1-1-2009

Deriving pertinent systems requirements : a methodology

Swapam Sikdar
Ryerson University

Follow this and additional works at: <http://digitalcommons.ryerson.ca/dissertations>

Part of the [Electrical and Computer Engineering Commons](#)

Recommended Citation

Sikdar, Swapam, "Deriving pertinent systems requirements : a methodology" (2009). *Theses and dissertations*. Paper 574.

This Thesis is brought to you for free and open access by Digital Commons @ Ryerson. It has been accepted for inclusion in Theses and dissertations by an authorized administrator of Digital Commons @ Ryerson. For more information, please contact bcameron@ryerson.ca.

DERIVING PERTINENT SYSTEMS REQUIREMENTS A METHODOLOGY

by
Swapan Sikdar

MBA, Indian Institute of Management, Ahmedabad, India, 1980
Bachelor in Electronics and Communication Engineering, University of Roorkee,
Roorkee, India ,1978

A thesis
presented to Ryerson University
in partial fulfillment of the requirements for the degree of
Master of Applied Science
in the Program of
Electrical and Computer Engineering

Toronto, Ontario, Canada, 2009
© Swapan Sikdar 2009

**PROPERTY OF
RYERSON UNIVERSITY LIBRARY**

Author's Declaration

I hereby declare that I am the sole author of this thesis.

I authorize Ryerson University to lend this thesis to other institutions or individuals for the purpose of scholarly research.

Signature

I further authorize Ryerson University to reproduce this thesis by photocopying or by other means, in total or in part, at the request of other institutions or individuals for the purpose of scholarly research.

Signature

Abstract

Deriving Pertinent Systems Requirements A Methodology

© Swapan Sikdar (2009)
Master of Applied Science
Electrical and Computer Engineering
Ryerson University

Poor understanding of needs results in incompletely captured requirements and causes project failures. Analysts and developers by training, practice preoccupation and lacking suitable methodologies are ill equipped to capture various dimensions of requirements. In early stages needs are vaguely expressed. They have to be extracted and reasoned with stakeholders using relevant vocabulary. Business reality limits time at stakeholders' disposal to participate in requirements development. Methodologies that are more workable are required.

Working with early stage goal oriented concepts from Goal Oriented Requirements Engineering (GORE) and fuzzy set theory based value aggregation, this thesis proposes a methodology to develop and select pertinent requirements. We use GORE concepts for identifying requirements and fuzzy aggregation to select alternatives. Use of fuzzy set theory and Ordered Weighted Aggregation (OWA) operators allows quantitative structuring of early stage decision problem consistent with human thinking and reasoning. The methodology is illustrated via two case studies.

Acknowledgements

I am deeply grateful to several individuals and professors who offered valuable insights and support during my studies and research at the University. I am thankful to the Department and the University for offering open environment to make my transition from practice to academic world. I would like to thank Professor Alagan Anpalagan for support and encouragement that I received during the program.

It was my privilege of working with Professor Olivia Das as my supervisor. My deepest gratitude and thanks to you for introducing me to research writing; your engaging interest in reviewing my ongoing work and research ideas and the encouragement I received from time to time. I found the interactions very rewarding, coming back clear and unstuck after the interactions.

Most of all, I was able to move ahead in my academic and research pursuits due to whole hearted support of my family, and Soma my wife taking up financial responsibility while I moved from work to school.

With deep gratitude, I acknowledge the support I received from NSERC.

Deriving Pertinent Systems Requirements A Methodology

Table of Contents

1	Introduction and Research Motivation.....	1
1.1	Problem Dimension	3
1.1.1	Requirements Pertinence	3
1.1.2	Requirements Selection	4
1.2	Thesis Contribution.....	5
1.3	Overview of Proposed Method	6
1.4	Thesis Organization	7
2	Background.....	9
2.1	Requirements and Requirements Development Process.....	9
2.2	Related Requirements Engineering Methods.....	11
2.2.1	Goals and Goal Oriented Requirements Engineering (GORE)	13
2.2.2	Non GORE Approaches.....	17
2.3	Organization Modeling Methods	18
2.4	Fuzzy Sets Basics in Decision Making.....	20
3	Approach for Identifying Pertinent Requirements.....	25
3.1	Source of Pertinence	26
3.2	Approach for Identifying Pertinent Requirements.....	27
3.3	Case Study I	29
3.4	Discussion	34
4	Requirements Selection	36
4.1	Outline of Proposed Method.....	36
4.2	Theoretical Background for Proposed Method.....	37
4.2.1	Decision Making and Nature of Evaluation Criteria	37
4.2.2	Ordered Weighted Averaging Operator	39
4.2.3	Linguistic Quantifiers	43
4.2.4	Aggregation with Criteria Importance	43
4.3	Establishing Decision Criteria	44
4.4	Case Study II.....	46
4.5	Discussion	59
5	Conclusion	62
6	Bibliography	63

Deriving Pertinent Systems Requirements A Methodology

List of Figures

Figure 1: Requirements Development Process	4
Figure 2: Thesis Outline.....	7
Figure 3: Notations Used	14
Figure 4: Goal Task Tree	15
Figure 5: Henderson-Venkataraman Alignment Model	19
Figure 6: Zachman Framework ^{TM}	20
Figure 7: Sample Fuzzy Set Membership Function.....	22
Figure 8: Union Operation on Fuzzy Sets.....	23
Figure 9: Intersection Operation on Fuzzy Sets.....	23
Figure 10: Exploring Context's Influence on Requirements.....	26
Figure 11: Exploring Influences on Project Goal-Tasks.....	28
Figure 12: Case Study I Actor Dependencies	30
Figure 13: Case Study I- Goal Tree	32
Figure 14: Case Study I- Updated Goal Tree.....	33
Figure 15: Case Study I Goal Task Decomposition.....	33
Figure 16: Methodology for Selection.....	37
Figure 17: Case Study II Indicative Goal Tree (functional goals).....	48
Figure 18: Case Study II- Indicative Implementation Goals	48
Figure 19: Case Study II- Evaluation Attributes Framework	49

List of Tables

Table 1: Prominent Project Success Factors.....	2
Table 2: Life Cycle Processes per IEEE 12207- 2008 Standards.....	11
Table 3: Case Study I - Actors and Roles	30
Table 4; Case Study I - Stakeholder Motivations and Resulting Goals.....	31
Table 5: Case Study I- Goal Decomposition	32
Table 6: Types of Attribute Assessments	38
Table 7: Case Study II- Decision Structure and Attribute Importance	50
Table 8: Case Study II - Aggregation Parameters	50
Table 9: Case Study II- Alternatives' Goal Satisfaction Assessment	52

1 Introduction and Research Motivation

Inadequate understanding of needs, poor scope definition and resulting requirements are common contributors for information systems project failures. Existing approaches for scope and requirements development have modeling gaps that impact determination of right scope and requirements. This work discusses an approach applicable at early stage of projects, aimed to narrow the gap. This could help us derive and select pertinent requirements and thereby curtail potential waste in downstream activities.

Last two decades saw increased consciousness in studying IT project failures, causes and impact. While research and practice effort devoted to causal investigation and mitigation methods have benefited the development and user community, researchers still assess failures to be of a high degree [13].

Yardstick for project success is fuzzy and perspectives of developers may vary from those sponsoring the projects. Project sponsors' concerns are focused on accomplishment of business objectives. Developers may consider delivering a robust product meeting signed off requirements as successful. There may be ongoing gaps between stakeholders and developers on what the needs are and what the requirements signify. Project Management Body of knowledge (PMBOK) [17] a widely consensual compendium of good project management processes observes that project success is measured by "product and project quality, timeliness, budget compliance, and degree of customer satisfaction" and suggests establishment of measurable success criteria in "project charter" which should happen at the stage of project conception. Customer satisfaction is dependent on meeting the objectives for which the project was initiated.

Much of the accessible research on causes of success and failure represents perceptions of developer practitioners. We do have general assessments from perspectives of leadership and project sponsors. While most research is based on surveys and not walkthroughs and analysis,

practice observations and analysis are visible in various research papers on related topics. From Verner et al's [44] study of failures from developers' perspective we abstract the dominant causes to include Disagreement on Schedules, Poor Project Management, Wrong estimation of efforts and so on. Boehm et al [7] found risk management to be dominant contributor in controlling cost overruns, a measure of project success. Berntsson-Svensson et al [5] explored factors for project success and inferred that Comprehension of problem, Customer Involvement (not user), Completeness of the Requirements are strong contributors for success. Procaccino et al [29] observe need for realistic expectations for project success. It is commonly understood that a clutch of factors are responsible for project success or failure, though a single factor may be dominant for a given project. We capture some of the factors, observed dominant by researchers in our abstraction in Table 1.

Table 1: Prominent Project Success Factors

Category of Factor	Examples
A. Process and Methods	
<i>A1. Requirements Development</i>	Complete and accurate requirements [5] Enough time for elicitation Requirements are clear and understood by team [30] Consistent , traceable and testable requirements [24]
<i>A2. Project Scope Definition</i>	Well defined project scope Clarity of business problem [5], project vision [45]
<i>A3. Planning</i>	Estimation and Planning [44] [45]
<i>A4. Collaboration</i>	Collaborative review and feedback process [29] Sufficient access to business/technical experts [29]
<i>A5. Project Management</i>	Management of changes Management of risks [7] Customer/users have realistic expectations [29]
B. Team and Work Infrastructure	Team size and skill fit Team motivation
C. Environment Complexity	Technical complexity Organization Politics Volatility of Requirements Complexity of Implementation and Change Management

It is intuitive common wisdom that good comprehension of project objectives and requirements plays a dominant role in project success. This work seeks to strengthen the

methods to uncover and understand needs and thus support development of appropriate scope and requirements (category A1 & A2 above)¹. Our focus is a class of information systems in typically commercial environments where time to turnaround systems from initial need statement to system deployment is short. *Specifically the work seeks to address problem of finding and selecting pertinent requirements in business-real time.* This thesis builds on our previous work [36], where we identified need for supporting tools that can be used by developer/analyst community to get pertinent requirements and [37] where we proposed a pertinence exploration approach under simplistic assumptions.

1.1 Problem Dimension

1.1.1 Requirements Pertinence

Researched and rigorously developed methodologies and guidelines are available to practitioners about writing well-formed specifications of requirements. Detail in specification however does not indicate if the requirement is captured correctly and specified completely. Yue [52] observed that completeness requires sufficiency as well as pertinence. A pertinent requirement is one that is necessary in achievement of goal, at least in one of the alternative ways the goal can be achieved. Yue defined pertinence rigorously in [52].

IEEE Guide for Developing Systems Requirements (IEEE 1223a-1998) recommends a process outlined in Figure 1. In the figure, we show path 1 as the path most rigorously traveled during requirements development. Early stage elicitation and modeling clarify the requirement. Subsequent analysis, modeling, validation, refine the requirements. The refined requirement set needs sorting out for dependencies, consistency and feasibility and a process of negotiation, prioritization and cost-benefit analysis may be used to arrive at a chosen set to which systems will be ultimately developed, assembled or configured. The steps are not necessarily discrete and there may be iterations amongst the steps. In practice however, *immediate concerns or needs* as expressed by user/ stakeholder dominate the requirement development (path 1 in Figure 1). Customer and environment influences as narrated to analysts, are frequently assumed valid, and there is a limited exploration of path 2.

---

¹ We chose to depict some other categories in Table 1 above for their relevance in requirements selection, which we cover later in this work.

Figure 1: Requirements Development Process

An assumption if any that the initially expressed need or system concept represents organizationally pertinent need is not necessarily valid, as indicated by review of project failures and identification of scope and requirement completeness factors for project success [5] [30]. Unless related back to the organization's context, the downstream decomposition and refinement may be divergent to what the organization missions to achieve or may be valuable only in limited context.

Methods available to analysts for exploring path 2 identified require a degree of iteration and wide span of interaction in order to grasp various dimensions of a need from given raw need, limiting their applicability in early stages and certainly for the class of commercial systems in focus.

1.1.2 Requirements Selection

Real life complexities and (business) dynamism in organization's wider context add dimensions of complexity to arriving at pertinent requirements. A specific window of opportunity, Competitor's response, Change in Owner-Stakeholder expectations, Competitiveness of the market, Shifting patterns of buyer-customers' expectations and so forth are factors that place bounds of timeliness and budget for project utility. This dynamism, which requires responsive architecture (not a subject of this thesis) demands timeliness of scope and requirements development and reasoning out course of actions amongst the varied alternatives that may exist.

Existing methods include prescriptive good practices in Project Management, Business IT alignment, Architecture as well as those oriented to requirements prioritization. Some of these practices imply a degree of analysis that may be in conflict with abovementioned timeliness or may suit only process mature organizations. Frequently practiced methods then are oriented to cost benefit type of analysis, which tend to compromise considerations on several other factors such as those identified in Table 1 above.

Both of the problems identified above are generic. Though recognized as critical as in Table 1, practitioners' training texts cover it poorly. Misic et al's 1998 [27] examination of 29 commonly used Systems Analysis texts showed that 40% of texts had no coverage for "Define scope and objectives of systems/projects" one of the 34 identified tasks. The rest had less than 1% coverage of the topic. The texts' coverage accorded poor 29th and 30th rank for "Define scope and objectives of systems/projects" and "Determine impact of new system on organization and users" respectively, the very reasons for which systems are developed. A later study by the Misic et al [26] exposed this irony when the study determined that these were top two ranked requirements in terms of importance in real life practice.

