is asked to identify the ends of the statements. Then it may be always possible to ask the question of why, and may possibly arrive at statements in higher levels that may not be under current control and is not desired. An appropriate ending point to ask why question is when customers have identified the statements that are important because it is essential and important to customers in the decision context. We than obtain real customer needs in the same level. Another kind of questions, for example, “how this customer statement can be better achieved?” is asked to identify possible means to implement the customer statement, and then the first kind of questions is asked to pursue the real customer needs of these means. It is really a creatively thinking process to identify all possible means and customer needs for further exploration.

When a set of sufficiently complete customer needs in the same level has been identified, it is then necessary to clarify the understanding of each customer need in depth. There are also two kinds of questions to be asked towards customers. One is to ask such question as “what do you mean by that customer need?” It helps to identify the parts of current need. Then we may also have the problem of when to end questioning. An appropriate ending point is when suitable attributes can be selected to measure the needs in the leaves in the hierarchy. On the other side, it is also possible to ask such question as “the customer need is a part of what?” It then traces the part to the whole. After questioning, a hierarchy of customer needs with part-whole relationships is established.

It is straightforward to find that these two techniques also support top-down and bottom-up reasoning that conforms to the usual style of human problem solving. So it is cognitively attractive for qualitative understanding of customer needs and values. Intuitively, it is similar to goal-oriented RE in structuring different levels of goals and sub-goals [9]. However, more careful examination is given to identify real customer needs and to represent them in a hierarchy with collectively complete and mutually exhaustive relationships, which facilitates to verify underlying independence conditions between them.

IV. UNDERSTANDING CUSTOMER NEEDS USING QUANTITATIVE THINKING ABOUT VALUE

The bridge that connects qualification and quantification is the selection of attribute to measure the degree to which the customer need is met. However, it is a missing element in analytic hierarchy process (AHP) and quality function deployment (QFD). Then weights of customer needs are possibly assigned independently of the attributes and their range information. But according to classic utility theory, it makes no sense to say that for example in context of selecting the best suitable software to buy, minimum software cost is important than maximum performance, or vise versa. It all depends on how much you talk about cost and performance, respectively, and where you start. It is meaningful to say that cost is important than performance only when the range of change in cost from the starting point of attribute cost is more important than the range of change in performance from the starting point of attributes performance. Three kinds of attributes are usually used for measuring with different pros and cons. The procedure and criteria to be satisfied to select desired attributes are extensively discussed in [7][10][11].

After specifying attributes, it is then necessary to check the independence conditions between the set of selected attributes. There are usually several kinds of independence conditions in concert with three major function forms [12]. When additive independence is satisfied, that is, the preference order for lotteries depends only on their marginal possibility distribution, additive function form

$$u(x_1, \dots, x_M) = \sum_{i=1}^{M} k_i u_i \, (x_i) \qquad (1)$$

exsits, where u_i is a single attribute utility function over attributes $X_{i,}$ and the k_i are scaling constants subjecting to $\sum_{i-1}^{M} k_i = 1, k_i \geq 0, M \geq 2$.

This function form is similar to typically additive linear function form to calculate customer satisfaction in QFD [13][14],

$$S = \sum_{i=1}^{M} d_i s_i \qquad (2)$$

where S is overall customer satisfaction, d_i is degree of importance of the ith customer need, and s_i is degree of attainment of the jth engineering characteristic.

However, there are at least two obvious distinctions as follows:

1) k_i in equation (1) is a relative weight of attribute X_i or corresponding need. It is determined by making value trade-offs between attributes. To assess k_i, at least M equations with k_i's (i=1, …, M) as unknowns should be found and solved while it is necessary to identify a pair of two consequences $C_1=(x_1, x_2, \ldots, x_M)$ and $C^*=(x_1^*, x_2^*, \ldots, x_M^*)$ that are indifferent to customers to construct one equation. d_i, however, is usually determined by a direct weighting methods and then a normalization process without considering attributes information. Some representative methods in this kind are Analytic Hierarchy Process, 9-point direct-rating scale.

2) u_i is a single attributes utility function over attribute $X_{i,}$. It can be an increasing, decreasing or non-monotonic utility function and be of concave, linear or convex form. For example, one customer is of risk aversion over cost of buying software, the corresponding function form is a decreasing utility function with concave shape. When customer is of risk neutrality, the function form is linear and consistent with typically used function form as equation (2). In this situation, every unit of achievement of attribute has the same effect on customer satisfaction. This formalization of single attribute utility function is similar to the discussion on KANO model that distinguish three categories of customer needs with distinctive customer satisfaction relationships, respectively. In KANO model, however, there is no assessment of mathematical function between customer satisfaction and different levels of achievement of customer need and no consideration of risk attitudes toward uncertainty of attribute attainment.

It is easy to find that even in this simple function form distinctions can be made. So it is necessary to rigorously test the underlying relationships between the attributes and to verify whether certain independence conditions are satisfied.