1.2 Thesis Contribution

Motivated by need to develop requirements aligned to organization goals in a business-real time, the thesis provides an early stage methodology for capturing pertinent requirement alternatives and comprehensively evaluating them for overall goal satisfaction:

We believe our methodology using fuzzy aggregation for overall goal satisfaction is first application in requirements selection covering comprehensive assessment on functional, nonfunctional and implementation criteria. The methodology for quantitative evaluation of alternatives is consistent with human reasoning and decision-making.

Proposed methods are applicable in contexts where costs or time required in detailed stakeholder interviews for requirements analysis is high or fidelity of such process is suspect. This would include a large class of commercial systems.

1.3 Overview of Proposed Method

Requirements exist to solve a business need. The need is shaped by organization's, goals, stakeholder intensions, strategies and changes in its environment. In proposed approach, our initial target is to derive a set of "business components", the higher-level abstractions of requirements that will solve the business need. Once we arrive at the potential alternatives, we evaluate them for potential goal satisfaction on desired criteria to select the best alternative. The methodology in brief is a follows:

- Expressed business need is explored via evaluation of influences by organization and stakeholder goals. This uncovers related business needs otherwise implicit.
- This set of redefined needs is decomposed via means-ends analysis leading to identification of required business components. The required business components are decomposed to tasks that would define requirements at a lower level.
- Multiple configurations of business components could address the business need. The identified configurations are evaluated for selection. We select based on assessed (potential) satisfaction of goals by the alternative.
  - We analyze the decision parameters and structure the criteria set
  - We analyze decision goals and attitude, and create applicable decision model
  - We evaluate each alternative using the decision model and select what fits our decision goals

Selected configuration is then available for elaboration and requirements specification using applicable techniques (not discussed in this thesis). During elaboration if there are any changes to high-level requirements, constraints or goals, the process is iterated to arrive at next set. There may be additional activity of optimization, when evaluated alternatives do not meet the goals; this again is not discussed in this thesis.

1.4 Thesis Organization

Rest of this thesis is organized in five chapters and bibliography.

Figure 2: Thesis Outline

We cover the preliminaries of requirements development and background material relevant to our approach in Chapter 2. We introduce the Goal Oriented Requirements methods, as well as organization modeling methods that are used in our approach for pertinent requirement development. We discuss other work that seeks to address similar problems. We also introduce the basics of fuzzy sets in this chapter to form potential support in our aim to address selection problem.

Chapter 3 introduces the environmental influences that impact requirements and develops the approach to incorporate these influences in identifying pertinent requirements. We demonstrate the approach steps via a case study for a consumer packaged goods context.

Chapter 4 discusses evaluation and decision making in requirements and alternatives selection. The chapter discusses the fuzzy aggregation and linguistic concepts that enable desired decision modeling. We demonstrate the steps via a case study of a provincial healthcare initiative.

We review the approach with respect to other real life demands that the approach must address and reflect on potential directions in Chapter 5.

2 Background

This section briefly outlines the nature of requirements and requirements development within the overall software engineering lifecycle processes. It further outlines existing Requirements Engineering (RE) methods that relates to seeking pertinence in development and reasoning of requirements (Section 2.2). The section also covers the basics of Goals and Goal Oriented methods on which we build our approach. Work in accompanying disciplines of organization modeling and architecture is covered in Section 2.3. We cover basics of fuzzy set math that our requirements selection approach uses, in Section 2.4. Important terms and notations used elsewhere in the document are introduced within these sections.

Before reviewing existing work, it is worth highlighting that scope determination and identification of stakeholders and requirements cannot be done in isolation. It must be done in the context of organization's technical infrastructure and skills. Organization's systems engineering activities must dovetail with above activity. While needs seed the requirements, the technical capabilities and constraints may reshape the needs and scope. PMBOK [17] recommends determination of scope at the stage of project charter.

2.1 Requirements and Requirements Development Process

Per IEEE standard (IEEE Std 610.12-1990) Requirements are defined as

- (A) A condition or capability needed by a user to solve a problem or achieve an objective
- (B) A condition or capability that must be met or possessed by a system or system component to satisfy a contract, standard, specification, or other formally imposed document
- (C) A documented representation of a condition or capability as in definition (A) or (B)

Notably in system development context, concept of “user” leads to various viewpoints including those who sponsor the project, who interact with the system and who use the results of the system. They all are stakeholders in the project. 2008 published IEEE Standard on Systems and Software Engineering -Software Life Cycle Processes (IEEE Std 12207-2008) views the requirements in overall organization context and elaborates the stakeholder requirements to include “the needs and requirements imposed by society, the constraints imposed by an acquiring organization and the capabilities and operational characteristics of users and operator staff”.

Elaborations for various stakeholder roles require different level of details in requirements specification. Business view of requirements abstracts high level objective with limited narrative, while from developer’s view it must be populated with pre and post condition event description, constraints, exceptions, underlying objects and so forth. Requirements represent functional capability of what must be accomplished as well as user or technically oriented quality attributes (e.g. usability, maintainability) that the system must possess. Second category of requirements is commonly referred as Non Functional Requirements.

This work concerns *development of appropriate requirements*, and within the overall life cycle processes defined by the life cycle process standard Table 2, the scope of this work is within the following Technical Processes under System Context Process category:

1. Stakeholder Requirements Definition Process (clause 6.4.1 of the standard)
2. System Requirements Analysis Process (clause 6.4.2 of the standard)

The standard mandates the above processes to include identification of Stakeholder and Requirements, definition of constraints, evaluation of requirements, agreement, recording and specification of requirements.

Table 2: Life Cycle Processes per IEEE 12207- 2008 Standards

System Context Processes			Software Specific Processes	
6.1 Agreement Processes 6.1.1 Acquisition Process 6.1.2 Supply Process	6.3 Project Processes 6.3.1 Project Planning Process 6.3.2 Project Assessment and Control Process 6.3.3 Decision Management Process 6.3.4 Risk Management Process 6.3.5 Configuration Management Process 6.3.6 Information Management Process 6.3.7 Measurement Process	6.4 Technical Processes <i>6.4.1 Stakeholder Requirements Definition Process</i> <i>6.4.2 System Requirements Analysis Process</i> 6.4.3 System Architectural Design Process 6.4.4 Implementation Process 6.4.5 System Integration Process 6.4.6 System Qualification Testing Process 6.4.7 Software Installation Process 6.4.8 Software Acceptance Support Process 6.4.9 Software Operation Process 6.4.10 Software Maintenance Process 6.4.11 Software Disposal Process	7.1 Software Implementation Processes 7.1.1 Software Implementation Process 7.1.2 Software Requirements Analysis Process 7.1.3 Software Architectural Design Process 7.1.4 Software Detailed Design Process 7.1.5 Software Construction Process 7.1.6 Software Integration Process 7.1.7 Software Qualification Testing Process	7.2 SW Support Processes 7.2.1 Software Documentation Management Process 7.2.2 Software Configuration Management Process 7.2.3 Software Quality Assurance Process 7.2.4 Software Verification Process 7.2.5 Software Validation Process 7.2.6 Software Review Process 7.2.7 Software Audit Process 7.2.8 Software Problem Resolution Process
6.2 Supply Process 6.2 Organizational Project-Enabling Processes 6.2.1 Life Cycle Model Management Process 6.2.2 Infrastructure Management Process 6.2.3 Project Portfolio Management Process 6.2.4 Human Resource Management Process 6.2.5 Quality Management Process			7.3 Software Reuse Processes 7.3.1 Domain Engineering Process 7.3.2 Reuse Asset Management Process. 7.3.3 Reuse Program Management Process	

Notes:

1. Numerals before the process represent the respective clause number of the Standard
2. This Thesis scope relates to clause 6.4.1 and 6.4.2 highlighted above

2.2 Related Requirements Engineering Methods

This section reviews Requirements Engineering methods that relates to seeking pertinence in development and reasoning of requirements. The scope of this thesis is large systems development. The development notion also includes acquisition and configuration of ready applications, application re-engineering apart from fresh development. Further the presumed reality for this thesis is that of dynamic business environment that allows limited time for analysis. Hence, some of the iterative methods that rely on series of prototypes for requirements development are not reviewed here. Methodology and modeling support for active exploration of

stakeholder objectives is core to seeking pertinence and this section reviews existing methods supporting the same perspective. For large real life systems, commonly used methodologies are prescriptive and inquiry based, requiring the analyst to seek answers from relevant stakeholders about objectives.

Commonly used methods of requirements development and analysis are Structured Systems Analysis and Design (SSAD) [12] and Object Oriented Analysis and Design (OOAD) [8]. They have been practiced since 80s. The methods operate at atomic level of data, process or objects. The methods are not directly usable in identification of expectations and concerns outside the given process, data or objects. For instance in commonly used Data Flow Diagrams (DFD) based structured analysis methods, the view is focused on flow and transformation of data through various steps of intended business process. Why a certain business process should be adopted is discovered by enquiry and analysis. Identification of stakeholders that have relationship with the data (as processor-users or external recipient-users or external sources) is primarily driven by the analysis of proposed business process. Modeling support for seeking out the objectives of the stakeholders does not exist. Modeling for stakeholder expectations have to be stretched from logical dialog outlines and DFDs.

Formalized Objects in OOAD methodologies embody a role (services assigned to the object), relationship properties as well as atomic data of the object. Sequence and Collaboration diagrams model the interactions between the objects. Process is captured by behavioral descriptions such as Use Cases or Scenarios along with Sequence and Collaboration diagrams. While entity relationship provides semantically close to reality model and the stakeholders roles can be modeled as objects, there is no direct modeling support for identification of stakeholder interests and rationale for selecting a proposed business process.

Elegant modeling in these approaches is amenable to efficient conversion in to design and software artifacts. However the limited support for modeling stakeholder goals gives the system concept refinement a craftsman like activity based on analyst's awareness of the context and application of her/his heuristics. Requirements exist for a goal (of real business need), the approach to analyzing the requirements continued to be informal analysis or enquiries based. The OOAD methods however were extended by incorporation of goals and stakeholder intents in a set of approaches that are now termed as Goal Oriented Requirements Engineering (GORE) approaches discussed below.

2.2.1 Goals and Goal Oriented Requirements Engineering (GORE)

Synonyms of Goal Directed, Goal Based, Goal Driven have also been used for Goal Oriented approaches. GORE approaches are requirements engineering methods whereby RE tasks of requirements identification, elicitation, elaboration, validation, reasoning and so forth are done on the basis of goals in the context. Having Goals in the specification is argued to provide completeness to requirements [52]. Formalized GORE concepts started appearing in research publications in early 90s. Drive behind such modeling approaches being inclusion of individual requirements' rationale in accomplishing system objective.

2.2.1.1 Goals, Softgoals, Subgoals, Tasks and Notations

Dardene et al [11] viewed goal as a non-operational objective to be achieved by the composite system. Anton visualized Goals as higher level objectives of the business or system entity, which captures rationale for development of a system [2]; the associated requirements providing the "how" of goal accomplishment. Lamsweerde [43] provides a concise definition of Goal as below.

Definition: "A *Goal* is an objective the system under consideration should achieve"

Lamsweerde further observed that software and its environment that includes human and devices compose the system. Thus, software, devices and humans are considered agents in the system. A Goal could be considered an objective that is accomplished by co-operation between the agents, while a requirement could be viewed as something at more atomic level namely "an objective that can be accomplished by a single agent".

There are goals that can not be fully achieved. This may be because the goal itself can not be fully defined or/and the strategies to meet the goal are not fully comprehensible to start with. Let us take the example of a goal such as "the system should provide the client a good service experience". What is "service experience" and what is "good" is not easily definable. Further it is not possible to map and define all possible tasks that need to be completed to satisfy "good service experience". Such goals are termed as *Softgoals*. Many of nonfunctional requirements (e.g. "user friendliness", "flexibility for modification") and high-level functional business

requirements (e.g. “customer service”, “inventory replenishment timeliness”) are of softgoal nature and can be satisfied to a degree only.

The document uses following notations (Figure 3), which are taken from User Requirements Notation [33] consistent with Goal-oriented Requirement Language (GRL) notations and now part of 2008 adapted International Telecommunications Union’s Recommendation Z.151.

Figure 3: Notations Used

A (soft)goal may be composed of component goals achievement of which mean achievement of goal. These component goals are termed as *subgoals*. Alternative subgoals (and tasks) may exist to accomplish a goal, or all subgoals (and tasks) may need to be accomplished or an accomplishment of some subgoals (and tasks) may lead to partial accomplishment of goal. Thus the tasks, subgoals, goals have means-end as well as composition relationship. Figure 4 shows a goal-task tree that is useful in depicting such relationships. In this example (adapted from [33]), the system security softgoal is achieved when Process, Network and Application security softgoals are accomplished and if any of these is not accomplished the system is insecure. Thus the subgoals of Process, Network and Application security have a AND relationship with System Security goal. In the example Authentication and Message encryption show a composition relationship and achieving either leads to partial achievement of Application security. Authentication via Password or via Biometric means are considered to be independent OR methods either of which is required to achieve authentication.

Published literature terms *operationalization* of a goal as conversion in to specifications.

Figure 4: Goal Task Tree

2.2.1.2 GORE Approaches

Notable GORE approaches include Goal Based Requirements Analysis Method (GBRAM) [1], [2], Knowledge Acquisition in AuOtomated Specification (KAOS) [11], intention driven i*/Tropos [10]. These are briefly discussed below.

GBRAM

GBRAM uses enquiry based approach for requirements analysis. Arguing that goals are not explicit and have to be discovered prior to requirements development, the approach leans on analysis of existing artifacts such as starting narrative of the need, business documents, flow charts, ER diagrams for identification of goals and constraints. Identified goals are then elaborated by finding obstacle that could prevent goal accomplishment and analyzing scenarios (behavioral description of system) to find alternatives. This leads to steps of goal refinement whereby goals are rationalized by removing redundant ones, reconciling “Synonymous goals” and categorizing and priority ordering of goals. Thus preference guidance is available during reasoning. With sufficiently elaborated detail, entity or process agents are identified and assigned responsibilities for the goals. Modeling constructs include schema for goals that include the goal, category, agent, stakeholder, obstacles, constraints and scenarios. This artifact is available for

conversion into requirements specification. Overall the method starts with available narrative of the need, and then goes through discovery of goals and evolution with sufficient elaboration, which in turn is converted into requirements.

KAOS

One of the early formalized GORE approaches is KAOS [11], a modeling method with own textual grammar for specification, graphical constructs and strategies for elaboration of requirement models from initial goals. The method is enabled by a software that provides guidance and automation support for elaborating requirement model. The method conceptualizes a meta model with meta concepts of goal, agent, action, relationship, constraint and so forth, and gradually acquires domain specific instantiation via the method's strategies. In KAOS graphical syntax the meta concepts and their instances are represented as nodes while the edges represent structuring links. One starts with Goal meta concept and discovers it in the domain context via potential reduction strategies. Abstract Goals have to be refined to formal concrete goals, which in turn could be reduced to subgoals. Reduction is done via AND-OR relationships of subgoals to parent goal, Objects (including agents, events, entities) concerned with the goal are next discovered. Agents and Capabilities instances are considered and elaborated in the domain; following which actions to meet the constraints are identified. This process may call for back review of Objects and Actions as new concepts may emerge. After refinement, actions are strengthened to meet exception conditions. Subsequently AND-Or responsibility links between agents and constraints are explored for alternatives in Agent responsibilities. At each stage, graphical representations are rigorously defined and formally specified in KAOS specific language. Thus, rationale goals and constraints are the original sources from which assigned actions (operational requirements) evolve.

***i**/Tropos**

Tropos [10], [15] is a goal oriented methodology providing support for goal based early requirements development and subsequent elaboration and formal specification of requirements using its' own grammar and potential support for architecture design. For this work the method is discussed only in requirements analysis context. Tropos uses *i** [50] [51] agent-modeling method, which is composed of two core concepts namely "Strategic Dependency (SD)" and "Strategic Rationale (SR)". SD model is a network of dependencies between the actors. Actors in organizational settings could be Agents (human or system component), Role (an abstract

responsibility) or Position (a socially recognized role). Actors in an environment depend on others (“dependee”) to get “dependum” to accomplish some goal that otherwise is not achievable fully or is partially achievable by itself. SR modeling provides a method of describing the relationship between elements such as goals, subgoals, tasks, resources via means-ends and composition type links. In Tropos, first a SD model describing the relationship amongst the actors, goals, tasks or resources is captured. SR analysis is conducted on actor dependency graph that was arrived by SD modeling. This leads to elaboration of requirements into components and generates alternatives. No specific approach is indicated for exploring the context and implicit influences.

2.2.2 Non GORE Approaches

By themselves goal oriented approaches provide a modeling formalism and notations to capture goals and requirements relationship. As identified in the Section 1.1.1, our aim is to explore ways to improve requirements identification in the context of stakeholder goals and influences. We review some of the proposals in this regard.

Niu et al [28] provide a rigorous and formal approach for context analysis. The approach uses Repertory Grid Technique (RGT) to map stakeholder vocabularies for facilitating communication and identify concerns and uses Formal Component Analysis (FCA) to uncover inconsistencies and restructure the requirements. Large system contexts with large but stable stakeholder base will find good application of this rigorous methodology. Effort in the process is non-trivial and potentially time consuming. We borrow the philosophy of common understanding behind the rigorous stakeholder vocabulary mapping of the approach in our method that aims fast even if simple early assessment.

Kotonya et al’s [24] Viewpoints Oriented Analysis provides structured path of walking through identification, elicitation and reasoning amongst viewpoints and operates at various levels of abstraction. Viewpoint philosophy is integrated in current GORE approaches. Stakeholder identification itself is a critical activity that must be consciously done. Sharp et al [35] observed that assumption that the stakeholders are obvious ones, the immediate interaction points is not valid. Their proposal suggests viewing each immediate stakeholder be explored for their own “supplier”, “client” and satellite relationships thus uncovering next level of stakeholders, who in turn need to be explored their stakeholders thus uncovering a web of stakeholders. We find that i^*

SD model, discussed previously is based on relationship concept and provides us sufficient formalism for our need.

Scenarios [20] [31] have been used to uncover requirements from various perspectives. Scenarios (representation of potential process steps) provide more tangible artifacts and belief is that walking through scenarios with stakeholders provides a better chance for uncovering requirements and context's implicit influences. Proposals specifically motivated to explore contextual operationalization [41] are based on scenarios. Scenarios are powerful and apt for identifying process details, exceptions, actors and so forth in the elaboration stage. At initial stage of identifying conceptual configuration of components, scenarios are complex to compose with adequate coverage, as there is still vagueness about intentions and goals behind the stated needs. Proposals exist for combining Goals and Scenario based exploration, primarily operationalizing goals via scenarios. Rolland et al [32] propose a formalization of goal scenario combination, which aims to discover subgoals via scenario authored for the goal. We consider "why" interrogatives and means end analysis of Goal Oriented approach as a suitable method for our approach.

2.3 Organization Modeling Methods

A body of work motivated by factors including pertinence in accompanying discipline of Organization Modeling has created knowledge that is used in our approach. Practice of using a set of Business- IT relationship frameworks may guide the analyst to design a motivated inquiry to elicit the goals. Henderson-Venkataaraman model [19] and Zachman framework [53] are examples of such frameworks and can provide foundation for business relevant systems development. The frameworks are discussed briefly, as their underlying concepts are used in proposed approach.

Henderson-Venkataaraman's Strategic Alignment Model provides a perspective of strategic choices in business and in technology and how organizing the supporting business and technology infrastructure and process can deliver strategic benefits. The Model views four primary choices (Figure 5), each consisting of 3-tuples as below:

- "Business Strategy" consisting of Business Scope, Distinctive Competencies and Governance

- “Technology Strategy” consisting of Technology Scope, Systemic Competencies and I/T Governance
- “Organizational Infrastructure and Processes” consisting of administrative infrastructure, process and skills
- “Information Systems Infrastructure and Processes” consisting of architectures, processes and skills

Linkages between the strategic elements (Business & IT strategies) and Infrastructure elements determine the strength of strategic fit. The linkages between Business elements (strategy and infrastructure) and Information systems elements determine the strength of functional integration. Strengthening the linkages leads to improvements in organization’s ability to meet its goals. We use this philosophy of linkages in our requirements development approach.

Figure 5: Henderson-Venkataraman Alignment Model

Framework proposed by Sowa and Zachman [39] is a classification of organizational artifacts in systems development when viewed from different perspectives through evolution. It is a matrix of 6X6 views. Transformational views cover evolution of system from contextual view, to conceptual, logical, physical, deployment and operation. Interrogations of What, Why, How, Where, Who and When capture the related artifacts in respective transformational view. The model has an appeal due to comprehensiveness of representation and level wise artifact abstractions are good for interacting with various levels of stakeholders in the organization (Figure 6). Early stage “what” and “why” interrogations are relevant in our thesis. “What” leads to business components in conceptual and logical level of transformation; “Why” provides the rationale exploration for those components.

Does a Job that takes 2 hours and 3 minutes considered to take “about 2 hours” or not? So concept of “about” is not covered by fixed value math where a parameter takes a definite value (including binary value). Such concepts are better represented by “fuzzy” math, where we associate a notion of closeness to the concept. Job J1 taking 1hour 55 minutes and event J2 taking 1 hour 40 minutes could belong to the notion “about 2 hours” distinguished by a membership function that represents how close J1 or J2 is to 2 hours.

Definition: A Fuzzy set $\tilde{A}$ on X set of elements generally denoted by x is a set of ordered pairs of the type $\{x, \mu_{\tilde{A}}(x)\}$, where $\mu_{\tilde{A}}(x)$ called membership function represents the degree of membership to the concept represented by $\tilde{A}$.

In Figure 7, we cite an example where set $\tilde{A}$ is $\{(J1, 0.95), (J2, 0.4)\}$, meaning J1 belongs to “about 2 hours” to a degree of 0.95 and B to a degree of 0.4. In addition, we may have job J3 and J4 with membership function of 0.97 and 0.4 respectively. Membership of J3 will be high in yet another fuzzy set of “very close to 2 hours”. If we defined another fuzzy set “Much longer than 2 hours”, J4 will have a high membership in that. This assignment was dependent on user’s view and may be considered imprecise or arbitrary, but it is still more realistic than alternate crisp definition which does not accept graded values and could term any job not in interval [1Hr 55 Min – 2Hr 05 Min.] as not “about 2 hours”. The membership function formalism provides a method of modeling intuition and vagueness of real life. The concept is extended by fuzzy relations, which provides formal methods using membership functions in modeling relationships and inferences with fuzzy sets. After math operations, aggregated membership functions (the results) can be transformed back to language that intuitively reflects real life understanding such as “very much longer than 2 hours” or “Little over 2 hours”. We cover some relevant definitions below:

Figure 7: Sample Fuzzy Set Membership Function

Definition: Complement of a fuzzy set $A = \{x, \mu_A(x)\}$ is another fuzzy set $\bar{A}$ consisting of elements from the universe that do not belong to A .

$$\bar{A} = \{x, \mu(x)\}, \text{ where } \mu_{\bar{A}}(x) = 1 - \mu_A(x)$$

Definition: Cartesian product: Let A and B be two crisp subsets in X and Y , respectively. The Cartesian product of A and B , denoted by $A \times B$, is defined by

$$A \times B = \{(x, y) | x \in A, y \in B\}$$

Like crisp sets we can do operations on fuzzy sets, which result in fuzzy relations.

Definition: Fuzzy Relation over the pair X, Y is defined as a fuzzy subset R of a Cartesian product $(X \times Y)$ of sets. Membership grade $R\{x, y\}$ estimates the strength of the link between elements x, y .

Union, Intersection, or averaging are some of the operations that result in Fuzzy relations. Fuzzy relations can represent concepts that are composed of individual fuzzy concepts. E.g., a “desirable” system may be one that is “high performance” as well as “functionally compliant”.

Definition: Union of two fuzzy sets $A = \{x, \mu_A(x)\}$, and $B = \{x, \mu_B(x)\}$, is defined as

$$C = \{x, \mu_C(x)\}, \text{ where } \mu_C(x) = \text{Max } \{\mu_A(x), \mu_B(x)\}$$

In Figure 8, shaded area indicates a union of fuzzy sets of “Low Cost” and “Stable” systems, which we may choose to term as “Candidate Systems”. Member ship of Systems S6, S8, and S3 are the max of their membership values in either “Low Cost” or “Stable” systems fuzzy sets. Note S3 has membership value =0 in “Low Cost” but has membership value in “Candidate Systems”. This operation is an OR operation².

Figure 8: Union Operation on Fuzzy Sets

Definition: Intersection of two fuzzy sets $\tilde{A} = \{x, \mu_A(x)\}$, and $B = \{x, \mu_B(x)\}$, is defined as $C = \{x, \mu_c(x)\}$, where $\mu_c(x) = \text{Min} \{\mu_{\tilde{A}}(x), \mu_B(x)\}$

In Figure 9, shaded area represents intersection of fuzzy sets “High Performance” and “Functionally Compliant” systems, let us call it “Desirable” system fuzzy set. Of the systems S1, S6 and S3 only S1 is member of the “Desirable” set. This is an AND operation.

Figure 9: Intersection Operation on Fuzzy Sets

Union (OR) and Intersection (AND) operators are Max and Min operators respectively as we saw from resulting fuzzy set's membership functions. An operator that results in membership between Max and Min is called an Averaging Operator. Real life situations frequently cannot be covered

² “Low Cost” and “Stable” are two different views and should be ideally visualized in a multidimensional projection. We used 2D for ease of drawing.

by mere AND and OR relations. Averaging operator can be used to model concepts that are in between the union and intersection concepts. For example if wanted to be a little accommodative about “Desirable” concept, we may have considered S6 also in the set. The “New Desirable” set would be achieved by an Averaging Operation on fuzzy sets “High Performance” and “Functionally Compliant”. We will use a special class of Averaging Operator in our selection approach covered in Chapter 4.