By introducing multiple attributes preference theory, there are at least following extra benefits:

1) Design features (means) are important because of their implications to implementation of real customer needs. Then weights of the means should be derived from weights of customer needs multiplied by their contribution to the achievement of customer needs. An opportunity exists to model the relationships and function between means and achievement of customer needs,

2) Weights of needs are subjective and may be imprecise. It is then useful to do sensitivity analysis based on the available weighting information on the constructed value model,

3) Customers' group preferences can be reasonably derived from individual customer preferences by formalizing customer preferences using preference theory. It is then also possible to discuss the fairness between customers, and

4) It is also possible to improve the release planning problems by selecting a set of requirements to be implemented to maximize customer values. The *value* in cost-value approach and the *benefit* in benefit-cost ratio are then should be rectified by introducing attributes and their possibly non-linear utility functions.

V. Related Work

Researches on value in RE are far to be satisfying. The concepts, sources and dimensions of value are usually discussed in literatures without a widely acceptable definition or formulation [3][4]. Some possibly limited quantification about value focus only on weighting or prioritize requirements and resource allocation [5][6]. Several approaches are usually used in the process, e.g. AHP, QFD and cost-value approach. However, these approaches, are all controversial in their validity to model customer preferences under uncertainty. Especially, weighting and relative customer satisfaction calculation in QFD is subject to certain strong set of preference independence assumptions. It is not appropriate to use them directly without verifying preference independence assumptions among attributes.

A recent proposition that is relevant to value-based RE is value-based software engineering. It proposes a framework on the basis of "4+1" theories [15]. Four of five theories in the framework: utility theory, decision theory, dependency theory (causality) and theory W (group decision making) perform almost the same works as decision analysis does. We think it is also interesting to adopt these theories to RE stage.

Multiple attributes preference theory is in the core of modeling customer values in decision analysis, and it is especially useful when customers have multiple, conflicting needs and when there may be uncertainty in the software performance and cost. Integrating these techniques from decision analysis provides a fertile field to be explored to model customer value reasonably and effectively.

VI. Conclusions and Future works

We have presented the approach to understand customer needs using value-focused thinking. Customer values then become an explicit construct that can be modeled qualitatively and quantitatively. Weighting of customer needs and requirements are also discussed to enable a reasonable way of assigning. These give important implications to value-based RE.

However, it is found that sometimes there is difficulty to make judgments whether a statement is a means or a part of another statement, although they are obviously different in concept. Some extra researches are needed to explore the point. And the proposed approach adds cognitive and modeling burden to customers and engineers, and is time-consuming. It is expected that some reasonable simplifications or approximations can be made according to the actual application contexts, making it more practical and applicable.

We are currently preparing its applications in the RE stage of aircraft system development to test its validity. A work package "requirements establishment and value generation" is initiated and some test cases have been collected. Further results about the approach and its practical applications will be reported.

ACKNOWLEDGMENT

The research leading to these results has received funding from the European Community's Seventh Framework Programme (FP7/2007-2013) under grant agreement n◦234344.

REFERENCES

[1] M. Jackson, "The world and the machine," Proc. the 17th International Conference on Software Engineering, ACM Press, Seattle, Washington, United States, 1995.

[2] M. Jackson, "Problem frames: analyzing and structuring software development problems," Addison-Wesley Professional, 2001.

[3] R. Proynova, B. Paech, A. Wicht, and T. Wetter, "Use of personal values in requirements engineering –a research preview," Proc. FEFSQ 2011, LNCS 6182, pp. 17-22. Springer, Heidelberg.

[4] M. Kauppinen, J. Savolainen, L. Lehtola, M. Komssi, H. Tohonen, and A. Davis, "From feature development to customer value creation," Proc. the 17th International Requirements Engineering Conference, IEEE Computer Society, Atlanta, United States, 2009

[5] J. Karlsson and K. Ryan, "Prioritizing requirements using a cost-value approach," IEEE Software, vol. 14, pp. 67-74, 1997.

[6] J. Karlsson, S. Olsson, and K. Ryan, "Improved practical support for large-scale requirements prioritizing," Requirements Engineering, vol. 2, pp. 51-60, 1997.

[7] R. L. Keeney, "Value-focused thinking," Harvard University Press, Cambridge, MA, 1992.

[8] N. Leveson, "Intent specifications: an approach to building human-centered specifications," IEEE Trans. on Software Engineering, vol. 26, pp. 15-35, 2000)

[9] A.V. Lamsweerde, "Requirements engineering: from craft to discipline, Proc. the16th ACM Sigsoft Intl. Symposium on the Foundations of Software Engineering, ACM press, Atlanta , 2008.

[10] R. L. Keeney, "Common mistakes in making value trade-off," Operation Research, vol. 50, pp. 935-945, 2002.

[11] R. L. Keeney, "Selecting attributes to measure the achievement of objectives," Operation Research, vol. 53, pp. 1-11, 2005.

[12] R. L. Keene and H. Raiffa, "Decisions with multiple objectives: preferences and value trade−offs," Cambridge University Press, 1993.

[13] I. V. de Poel, "Methodological problems in QFD and directions for future development," Research in Engineering Design, vol. 18, pp. 21-36, 2007.

[14] J. R. Hauser, and D. Clausing, "The house of quality," Harvard Business Review, vol. 66, pp. 63-74, 1988.

[15] S. Biffl, A. Aurum, B. Boehm, H. Erdogmus, and P. Grunbacher, "Value-based software engineering," Springer, New York, 2006.