Bellman and Zadeh’s seminal work [4] demonstrated use of fuzzy logic to model decisions involving imprecise notions. Since then fuzzy techniques have been utilized in modeling and evaluating such notions in a large number of areas such as [14] [3] [22] [55]. This work relates to decision making where a decision space is composed of set of decision alternatives and their values on desired utility function. Specifically in systems development context the decision-making incorporating multiple attributes is covered. The decision maker may have to make a choice based within operative constraints, and may have to pursue some optimization. The techniques have also been used in sorting out common ground amongst diverse viewpoints and “normalize” the subjectivity, with support of additional techniques such as Analytic Hierarchy Process (AHP). We propose to adopt the same techniques in our requirements selection approach discussed in Chapter 4.

3 Approach for Identifying Pertinent Requirements

Previous Chapter discussed GORE approaches that use goals to develop requirements as well as analytical methods that are in practice. Section 1.1.1 highlighted that assuming a given narrative of need necessarily reflects organization goals and true needs may not be correct. One needs to identify stakeholders' influences applicable on the To-Be Processes/System to analyze and develop the pertinent set of requirements. The approach discussed herein leverages GORE concepts while mapping stakeholders' goals and influences to the To-Be System in order to develop requirements.

Need and opportunity for determining scope and pertinent requirements exists at the early stages of the project when the questions of utility and objectives of the project are in focus. This thesis focuses on early stage processes hence we work with requirements that are less granular and relatively abstract. One may view them as conceptual, logical requirements as discussed in Section 2.3. Once we achieve pertinence at this level, already established requirements engineering methods can be used for subsequent analysis, elaboration, validation, negotiation and refinement.

This chapter discusses approach for identifying pertinent requirements. We first discuss the source of pertinence in section 3.1; we then discuss the steps of our method in section 3.2. An application of the approach is discussed via a case study in a consumer packaged goods (CPG) context in section 3.3. Concluding discussion is covered in section 3.4

3.1 Source of Pertinence

The needs exist in a wider context of the organization, (business) environment in which it exists, and the owner stakeholders. Needs are influenced by forces of stakeholder expectations, organizations goals and beliefs, and impact of environment on organization's working. These forces shape organization's components and their configuration to fulfill the goals and beliefs. We depict this in Figure 10.

Figure 10: Exploring Context's Influence on Requirements

In practice volume and intensity of requirements definition process is devoted to various refinement activities of modeling, decomposition, validation and specification and so forth from a given need howsoever crisply or vaguely expressed. Resulting requirements are pertinent in specific context or viewpoint but the need's priority itself may be viewed differently in organization's context. Is it a transient need that will go away? e.g. when a new government regulation is implemented next month?. Did the stated need cover relevant dimensions? Are there related needs that accompany stated need? The questions of scope and pertinence cannot be answered unless we consciously explore the wider organization context during requirements development (Path 2 in Figure 10).

We also argue that if project scope conflicts with or is not aligned to deliver to overall business goals, ongoing support for the project will weaken and the project may be dropped midway.

3.2 Approach for Identifying Pertinent Requirements

Our proposed approach uses GORE methods, which model levels of abstractions of goals, and tasks. Thus, they can map to vocabulary appropriate for stakeholders. The methods' graphical representation provides a view that facilitates communication with various levels of stakeholders. Available GORE methods provide formalism to model relationships between stakeholders' intentions and goals. They also provide formalism for decomposition of goals into tasks that would represent requirements. Goals are viewed as more stable concepts from which to develop requirements [1]. GORE methods also provide formalism for specifications during downstream refinements. Dependency modeling and goal task decomposition concepts from i^* framework [51] discussed in section 2.2.1.2 are used in the approach.

To review the requirements in wider organization context, we identify and validate goals, policies and beliefs that affect the given need. The assessment is done from organization's artifacts or stakeholders that are identified. We also identify any attributes, policies and constraints that apply specifically to information systems in the organization. These could be viewed as top two quadrants of Henderson-Venkatarman model discussed in Section 2.3. Inferring the goal (herein after called Project Goal) behind a given need, or a system concept, we identify stakeholders, their motivations and their goals related to the project. We build the relationship tree with respect to project goals (Figure 11) and then decompose the goals down to high-level tasks akin to

conceptual logical components discussed in Zachman Framework™. These components when elaborated will form the requirements.

Figure 11: Exploring Influences on Project Goal-Tasks

Sequence of Steps in analysis and modeling is following:

- Start with inferred Project Goal behind given need
- Identify project Stakeholders using i^* SD modeling concept
- Identify organization goals, beliefs and policies and resulting Stakeholders' Goals
- Establish relationship of related Stakeholders' Goals to Project Goal
- Build Project Goal tree incorporating Stakeholders' related Goals
- Refine/ Decompose/ Abstract Goals using i^* SR modeling concept
- Establish Tasks to Goal Mapping using means-end and decomposition techniques

Validation steps are required once the goal tree is established and when decompositions and mapping is completed. By forcing review of relationship of Project Goals with organization's goals, we achieve pertinence. We illustrate the method by following Case Study.

3.3 Case Study I

Situation and expressed Need

The present case study concerns a situation where a systems team has been mandated to develop a *“tracking system for direct promotions of a consumer packaged goods company”*³. Aim is to make effective promotions. What is effective promotion is not explicitly defined and has to be analyzed. Analyst has to identify pertinent system requirements. The project is sponsored by the Category Manager, who is responsible for success of a range of similar products. *It is worth noting that real life business narratives can be vague to start with, as mentioned above and process of analysis and modeling is required to tease out the components of “effectiveness”, “success” to derive the requirements.*

Analysis and Modeling Process

There could be multiple starting points from various perspectives e.g. top down from a Strategist’s viewpoint or from a Brand Manager’s viewpoint. For practical reasons of accessibility, this instance starts building analysis with a Brand Manager and then seeks validations of other perspectives. Physical process or To-Be processes provide a tangible path on which rationale and goal relationship is explored to find pertinent requirements. By dialoging and understanding how a promotion works, other initial actors are identified as below (Table 3) and the initial actor-goal relationship as in Figure 12.

---

³ Consumer Packaged Goods (e.g. cereals, shampoos) manufacturers conduct ongoing promotions about their products. This is done to maintain interest in their products as consumers may otherwise become indifferent, protect market share when competition induces loyal customers to switch and try their brands and to gain market share by winning new customers who may start liking the brand once they have been lured to try them out because of promotion. Direct promotions with consumers can take many forms including distribution of free sample, bundling of a sample with another product, special pricing sale, trial packs with attraction of good pricing or gift and so forth.

Figure 12: Case Study I Actor Dependencies

Table 3: Case Study I - Actors and Roles

Category ⁴ Manager	Responsible for sales of a group of similar products
Executive	Wants effective promotions for company's success
Market Analyst	Tracks and Analyzes market data
Brand ⁵ Manager	Responsible for a brand value
Logistics Manager	Ensures stock availability at stores
Territory Manager	Manages relationship with Stores; prepares Stores for promotion
Advertising Manager	Develops messages and engages channel (e.g. TV) for message delivery
Stores	Sells to consumer and hosts in-store promotion activity
Consumer	End buyer (household rep) of company's products and subject of promotion

⁴ A Category is a group of similar products (e.g. cereals) which may be sold in multiple packaging (or SKU- stock keeping units). There may be many brands within a category e.g. "All Bran", "Vector", "Special K" in breakfast cereals category.

⁵ Brand is a distinctive recognition of a product by name and logo and signifies specific values that the buyers associate with this distinctive product e.g. "pure", "all round nutrition", "economical" etc.. Consumers are willing to pay a certain "(premium) price" and have a "preference (loyalty)" for the value of they associate with the brand.

Actors' motivations and intentions influence their goals related to Promotions. Actor's motivations are themselves influenced by company's beliefs and goals. In this instance beliefs and goals include Sustained Growth (in market share and profitability), Growth of core product category, Expansion of new Products business, Serving consumer's changing needs, Operational efficiency, strong brands and selling healthy products.

Analyst builds the following (Table 4) mapping by analysis of Stakeholder motivations.

Table 4; Case Study I - Stakeholder Motivations and Resulting Goals

<i>Stakeholder</i>	<i>Motivations</i>	<i>Related Goal</i>
Brand Manager	Protect Brand Value	Track Brand Value
Category Manager	Grow Category Volume, Grow New Products	Track Volumes (core and new products), Track Growth
Territory Manager	Timely Promotion Materials, Grow Territory Volumes	Merchandising effectiveness
Logistics Manager	Efficient Distribution	Track In-Store Availability
Advt. Manager	Effective Messaging	Track Message Effectiveness
Market Analyst	Promotion Analysis	Analyze Promotion
Executive	Market Position	Review Promotion Performance
Store	Store sales growth, Store Efficiency	Easy Data Reporting
Consumer	Good Products and Price	

This understanding of the related goals and how they impact the Goal of effective promotions allows us to construct goal tree (Figure 13).

Figure 13: Case Study I- Goal Tree

In the next stage individual goals are assessed for their composition. This is an analytical process, which can be supported by respective stakeholder/domain expert. Decomposition to the next level provides breakdown as shown in Table 5.

Table 5: Case Study I- Goal Decomposition

<i>Related Goal</i>	<i>Subgoals</i>
Track Brand Value	Track Loyalty (AND) Track Brand Premium (AND) Track Market Parameters
Track Volumes (core and new products) Track Growth	Track Product Volume (AND) Track Market Share
Merchandising effectiveness	Track Consumer Eyeballs
Track In-Store Availability	Track In store inventory (AND)Track (Demand Information OR Stock out)
Track Message Effectiveness	Track promo awareness, Track Traffic
Analyze Promotion	
Review Promotion Performance	Get Timely Drilldown on KPIs
Easy Data Reporting	

Figure 14 shows an updated goal tree with updates for two of the goals, namely Track Brand Value and Track Volume.

Figure 14: Case Study I- Updated Goal Tree

Decomposition of next level of goals (e.g. TrackBrandValue) is the next step. For illustration purposes we decomposed two goals TrackBrandValue and Track_Volume as shown in Figure 15.

Figure 15: Case Study I Goal Task Decomposition

This decomposition leads us to identification of series of tasks which are now ready for operationalization into requirements. A goal/subgoal may be accomplished in different ways i.e. by different set of tasks. These, then will form alternatives for consideration. We use components at this level to make early stage assessments of which alternative to choose for deeper decomposition and actual requirements development. Implementation may use common services or components. In above instance Track_SKU_Volume is conceived in both TrackBrandPremium and Track_Volume goals and cost assessment calculations during evaluation would not double count the costs.

3.4 Discussion

GORE methods offer a modeling and formalism for specification of requirements and their relationship with goals. They do not suggest any specific approach for exploring the context while developing requirements. We discussed an approach and walked through the steps in case study. The approach considers need for Goal discovery as suggested by GBRAM. The approach uses Strategic Dependency and Strategic Rationale concepts forwarded by *i** model. The result of the approach is usable for downstream requirement elaboration and specification by traditional approaches, scenario based methods as well as by lifecycle spanning GORE tools when they are available. Some may be in various stages of development⁶.

Using the approach, we were able to articulate stakeholder motivations by relating to organizations beliefs and goals. By forcing consideration of other stakeholders' motivations, we arrived at stakeholder goals and needs that influence and more comprehensively define the business need. Via goal-task decomposition, we arrived at requirements set that now represents relevant viewpoints (stakeholder goals). Stakeholder related goal is more tangible and generally verifiable artifact, as that is normally how the stakeholders review their performance. The approach gave relevant vocabulary for early stage stakeholder dialog elicitation and concept development dialog. Arguably, we are working with goal concepts that are vague, and potentially interpreted differently by different stakeholders. We believe introducing formalism and strict goal

---

⁶In particular, a Tropos specific tool was sourced during 2008 research but could not be evaluated due to lack of documentation and non availability of support.

definition will be at the cost of increased complexity, and reduced collaboration due to vocabulary unfamiliar to stakeholders.

In absence of adequate scenario coverage and limited time to cover wide span of stakeholders via traditional interviewing methods, one would have conducted the elicitation dialog with selected stakeholders. This would drive task decomposition from a viewpoint limited by the stakeholder who gave requirements and discussed influences that he/she could think of. Validation of such requirements can be extremely time consuming as stakeholders motivations may be at variance with what was narrated. Also in traditional methods, the communication from analyst would have used vocabulary of objects, classes, etc. something not so native to the business stakeholders.

While there is a model (and graphical) support for early elaboration, the process is analytical. This also implies that analysis by different individuals will result in different decomposition. We consider it valuable, as the method documents the rationale behind the decomposition and the fact that real life environment does not have a rigid structure that only a single solution will evolve. Goals happen to be more stable [1] than processes and provide a better foundation for requirements. The artifacts generated can be updated by consideration of additional goals or review of goals upon temporal and contextual changes, and will not be wasted. The method provides framework for building a quantitative evaluation and reasoning of alternatives, which we discuss in next Chapter. Organizations may be practicing specific methodologies due to organizational skills, learning and experiences. Tropos [15] methodology covered in literature from transformational perspective [9], [10], could also be used to arrive at the result via its bottom up reasoning.

4 Requirements Selection

Configurations of the requirements (derived from related business component) that we explored in previous chapter define our alternatives. Need is to select the most appropriate alternative for further refinement. Appropriateness depends on strength of an alternative on some decision criteria. Those criteria would be context dependent and could be along dimensions of strategic fit i.e. the project that we wanted to achieve, cost, feasibility and so forth. The context includes internal and market factors prevailing at the time. This is a Multi Criteria Decision Making (MCDM) problem. We applied decision-making concepts and fuzzy set concepts to develop proposed selection method. We use a generalized class of fuzzy aggregation operations that allow us to model overall goal satisfaction under multiple criteria. This enables us to select the best fitting requirements set. Approaches using fuzzy aggregations have been successfully used in such decisions [49], [55].

This Chapter first outlines proposed selection method steps (section 4.1). We then cover some theoretical background related to decision-making and the fuzzy math operators that we use in our method in Section 4.2. We then discuss creation of decision model in Section 4.3 . We demonstrate the method application via a case study in Section 4.4; and follow it with a discussion on the method and its relevance with respect to alternate methods in Section 4.5.

4.1 Outline of Proposed Method

Multiple configurations of business components could address the business need. The identified configurations are evaluated for selection. We select an alternative based on assessed potential satisfaction of goals by the alternative. Figure 16 indicates the selection methodology.

Figure 16: Methodology for Selection

The steps in our method include:

- Analyze goals, decision parameters and structure criteria set (in hierarchy if relevant)
- Grasp decision attitude and linguistic decision preferences
- Create decision model based with relevant OWA aggregation parameters and linguistic quantifiers on attitude and preferences
- We evaluate each alternative using the decision model and select what fits our decision goals
- Outcome is iterated with stakeholders for decision

We illustrate the steps via the case study of a provincial healthcare initiative (Section 4.4), before which we discuss construction of decision criteria in Section 4.3.

4.2 Theoretical Background for Proposed Method

4.2.1 Decision Making and Nature of Evaluation Criteria

Inherent to selection decision in systems development context are some complexities that may exist in varying degrees in a particular context. These may change with time due to changes in internal or external environment or the organization's goals and policies.

- Multiple attributes: diversity of dimensions (e.g. feasibility and cost) on which alternatives need to be assessed.
- Multiple Viewpoints and Objectives: Alternatives evaluated from varied perspectives demand different strengths on different attributes.

- Assessment subjectivity: A given alternative may be scored differently by individuals e.g. risky for some individual and may be low risk for another
- Scale variability: Assessments on different attributes may be of different types. On some attributes one may be able to only establish comparative merit while on others one may be able to compare on some ordinal, cardinal scale.

We discuss handling of these complexities in relevant places within the Chapter.

Often there is difficulty of making precise assessment of an alternative along some attribute and assessment may be expressed in linguistic terms e.g., “Alternative A is low cost”. In addition, we observe that some of the criteria may be composed of sub-concepts themselves not well defined. “Organizational Readiness” may consist of “state of motivation” of the organization and/or its’ partners to adopt proposed change, “availability of resources to implement the project”, “availability of adequate time window within the series of other projects scheduled” and so forth. We generalize the types of assessments in Table 6, with some examples of criteria used during systems development.

Table 6: Types of Attribute Assessments

Type of Attribute assessment	Examples
Objective - Fixed Value	Digital Certificate –Yes or No Per Transaction Cost- X cents/transaction
Objective - Fuzzy	Estimated Maintenance Cost should be Low
Subjective - Fixed Value	Is it achievable in 2 months?- Yes or No
Subjective - Fuzzy	Should have high appeal for business partners Should be easy to implement

Taxonomy of attributes considered for evaluating alternatives may include:

- Benefit Oriented: e.g. Strategic fit i.e. satisfaction on project objectives
- Feasibility or Constraints Oriented: e.g. Cost, Schedule, Implementation feasibility, Risk
- Non Functional: e.g. System performance, reliability
- Architectural: Technical architecture, Platform longevity, System granularity

As introduced in Chapter 2, fuzzy set theory has been utilized in similar decision-making. We use fuzzy set and aggregations to make sense under such complexities. Should it be possible to

define and compute an aggregate strength, we could use the resulting aggregate values for selection. Subject to feasibility or reassessment of the criteria, one may iterate the steps with modified set of criteria or their importance. In our approach, we model the constraints in similar fashion as the criteria.

4.2.2 Ordered Weighted Averaging Operator

Section 2.4 discussed AND / OR operators on fuzzy sets. AND operation on two fuzzy sets would lead to resulting fuzzy set with membership function as minimum of individual sets, whereas OR operation would result in a fuzzy set with maximum of individual sets. While AND is conjunctive, OR is disjunctive; anything in between would be an averaging operator. Averaging operator compensates between the membership functions.

Example 4.1: We assume alternative systems (x, y, z) are to be evaluated for selection in a company. Attributes of “Low Cost” (L), “Ease of Implementation” (M) and “Richness of Functionality” (F) are to be considered.

Let us consider L, M and N as fuzzy sets. Let us also assume we have assessment from experts on how much each alternative satisfies the concept L, M and N as shown below.

$L = \{(x, 0.4), (y, 0.5), (z, 0.5)\}$ where x has 0.4 satisfaction on Low Cost

$M = \{(x, 0.2), (y, 0.7), (z, 0.4)\}$ where x has 0.2 satisfaction on Ease of Implementation

$F = \{(x, 0.8), (y, 0.4), (z, 0.7)\}$ where x has 0.8 satisfaction on Richness of Functionality

AND operator will result in a new fuzzy set A

$A = \{(x, 0.2), (y, 0.4), (z, 0.4)\}$

If set “A” represented some aggregation desirable property, goodness of alternative x on other attributes would have got negated by poor score on M, and x would be the poorest alternative.

Whereas OR would lead to a new fuzzy set O

$O = \{(x, 0.8), (y, 0.7), (z, 0.7)\}$; x would be the best alternative if O had a desirable meaning.

One may do a Mean operation (averaging) which would result in fuzzy set H

$$H = \{(x, 0.47), (y, 0.5), (z, 0.53)\}.$$

The set H has compensatory membership values x , y , and z alternatives.

If the attributes had weights (of importance), a weighted averaging would compensate by respective importance ratios. Generalized approaches using Weighted Averaging Operators on fuzzy sets [38] have been used for problems involving criteria with relative importance. Such averaging operators are not able to represent situations where we would want to value an alternative's exceptional desirability on categories of attributes. They also fall short when desirability of an alternative is a synergistic function of its strength on various attributes. For example, if alternative system configuration $S1$ fares well on functional compliance of "customer service", "analysis granularity" and "data synchronization" needs, we will be willing to pay a much higher price.

Yager [46] introduced Ordered Weighted Averaging Operator (OWA), a parameterized operator that introduces order weights (not the same as criteria/attribute importance) thus providing the ability to vary the degree of AND or ORness in averaging. This can represent situations where we want to value "all", "most", "some", "half", "at least one" criteria in our decision making. In extreme case of "all", the OWA operates as AND whereas on other extreme of "at least one" it operates as OR. By varying the parameter, one can vary the degree of AND/OR ness. OWA operators can model human decisions more closely than pure weighted averaging operators can.

Yager [46] defined OWA operator as below:

Definition: An OWA operator F of dimension n is a mapping

$$F: R^n \rightarrow R$$

which has an associated weighting vector W

$$W = \begin{bmatrix} w_1 \\ w_2 \\ - \\ - \\ w_n \end{bmatrix}$$

such that

$$w_j \in [0,1]$$

$$\sum_{j=1}^n w_j = 1$$

$$F(a_1, a_2, \dots, a_n) = \sum_{j=1}^n w_j b_j$$

b_j , being the j th largest of the collection a_i .

e.g. if a System alternative a had membership of $a_1, a_2, \dots, a_n$ in fuzzy sets $A_1, A_2, \dots, A_n$, then by applying the OWA operator we will get a result that is (additive) aggregation of a unique kind depending on the weight vector W .

By changing weights, we can change the nature of aggregation. When the weights are concentrated towards the top of weight vector the affect of aggregation is more OR like, which promotes a tolerant decision-making not so much penalizing the alternative's low scores on attributes, but rewarding high scores.

When the weights are concentrated more towards the bottom of weight vector (extreme case $[0, 0, \dots, 1]$), decision is stricter; weighing higher the poorer qualities (attributes where alternative scores poorly) of an alternative.

Example 4.2:

We take the same fuzzy sets as in example 4.1

$L = \{(x, 0.4), (y, 0.5), (z, 0.5)\}$ where alternative x has 0.4, y 0.5 and z 0.5 satisfaction on Low Cost

$M = \{(x, 0.2), (y, 0.7), (z, 0.4)\}$ where alternative x has 0.2 satisfaction on Ease of Implementation

$F = \{(x, 0.8), (y, 0.4), (z, 0.7)\}$ where alternative x has 0.8 satisfaction on Richness of Functionality

We will do aggregation with Weight vector $\begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}$

Let us take alternative x. The new ordering (b_j) is $[0.8, 0.4, 0.2]$ where highest membership (0.8) gets first position.

$$\text{Aggregation for alternative } x = [0.8, 0.4, 0.2] \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} = (0.8)(1) + (0.4)(0) + (0.2)(0) = 0.8$$

Which is the OR aggregation.

$$\text{Now we aggregate with Weight vector } \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$$

Aggregation for alternative x now is =

$$[0.8, 0.4, 0.2] \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} = (0.8)(0) + (0.4)(0) + (0.2)(1) = 0.2 \text{ Which is the AND aggregation.}$$

$$\text{If we take the Weight vector as } \begin{bmatrix} 1/3 \\ 1/3 \\ 1/3 \end{bmatrix} \text{ the aggregation on } x \text{ looks as below}$$

Aggregation for alternative x =

$$[0.8, 0.4, 0.2] \begin{bmatrix} 1/3 \\ 1/3 \\ 1/3 \end{bmatrix} = (0.8)(1/3) + (0.4)(1/3) + (0.2)(1/3) = 0.47 \text{ Which is Mean}$$

operation in previous example.

Yager showed the OR ness of the OWA as

$$\text{ORness } (w) = \frac{1}{n-1} \sum_{i=1}^n (n-1)w_i \quad \text{----- (1)}$$

We will use this AND ness and OR ness capability in conjunction with the concept of Linguistic Quantifiers that represent human decision making.

4.2.3 Linguistic Quantifiers

Linguistic Quantifiers represent common language words e.g. *most*, *almost all*, *some* etc. that quantify a concept and are common in natural thinking and decision making. An alternative may be chosen only if *most* stakeholders rate it “highly”. An alternative may be listed for consideration if it fares well on *much more than half* of the attributes.

Zadeh [54] viewed Quantifiers as linguistic probability, which determines the degree that the concept Q has been satisfied by r.

Yager provided following method for working with Quantifiers using OWA.

If the decision agenda requires *Q criteria to be satisfied by an acceptable alternative*, where Q is a RIM (Regular Increasing Monotone) quantifier e.g. *most*, *all*, then we can find weight vector of an OWA that represents the Q. The procedure is as follows:

For OWA of n dimension,

$$\forall \quad i = 1, \dots, n \quad w_i = Q\left(\frac{i}{n}\right) - Q\left(\frac{i-1}{n}\right) \quad \text{-----}(2)$$

Q may take the form $Q(r) = r^\alpha$, where r exists in unit interval and α a number that varies with the linguistic concept. Concept “most” it is represented by $Q(r) = r^2$.

$$\text{ORness } Q_\alpha = \int_0^1 r^\alpha dr \quad \text{-----}(3)$$

α value of 1 gives a mean operation while $\alpha < 1$ gives more OR like accommodative “some” etc.

4.2.4 Aggregation with Criteria Importance

In some decision situations we may have attributes that are more valued. Our decision agenda requires us to value goodness on important attributes differently than that by weighted averaging. Yager [48] demonstrated aggregation with criteria importance using OWA operators as below.

Let $\{A_1, \dots, A_n\}$ be the criteria (attribute satisfaction membership) and V_i be the importance of ith criterion, $x \in X$ be set of alternatives.

We reorder $A_i(x)$ such that b_j is the j-th largest of $\{A_1, \dots, A_n\}$.

Let u_i be the importance associated with the criterion that has j-th largest membership.

We can create a OWA of dimension n that to get an aggregation with criteria importance. The n dimensional OWA vector is defined by

$$w_j(x) = Q\left[\frac{\sum_{k=1}^j u_k}{T}\right] - Q\left[\frac{\sum_{k=1}^{j-1} u_k}{T}\right]; \text{ where } T = \sum_{k=1}^n u_k \quad \text{-----}(4)$$

And our aggregation with incorporating relative importance is $D(x)$

$$D(x) = \sum_{j=1}^n b_j w_j(x) \quad \text{-----}(5)$$

This is used in the Case Study II covered in Section 4.4.

4.3 Establishing Decision Criteria

This section discusses development of decision criteria. For making early stage decisions about fitness of alternative we need to arrive at criteria that can be acted upon for analysis of alternatives. We identify the goals (also called attributes), whose satisfaction is required by alternatives.

Following considerations may apply while building the evaluation criteria.

- Relative importance amongst the attributes and/or category of attributes
- Structure of attribute set
- Compensation or otherwise of weakness of an attribute with strength in other attributes
- Cutoff value of attribute(s)

Attributes may need to be considered a unique way with respect to their relationship, precedence, significance (weight) and complementarities. For instance for a system that envisions a long term organization and partner wide methods improvement, strategic fit and “ease of change deployment” may have more importance (weight) than “time schedule”. Decisions may be based on criteria structure such as whether all attributes are to be considered or assessment on some or most (e.g. 6 out of 8 or best 5 of 8) attributes can be considered. One may find it appropriate or

otherwise to balance (referred as Compensation) weakness of an alternative on one attribute with strength(s) on other attribute(s). Alternatives may qualify for comparison only upon achieving some cutoff. For a system to process electronic payments may have “technical robustness” very high on criteria importance list (precedence) and only if an alternative provides a certain level of robustness would a decision maker qualify it for consideration.

Decision can be based on respective ranking of alternatives or on overall strength of alternatives on some function that approximates or represents the benefit of the alternative. The notion of strength is important as in many situations best of poor alternatives (arrived by ranking) does not lead to good decision. We will discuss strength score oriented decision. The discussion also precludes situations where decision makers are not capable of assessing an alternative on an attribute or an alternative’s predicted behavior is uncertain.

Criteria emanate from the stakeholders’ perspective of the context, their intuitions and knowledge e.g. a stakeholder wanting to catch a market opportunity window, may view “deployment timelines” as a criterion, someone managing finances may value “cost” as a criterion or someone tasked with implementation of the system may view “organizational readiness” as a criterion. By their experience of such decision-making, organizations have intrinsic knowledge and may have some established criteria for similar decisions. Guiding taxonomies may provide a template that could be explored for developing criteria for a specific decision instance.

When based on single individual’s choice or hard policy it is straightforward and criteria assigned by the individual may be the starting point. A review with respect to guiding taxonomies and organizational goals would refine the criteria. If policies or criteria template are non-existent or found to be not applicable, one could establish the criteria by sourcing expert opinion, or group techniques. For a group based criteria elicitation and validation techniques such as brainstorming, Nominal Group Techniques, Delphi method [42] are available. The techniques have also been used in sorting out common ground amongst diverse viewpoints and “normalize” the subjectivity. Approaches for establishing relative importance of criteria include use of Analytic Hierarchy Process (AHP). A discussion of this nature is available in [34].

4.4 Case Study II

Background

We consider example of selection of potential software solution alternatives that have been conceptualized for a provincial health organization. The organization is chartered by government to facilitate on improvement of medical services in diagnosis and care of a certain disease. The organization works with regional centers, medical practitioners and participates in improving awareness about disease detection, control and treatment options with all stakeholders including private citizens. The organization has adopted key goals and principles based on Government expectations, experiences and knowledge of decades as below:

1. Improve disease screening
2. Streamline diagnosis
3. Adhere to Privacy regulations
4. Improve Client experience
5. Provide equitable coverage across regions
6. Adopt methods to utilize medical and technological advances
7. Run effective programs: Optimal return on taxpayer dollars

Building on its' experiences and knowledge of decades, the agency has identified a strategic program to meet stakeholder expectations. Using the methods similar in Chapter 3, the agency has arrived at some alternative set of system components to enable the strategic program of Preventing Cervical Cancer. *Now the agency wants to select the best alternative.*

Decision Modeling and Analysis

Based on its knowledge and experience the agency has evolved decision guidelines that value both "strategic importance" of the program as well as the agency and their partners "ability to execute" the proposed program. This year decision makers want to be stricter in their evaluation of "ability to execute" as province has several other programs that are dependent on proposed program.

The alternatives are assessed for their compatibility to decision attributes by two categories of individuals that includes subject matter experts and executive stakeholders. Importance assigned to individual category's opinions is decision attributes dependent e.g. solution longevity opinion is more valued from experts as they have a better judgment on discipline progress and applicable technologies.

Following methodology proposed in Figure 16, decision-modeling steps are as follows:

First, we establish the decision criteria. This includes:

Decision agenda is decomposed to goals and attributes on which the alternatives need to be assessed.

This decomposition includes any hierarchical decomposition if necessary. Clearly it must include evaluation of alternatives on organizations business goals. Let us assume the current initiative is neutral to equitable regional coverage, and medical technology advance adoption. We must evaluate compliance to remaining five of the seven goals identified by the organization as mentioned earlier in the case study. Let us initially classify them as *strategic fitness* category of goals. Using organization experience and dialog with stakeholders we determine the components of "ability to execute". They represent feasibility, risk, cost constraints type of factors and we categorize them in *execution fitness* category. With stakeholders' consent, we move cost related effective use of tax payer dollars goal to *execution fitness*. Now our criteria set is representative of business goals as well as operational implementation goals. With GORE based approach as in chapter 3, we would have decomposed the functional goals already (Figure 17). Similarly, we decompose the implementation goals (Figure 18). The result represents all goals on which we must evaluate the fitness of alternatives, which we arrange in logical hierarchy.

Figure 17: Case Study II Indicative Goal Tree (functional goals)

Figure 18: Case Study II- Indicative Implementation Goals

We present this in Figure 19.

Figure 19: Case Study II- Evaluation Attributes Framework

We build the decision structure based on the decision attitude.

This decision structure includes strictness or risk acceptance in decisions as well as relative importance of attributes, attribute groups. This is given in Table 7, which is arranged in consistence with the evaluation framework. First column corresponds to the root level decision and the last column corresponds to leaf level decision.

Table 7: Case Study II- Decision Structure and Attribute Importance

Higher Importance (0.6) to Execution Fitness Strategic Fitness importance=0.4	Only alternatives meeting privacy adherence norms will be considered. No additional points for alternatives that exceed the norms.	The alternatives must satisfy few of the components of Disease Screening Support Appointment Ease has 50% importance in the category
	All other strategic fitness components have equal importance.	Alternatives must satisfy most of the Wait Time Reduction attributes
	Alternatives must satisfy most of the attributes.	Scheduling support has 50% of importance of the category
		Neutral about respective satisfaction of individual Client experience components
	Strict evaluation of Execution fitness; All of the component attributes must be satisfied; Components have equal importance	Neutral about respective satisfaction of Spend fitness components

Next we find the parameters in OWA aggregation that will represent above decision attitude. This is shown in Table 8

Table 8: Case Study II - Aggregation Parameters

Higher Importance (0.6) to Execution Fitness Strategic Fitness importance=0.4	All other strategic fitness components have equal importance. Satisfy most of the attributes. OWA with $\alpha = 2$	Satisfy <i>few</i> of the components of Disease Screening Support OWA $\alpha = 0.7$
		Satisfy <i>most</i> of the Wait Time Reduction attributes OWA $\alpha = 2$
		Neutral about respective satisfaction of individual Client experience components OWA with Mean
	Really <i>high</i> satisfaction of component attributes. OWA with $\alpha = 4$	Neutral about respective satisfaction of Spend fitness components OWA with Mean

We will now aggregate the goal satisfaction along the decision structure from bottom up. The general aggregation formulation is:

$$D_{lC_n}(S_x) = \sum_j^{k_{(l-1)}} b_j w_j \text{-----}(6)$$

Where

$D_{lC_n}(S_x)$ is the aggregation for Criterion C_n at level l for alternative S_x

$k_{(l-1)}$ is the number of child criteria (i.e. at level $l-1$) to the parent criterion C_n at level l .

b_j is in rank order of criteria satisfaction, b_j being the j th largest satisfaction.

Alternative S_x with largest value of goal satisfaction ($D_{lC_n}(S_x)$ value) at the highest level of the decision hierarchy is considered the best candidate alternative.

Now the decision calculations are as below:

In this Case study, we will have to perform 3 levels of aggregation. Satisfaction on leaf level criteria of “target coverage”, “appointment ease” and so on are aggregated to level 1 parent criterion of “Disease Screening Support”. Level 1 criteria of “Disease Screening Support”, “Wait Time Reduction” and “Client Experience” will be aggregated to level 2 criterion of “Strategic Fitness”. Level 2 satisfactions will then be aggregated above at the program level which in our case will be level 3.

We assume 3 candidate alternatives namely S_1 , S_2 and S_3 .

Alternatives are evaluated at each of the leaf goals and aggregated upwards. Let us assume Subject Matter Experts made assessments on potential satisfaction on leaf level goals by each individual alternative. The assessments are normalized to arrive at a common unit value scale. Normalized attribute assessment of alternatives is as shown in Table 9. In this table bottom three rows represent individual alternative's (S_1 , S_2 and S_3) satisfaction on leaf goals (let us call it Level 0), which are aggregated to higher level 1 (e.g. Disease Screening, Wait Time), and then on to next higher Level 2 (e.g. Strategic Fitness).

Table 9: Case Study II- Alternatives' Goal Satisfaction Assessment

	Strategic Fitness										Execution Fitness			
	Disease Screening Support				Wait Time Reduction			Client Experience Support			Implementation Feasibility	How Risk Proof	Spend Fitness Importance of Components (1:1)	
	Importance of Components (1:3:1:1)				Importance of Components (2:1:1)			Importance of Components (1:1:1)						
Alternative	Target Coverage	Appt Ease	Supply Chain Support	Measurement & Monitoring	Scheduling Support	Referral & Stage Wise Management	Measurement & Monitoring	Citizens experience	Practitioner experience	Measurement & Monitoring			Investment Longevity	Cost
S_1	.6	.5	.9	.8	.5	.9	.6	.6	.7	.7	.6	.8	.9	.5
S_2	.7	.7	.8	.2	.9	.4	.6	.7	.8	.3	.8	.5	.7	.9
S_3	.6	.4	.8	.9	.7	.6	.5	.6	.6	.9	.8	.7	.5	.8

Level 1 Aggregations

From Equation (6) we have

$$D_{l_{C_n}}(S_x) = \sum_j^{k_{(l-1)}} b_j w_j$$

Here C_n are 1. *Disease Screening (ds) Support*, 2. *Wait Time (wt) Reduction* and 3. *Client Experience (ce) Support*. We will aggregate individually on these criteria. Criterion Privacy Adherence was a Go/No-Go criterion and above mentioned alternatives are assumed to have qualified on Privacy Adherence.

We use equation (2) for determining aggregation weights, α parameter from (3) for ORness value and (4) for assigning relative importance of criteria while performing aggregations.

Disease Screening (ds) Support Aggregation

Alternative S1

$$b_1 = 0.9 \quad u_1 = 1$$

$$b_2 = 0.8 \quad u_2 = 1$$

$$b_3 = 0.6 \quad u_3 = 1$$

$$b_4 = 0.5 \quad u_4 = 3$$

From equation (4), we have $T = \sum_{k=1}^4 u_k = 6$. Determined α value is 0.7.

$$w_1(S1) = Q\left(\frac{1}{6}\right) - Q\left(\frac{0}{6}\right) = \left(\frac{1}{6}\right)^{0.7} - \left(\frac{0}{6}\right)^{0.7} = 0.29, \dots \text{as.chosen.} \alpha = 0.7$$

$$w_2(S1) = Q\left(\frac{2}{6}\right) - Q\left(\frac{1}{6}\right) = 0.18$$

$$w_3(S1) = Q\left(\frac{3}{6}\right) - Q\left(\frac{2}{6}\right) = 0.15$$

$$w_4(S1) = Q(1) - Q\left(\frac{3}{6}\right) = 1 - \left(\frac{1}{6}\right)^{0.7} = 0.39$$

$$D_{1ds}(S1) = \sum_{j=1}^4 b_j w_j(S1) = 0.9(0.29) + 0.8(0.18) + 0.6(0.15) + 0.5(0.39) = 0.685$$

Alternative S2

$$b_1 = 0.8 \quad u_1 = 1$$

$$b_2 = 0.7 \quad u_2 = 3$$

$$b_3 = 0.7 \quad u_3 = 1$$

$$b_4 = 0.2 \quad u_4 = 1$$

$$w_1(S2) = Q\left(\frac{1}{6}\right) - Q\left(\frac{0}{6}\right) = \left(\frac{1}{6}\right)^{0.7} - \left(\frac{0}{6}\right)^{0.7} = 0.28$$

$$w_2(S2) = Q\left(\frac{4}{6}\right) - Q\left(\frac{1}{6}\right) = 0.47$$

$$w_3(S2) = Q\left(\frac{5}{6}\right) - Q\left(\frac{4}{6}\right) = 0.13$$

$$w_4(S2) = Q(1) - Q\left(\frac{5}{6}\right) = 1 - \left(\frac{1}{6}\right)^{0.7} = 0.12$$

$$D_{1ds}(S2) = \sum_{j=1}^4 b_j w_j(S1) = 0.8(0.28) + 0.7(0.47) + 0.7(0.13) + 0.2(0.12) = 0.69$$

Alternative S3

$$b_1 = 0.9 \quad u_1 = 1$$

$$b_2 = 0.8 \quad u_2 = 1$$

$$b_3 = 0.6 \quad u_3 = 1$$

$$b_4 = 0.4 \quad u_4 = 3$$

$$w_1(S3) = Q\left(\frac{1}{6}\right) - Q\left(\frac{0}{6}\right) = \left(\frac{1}{6}\right)^{0.7} - \left(\frac{0}{6}\right)^{0.7} = 0.29$$

$$w_2(S3) = Q\left(\frac{2}{6}\right) - Q\left(\frac{1}{6}\right) = 0.18$$

$$w_3(S3) = Q\left(\frac{3}{6}\right) - Q\left(\frac{2}{6}\right) = 0.15$$

$$w_4(S3) = Q(1) - Q\left(\frac{3}{6}\right) = 1 - \left(\frac{1}{6}\right)^{0.7} = 0.38$$

$$D_{lds}(S3) = \sum_{j=1}^4 b_j w_j(S1) = 0.9(0.29) + 0.8(0.18) + 0.6(0.15) + 0.4(0.38) = 0.647$$

Wait Time (wt) Reduction Aggregation

Alternative S1

$$b_1 = 0.9 \quad u_1 = 1$$

$$b_2 = 0.6 \quad u_2 = 1$$

$$b_3 = 0.5 \quad u_3 = 2$$

$$T = \sum_{k=1}^3 u_k = 4$$

$$w_1(S1) = Q\left(\frac{1}{4}\right) - Q\left(\frac{0}{4}\right) = \left(\frac{1}{4}\right)^2 - \left(\frac{0}{4}\right)^2 = \frac{1}{16} \dots \text{as } \alpha = 2, \text{ for "Most"}$$

$$w_2(S1) = Q\left(\frac{2}{4}\right) - Q\left(\frac{1}{4}\right) = \left(\frac{1}{4}\right) - \left(\frac{1}{16}\right) = \left(\frac{3}{16}\right)$$

$$w_3(S1) = Q(1) - Q\left(\frac{2}{4}\right) = 1 - \left(\frac{1}{4}\right) = \frac{3}{4} \dots$$

$$D_{lwt}(S1) = \sum_{j=1}^3 b_j w_j(S1) = 0.9\left(\frac{1}{16}\right) + 0.6\left(\frac{3}{16}\right) + 0.5\left(\frac{3}{4}\right) = \frac{8.7}{16} = 0.544$$

Alternative S2

$$b_1 = 0.9 \quad u_1 = 2$$

$$b_2 = 0.6 \quad u_2 = 1$$

$$b_3 = 0.4 \quad u_3 = 1$$

$$T = \sum_{k=1}^3 u_k = 4$$

$$w_1(S2) = Q\left(\frac{2}{4}\right) - Q\left(\frac{0}{4}\right) = \frac{1}{4}$$

$$w_2(S2) = Q\left(\frac{3}{4}\right) - Q\left(\frac{2}{4}\right) = \frac{5}{16}$$

$$w_3(S2) = Q(1) - Q\left(\frac{3}{4}\right) = 1 - \left(\frac{3}{4}\right)^2 = \frac{7}{16}$$

$$D_{1wt}(S2) = \sum_{j=1}^3 b_j w_j(S2) = 0.9\left(\frac{1}{4}\right) + 0.6\left(\frac{5}{16}\right) + 0.4\left(\frac{7}{16}\right) = \frac{9.4}{16} = 0.587$$

Alternative S3

$$b_1 = 0.8 \quad u_1 = 2$$

$$b_2 = 0.6 \quad u_2 = 1$$

$$b_3 = 0.5 \quad u_3 = 1$$

$$T = \sum_{k=1}^3 u_k = 4$$

$$w_1(S3) = Q\left(\frac{2}{4}\right) - Q\left(\frac{0}{4}\right) = \frac{1}{4}$$

$$w_2(S3) = Q\left(\frac{3}{4}\right) - Q\left(\frac{2}{4}\right) = \frac{5}{16}$$

$$w_3(S3) = Q(1) - Q\left(\frac{3}{4}\right) = 1 - \left(\frac{3}{4}\right)^2 = \frac{7}{16}$$

$$D_{1wt}(S3) = \sum_{j=1}^3 b_j w_j(S3) = 0.8\left(\frac{1}{4}\right) + 0.6\left(\frac{5}{16}\right) + 0.5\left(\frac{7}{16}\right) = \frac{9.5}{16} = 0.594$$

Client Experience(ce) Support Aggregation

Since this is a mean aggregation, we can do a simple calculation as below (we would get the same result with $\alpha = 1$, if we do a OWA aggregation).

$$D_{lce}(S1) = \frac{0.6 + 0.7 + 0.7}{3} = .67$$

$$D_{lce}(S2) = \frac{0.7 + 0.8 + 0.3}{3} = 0.6$$

$$D_{lce}(S3) = \frac{0.6 + 0.6 + 0.9}{3} = 0.7$$

Spend Fitness (sp) Aggregation

This is a mean aggregation; hence we do simple calculation as below

$$D_{1sp}(S1) = \frac{0.9 + 0.5}{2} = 0.7$$

$$D_{1sp}(S2) = \frac{0.7 + 0.9}{2} = 0.8$$

$$D_{1sp}(S3) = \frac{0.5 + 0.8}{2} = 0.65$$

Level 2 Aggregations

As in Level 1, we use Equations (6) for aggregation, (2) for determining weights, (3) for ORness and (4) for relative importance while performing aggregations. At level 2, C_n are (a). *Strategic Fitness* and (b). *Execution Fitness*. We will aggregate individually on these criteria.

Strategic Fitness (SF) Aggregation

Alternative S1

$$b_1 = 0.685 \quad u_1 = 1$$

$$b_2 = 0.67 \quad u_2 = 1$$

$$b_3 = 0.544 \quad u_3 = 1$$

$$T = \sum_{k=1}^3 u_k = 3$$

$$w_1(S1) = Q\left(\frac{1}{3}\right) - Q\left(\frac{0}{3}\right) = \left(\frac{1}{3}\right)^2 - \left(\frac{0}{4}\right)^2 = \frac{1}{9} \dots \alpha = 2, \text{ for "Most"}$$

$$w_2(S1) = Q\left(\frac{2}{3}\right) - Q\left(\frac{1}{3}\right) = \left(\frac{3}{9}\right) = \frac{1}{3}$$

$$w_3(S1) = Q(1) - Q\left(\frac{2}{3}\right) = 1 - \left(\frac{4}{9}\right) = \frac{5}{9} \dots$$

$$D_{2SF}(S1) = \sum_{j=1}^3 b_j w_j(S1) = 0.685\left(\frac{1}{9}\right) + 0.67\left(\frac{1}{3}\right) + 0.544\left(\frac{5}{9}\right) = 0.541$$

Alternative S2

$$b_1 = 0.69 \quad u_1 = 1$$

$$b_2 = 0.6 \quad u_2 = 1$$

$$b_3 = 0.587 \quad u_3 = 1$$

$$T = \sum_{k=1}^3 u_k = 3$$

Coincidentally $w_1(S2), w_2(S2), w_3(S2)$. and $w_1(S3), w_2(S3), w_3(S3)$. are same as for $w_1(S1), w_2(S1), w_3(S1)$. respectively so our calculations appear as below

$$D_{2SF}(S2) = \sum_{j=1}^3 b_j w_j(S1) = 0.69\left(\frac{1}{9}\right) + 0.6\left(\frac{1}{3}\right) + 0.587\left(\frac{5}{9}\right) = 0.603$$

Alternative S3

$$b_1 = 0.7 \quad u_1 = 1$$

$$b_2 = 0.647 \quad u_2 = 1$$

$$b_3 = 0.594 \quad u_3 = 1$$

$$D_{2SF}(S3) = \sum_{j=1}^3 b_j w_j(S1) = 0.7\left(\frac{1}{9}\right) + 0.647\left(\frac{1}{3}\right) + 0.594\left(\frac{5}{9}\right) = 0.623$$

Execution Fitness (EF) Aggregation

Alternative S1

$$b_1 = 0.8 \quad u_1 = 1$$

$$b_2 = 0.7 \quad u_2 = 1$$

$$b_3 = 0.6 \quad u_3 = 1$$

$$T = \sum_{k=1}^3 u_k = 3$$

$$w_1(S1) = Q\left(\frac{1}{3}\right) - Q\left(\frac{0}{3}\right) = \left(\frac{1}{3}\right)^4 - \left(\frac{0}{3}\right)^4 = \frac{1}{81} = 0.012 \text{ as } \alpha = 4 \text{ is taken for this}$$

strict satisfaction expectation

$$w_2(S1) = Q\left(\frac{2}{3}\right) - Q\left(\frac{1}{3}\right) = \left(\frac{16}{81}\right) - \left(\frac{1}{81}\right) = \frac{15}{81} = 0.185$$

$$w_3(S1) = Q(1) - Q\left(\frac{2}{3}\right) = 1 - \left(\frac{16}{81}\right) = 0.80$$

$$D_{2EF}(S1) = \sum_{j=1}^3 b_j w_j(S1) = 0.8(0.012) + 0.7(0.185) + 0.6(0.8) = 0.779$$

Coincidentally $w_1(S2), w_2(S2), w_3(S2)$. and $w_1(S3), w_2(S3), w_3(S3)$. are same as for $w_1(S1), w_2(S1), w_3(S1)$. respectively so our calculations appear as below

Alternative S2

$$b_1 = 0.8 \quad u_1 = 1$$

$$b_2 = 0.8 \quad u_2 = 1$$

$$b_3 = 0.5 \quad u_3 = 1$$

$$D_{2EF}(S2) = \sum_{j=1}^3 b_j w_j(S2) = 0.8(0.012) + 0.8(0.185) + 0.5(0.8) = 0.557$$

Alternative S3

$$b_1 = 0.8 \quad u_1 = 1$$

$$b_2 = 0.7 \quad u_2 = 1$$

$$b_3 = 0.65 \quad u_3 = 1$$

$$D_{2EF}(S3) = \sum_{j=1}^3 b_j w_j(S3) = 0.8(0.012) + 0.7(0.185) + 0.65(0.8) = 0.659$$

Level 3 Aggregation i.e. at Program (prg) Level

This is done by mean aggregation with importance assigned to Strategic & Execution fitness in the ratio of 4:6.

$$D_{prg}(S1) = 0.4(0.541) + 0.6(0.779) = 0.683$$

$$D_{prg}(S2) = 0.4(0.603) + 0.6(0.557) = 0.575$$

$$D_{prg}(S3) = 0.4(0.623) + 0.6(0.659) = 0.644$$

Goal satisfaction at the program level is highest for Alternative S1 as $D_{prg}(S1) > D_{prg}(S3), D_{prg}(S2)$.

Thus, Alternative S1 should be the preferred alternative for the program.

4.5 Discussion

We started with a need to plan a system to respond to a specific health sector initiative. Given the goals, we developed alternative system concepts using GORE approaches discussed in Chapter 3. The conceptual systems are composed of major components, decomposed to a level amenable to operationalization and quantifiable assessments. Next, we developed evaluation criteria based on the goals and developed a hierarchical structure of the criteria. The criteria and the structure determined the parameters (α , and type of averaging) that we should use for aggregation of satisfaction influence from leaf level up in the decision criteria structure. We then got assessments from experts on potential satisfaction of the individual leaf level criteria for each individual alternative. We aggregated from leaf to the next level, and then up. Thus we propagated the influences up to the apex level in steps. Final result gave us the potential satisfaction of goals that the alternative is supposed to achieve. We compared the results from all alternatives to determine the potentially most satisfactory alternative. This alternative is now available for Operationalization.

The approach gave us early assessment of good candidate alternative by using a generalized formulation. The formulation is suitable for multiple objectives and constraints, which we structure as attributes. Criteria structuring makes it possible to proceed via simple and methodical steps. Since the criteria nodes and leaves represent goals, the formulation does a goal-based assessment.

Approaches commonly practiced for similar decisions are prioritization and trade-off based. A commonly used simplification for alternatives selection is the high-medium-low priority classification of individual requirement. Pitfall apart from evaluators' individual biases about priority is the potential assumption of requirements at a single level and not doing apples to apples comparison. Another prioritization approach is that of scoring the alternative on chosen criteria by (subjective or objective) assessments and then doing an indexed (e.g. benefit/cost) or some formulated score (e.g. weighted average of cost score, time score, application stability score etc.). This again results in potentially aggregating scores at different levels if we do not consider the criteria structure.

Trade-offs based selection represents complex human thinking where the evaluators discard alternatives based on relative preference of a criterion versus another [18]. Such techniques when

normalized for subjectivity can work, though will become complex to handle when large number of attributes are present and are not structured in a hierarchy. A refinement of the relative importance philosophy is the more formalized Analytic Hierarchy Process (AHP) and has been proposed for requirements selection.

Some proposals exist for quantitative assessment of alternatives in extension of GORE methods. Letier et al's [25] proposal suggested for NFR, is predicated on decomposition of goals to levels where they can be completely defined by AND and OR components. The probability of satisfaction is then propagated leaf up via computed probability distribution function (pdf) at next level, with application specific formulation. While specificity makes great sense if proven domain models exist, it also increases the level of computational effort. We also did not find any method for generalized requirements of compensatory decomposition which cannot be modeled by purely AND or OR operations. Giorgini et al [16] propose an approach for reasoning NFR via propagation of "satisfiability" and "deniability" influence of component AND /OR goals to related goal. We need to study the approach further as far as propagation formulation types are concerned to verify if the method provides for aggregations that are required to reason with multiple objectives. As stated at the end of the Case Study I in section 3.3, the proposed tool could not be worked with due to lack of support.

Now we review some additional complexities that may be encountered. We assumed that it was possible to make assessments of satisfaction level of alternatives on chosen criteria. How do we make those assessments and how do we normalize the varied assessments of individuals if it is a group (of subject matter experts, or/and non expert stakeholders) decision. Normalization could be in terms own subjective bias, scale used, contextual assumptions made, goal interpretation and so forth. Interpretation of goals can be partially addressed by contextual dictionary, communication or facilitated communication processes such as walkthroughs. In a rigorous exercise one may adopt techniques such as Repertory Grid Techniques (RGT) which is discussed in requirements engineering context in [28]. A technique for reducing subjectivity could be to consider discarding one assessment observation on each extreme. Delphi techniques [42] have been practiced to normalize subjectivity. AHP has found extensive application in responding to such concerns in group decision [34]. Human mind makes complex decisions in wide range of activities on a day-to-day basis, evaluating pros and cons all the time. This could be interpersonal interactions or conscious decisions such as route to take to reach a destination in comfort, time, and desired cost. Extension of the phenomenon is the basis of assumption that it is possible for

evaluators to make such assessments. Some underlying formalization guidance could be required while evaluators are making assessments that have complexities of large number of dependencies.

OWA aggregation has been widely used in various industries for selection decisions. Yager [47] extended the OWA based aggregation to model decisions where both category prioritization and importance are involved. We worked with α parameter values associated with commonly established quantifiers. Researchers are increasingly adding more quantifiers. It is also possible to derive α from observations and from knowledge-based rules. Thus, an approach based on OWA is attractive. We need some more exploration of the approach in context of requirements selection. We worked with satisfaction membership. Working with integration of counter influences during aggregation and working on flexible decision structures are some of the identified future initiatives.

5 Conclusion

Early stage realities and business considerations require an approach that is quick to find pertinent coarse requirements, which can be reasoned with respect to early stage decision frame. We walked through an approach for developing pertinent requirements using GORE concepts. We discussed development and structuring of decision criteria in system requirements context. We used powerful fuzzy aggregation techniques that have been successful in wide variety of selection decisions in large number of alternative industries. We walked through a case study to illustrate the method.

The method based on Goals and decision concepts here can become powerful for general-purpose use in systems development decisions. Identified potential investigations in this regard include incorporating counterinfluences, interval value assessments and probabilistic assessments. Also identified is the need for generalized repertory of decision attitudes and related quantifiers.

Proposed method would achieve fast stakeholder comprehension and decision evaluation by working with goal abstractions, stakeholder intentions, limited span of goal task elaborations, standard quantifiers and in the process may compromise rigor and formalism to an extent. For a class of applications and for large number commercial environments this ability to speed up is a valid and desirable proposition. This may violate our existing mindset or preoccupation with rigorously definable and verifiable specifications right through the life cycle. We argue that this preoccupation comes from historical coding driven systems development contexts. In a keynote at ICSE 06, Boehm [6] discussed the future trends in software engineering and emergence of services and components based assembly of application systems with a need for quick turnaround. Precision and measurability aimed by rigorous formalized methods may not be achievable or impractical for a class of applications in required decision timeframes. In such cases we propose our methodology to get to pertinent high-level requirements first, which can then be formally defined and specified.

6 Bibliography

1. Anton, A. I. (1996). Goal-based requirements analysis. *Requirements Engineering, 1996., Proceedings of the Second International Conference on*, 136-144.
2. Anton, A. I., & Potts, C. (1998). Use of goals to surface requirements for evolving systems. *Proceedings of the 1998 International Conference on Software Engineering*, 157-166.
3. Bas, E., & Kahraman, C. (2009). Fuzzy capital rationing model. *Journal of Computational and Applied Mathematics*, 224(2), 628-45.
4. Bellman, R. E., & Zadeh, L. A. (1970). *Decision-making in a fuzzy environment* No. NASA-CR-1594). USA: California Univ., Berkeley, CA, USA.
5. Berntsson-Svensson, R., & Aurum, A. (2006). Successful software project and products: An empirical investigation. *ISCE'06 - 5th ACM-IEEE International Symposium on Empirical Software Engineering 2006*, 144-153.
6. Boehm, B. (2006). A view of 20th and 21st century software engineering. *28th International Conference on Software Engineering 2006, ICSE '06*, 12-29.
7. Boehm, B., Valerdi, R., & Honour, E. (2008). The ROI of systems engineering: Some quantitative results for software-intensive systems. *Systems Engineering*, 11(3), 221-234.

8. Booch, G. (1994). *Object-oriented analysis and design with applications* (2nd ed.). Redwood City, Calif.: Benjamin/Cummings Pub. Co.
9. Bresciani, P., Perini, A., Giorgini, P., Giunchiglia, F., & Mylopoulos, J. (2002). Modeling early requirements in tropos: A transformation based approach. *Agent-Oriented Software Engineering II. Second International Workshop, AOSE 2001. Revised Papers and Invited Contributions (Lecture Notes in Computer Science Vol.2222)*, 151-68.
10. Bresciani, P., Perini, A., Giorgini, P., Giunchiglia, F., & Mylopoulos, J. (2004). Tropos: An agent-oriented software development methodology. *Autonomous Agents and Multi-Agent Systems*, 8(3), 203-36.
11. Dardenne, A., van Lamsweerde, A., & Fickas, S. (1993). Goal-directed requirements acquisition. *Science of Computer Programming*, 20(1-2), 3-50.
12. DeMarco, T. (1979). *Structured analysis and system specification*. Englewood Cliffs, N.J.: Prentice-Hall.
13. El, E., Khaled, & Koru, G. A. (2008). A replicated survey of IT software project failures. *IEEE Software*, 25(5), 84-90.
14. Fung, L.W., Fu, K. S., (1975), An Axiomatic Approach To Rational Decision Making in A Fuzzy Environment. In Zadeh L.A. (Ed), *Fuzzy sets and their applications to cognitive and decision processes* (pp. 227-256), New York: Academic Press.
15. Giorgini, P., Kolp, M., Mylopoulos, J., & Pistore, M. (2004). The tropos methodology: An overview. *Methodologies and software engineering for agent systems. the agent-oriented software engineering handbook* (pp. 89-106). Dordrecht, Netherlands: Kluwer Academic Publishers.

16. Giorgini, P., Mylopoulos, J., Nicchiarelli, E., & Sebastiani, R. (2002). Reasoning with goal models. *Conceptual Modeling - ER 2002. 21st International Conference on Conceptual Modeling. (Lecture Notes in Computer Science Vol.2503)*, 167-81.
17. *A guide to the project management body of knowledge : PMBOK guide.*(2004). In Project Management Institute. (Ed.), (3rd ed.). Newtown Square, Pa.: Project Management Institute.
18. Hammond, J. S., Keeney, R. L., & Raiffa, H. (1998). Even swaps: A rational method for making trade-offs. *Harvard Business Review*, 76(2), 137.
19. Henderson, J. C., & Venkatraman, N. (1993). Strategic alignment: Leveraging information technology for transforming organizations. *IBM Systems Journal*, 32(1), 4-16.
20. Holbrook, H., (1990). A scenario-based methodology for conducting requirements elicitation. *SIGSOFT Software Engineering Notes*, 15(1), 95-104.
21. Jacobson, I., Booch, G., & Rumbaugh, J. (1999). *The unified software development process*. Reading, Mass: Addison-Wesley.
22. Kahraman, C. (2007). Fuzzy set applications in industrial engineering. *Information Sciences*, 177(7), 1531-1532.
23. Kitapci, H., & Boehm, B. W. (2007). Formalizing informal stakeholder decisions - A hybrid method approach. *40th Annual Hawaii International Conference on System Sciences 2007, HICSS'07*, 4076956.
24. Kotonya, G., & Sommerville, I. (1992). Viewpoints for requirements definition. *Software Engineering Journal*, 7(6), 375-87.

25. Letier, E., & Van Lamsweerde, A. (2004). Reasoning about partial goal satisfaction for requirements and design engineering. *Twelfth ACM SIGSOFT International Symposium on the Foundations of Software Engineering, SIGSOFT 2004/FSE-12*, 53-62.
26. Misic, M. M., & Graf, D. K. (2004). Systems analyst activities and skills in the new millennium. *Journal of Systems and Software*, 71(1-2), 31-6.
27. Misic, M. M., & Russo, N. L. (2000). Reading between the lines: An examination of systems analysis and design texts. *Journal of Systems and Software*, 50(1), 65-73.
28. Niu, N., & Easterbrook, S. (2007). Analysis of early aspects in requirements goal models: A concept-driven approach. *Transactions on Aspect-Oriented Software Development III, Lecture Notes in Computer Science Vol. 4620*, 40-72.
29. Procaccino, J. D., Verner, J. M., & Lorenzet, S. J. (2006). Defining and contributing to software development success. *Communications of the ACM*, 49(8), 79-83.
30. Reel, J. S. (1999). Critical success factors in software projects. *Software, IEEE*, 16(3), 18-23.
31. Rolland, C., Achour, C. B., Cauvet, C., Ralyte, J., Sutcliffe, A., Maiden, N., et al. (1998). A proposal for a scenario classification framework. *Requirements Engineering*, 3(1), 23-47.
32. Rolland, C., Souveyet, C., & Achour, C. B. (1998). Guiding goal modeling using scenarios. *IEEE Transactions on Software Engineering*, 24(12), 1055-1071.
33. Roy, J., Kealey, J., & Amyot, D. (2006). Towards integrated tool support for the user requirements notation. *Revised Selected Papers*, 198-215.
34. Saaty, T. L., & Shang, J. S. (2007). Group decision-making: Head-count versus intensity of preference. *Socio-Economic Planning Sciences*, 41(1), 22-37.

35. Sharp, H., Finkelstein, A., & Galal, G. (1999). Stakeholder identification in the requirements engineering process. *Tenth International Workshop on Database and Expert Systems Applications 1999*, 387-91.
36. Sikdar, S., & Das, O. (2008). Case for enriching developer's toolkit with goal oriented requirement methodologies. *IEEE Canadian Conference on Electrical and Computer Engineering, CCECE 2008*, 743-746.
37. Sikdar, S., & Das, O. (2009). Stakeholder appropriate requirement development approach. *IEEE International Advance Computing Conference, IACC 2009*, 1670-4.
38. Slowinski, R. (2001). Fuzzy sets in decision analysis, operations research and statistics. *Fuzzy Sets and Systems*, 123(3), 407-409.
39. Sowa, J. F., & Zachman, J. A. (1992). A logic-based approach to enterprise integration. *Proceedings of the First International Conference Enterprise Integration Modeling*, 1992, 152-63.
40. Sutcliffe, A. (2003). Scenario-based requirements engineering. *Proceedings 11th IEEE International Requirements Engineering Conference*, 2003, 320-9.
41. Sutcliffe, A., Fickas, S., & Sohlberg, M. M. (2006). PC-RE: A method for personal and contextual requirements engineering with some experience. *Requirements Engineering*, 11(3), 157-73.
42. Tomlinson, C. M., & Johnson, L. (1994). Notes on the techniques adopted for knowledge elicitation. *Systems Research and Information Science*, 6(4), 179-85.
43. Van Lamsweerde, A. (2001). Goal-oriented requirements engineering: A guided tour. *5th IEEE International Symposium on Requirements Engineering*, 2001, 249-261.

44. Verner, J., Sampson, J., & Cerpa, N. (2008). What factors lead to software project failure? *2008 Second International Conference on Research Challenges in Information Science*, 71-80.
45. Verner, J. M., & Evanco, W. M. (2005). In-house software development: What project management practices lead to success? *IEEE Software*, 22(1), 86-93.
46. Yager, R. R. (1988). On ordered weighted averaging aggregation operators in multicriteria decisionmaking. *Systems, Man and Cybernetics, IEEE Transactions on*, 18(1), 183-190.
47. Yager, R. R. (2008). Prioritized aggregation operators. *International Journal of Approximate Reasoning*, 48(1), 263-74.
48. Yager, R. R. (1998). Including importances in OWA aggregations using fuzzy systems modeling. *IEEE Transactions on Fuzzy Systems*, 6(2), 286-294.
49. Yoon, K., & Hwang, C. L. (1995). *Multiple attribute decision making : An introduction*. Thousand Oaks, CA: Sage Publications.
50. Yu, E. S. K. (1997). Towards modelling and reasoning support for early-phase requirements engineering. *Proceedings of the 1997 3rd International Symposium on Requirements Engineering*, 1997, 226-235.
51. Yu, Eric Siu-Kwong (1995). Modelling strategic relationships for process reengineering. Ph.D. dissertation, University of Toronto (Canada), Canada. Retrieved April 20, 2009, from Dissertations & Theses: Full Text database. (Publication No. AAT NN02887).
52. Yue, K. (1987). What does it mean to say that a specification is complete? *Proceedings - Fourth International Workshop on Software Specification and Design*, 1987, 42-49.

53. Zachman, J. A. (1999). Framework for information systems architecture. *IBM Systems Journal*, 38(2-3), 454-470.
54. Zadeh, L. A. (1983). A computational approach to fuzzy quantifiers in natural languages. *Computers & Mathematics with Applications*, 9(1), 149-84.
55. Zimmermann, H. -. (1987). *Fuzzy sets, decision making, and expert systems*. Boston, Kluwer Academic Publishers.