Enhancing the Vision Document in the Rational Unified Process with a Visual Representation of Goals

Kendra Cooper, Lawrence Chung, Sam Courtney

Abstract - The Rational Unified Process is a comprehensive process model that is tailorable, provides templates for the software engineering products, and integrates the use of the Unified Modeling Language (UML). The process supports the definition of requirements at multiple levels. Currently, early requirements, or goals, are captured in a textual document called the Vision Document; a template for this document is available. In this paper, we present an enhanced version of the Vision Document template, which provides a visual representation of the functional and non-functional goals of the system in addition to the textual representation already available. To accomplish this, two existing visual notations are adopted into the Vision Document template. A section for representing the functional goals using an AND/OR Graph is added; a section for representing the non-functional goals using a Softgoal Interdependency Graph is also added. The relationships between the visual and textual sections are described; the need for consistency between them is addressed by providing guidelines to the Analyst. The enhanced template is illustrated using an example system called the Quality Assurance Review Assistant Tool.

***Index Terms*— goal-oriented requirements engineering, Rational Unified Process, Unified Modeling Language, visual modeling**

I. Introduction

Ideally speaking, a software system is built to help the stakeholders of the system achieve their *goals*. For example, a stakeholder may need a system to provide a variety of searching capabilities and be easy to use. When using the Rational Unified Process (RUP), such goals can be captured in the Vision Document, very early in the development lifecycle. Once defined, the goals are subsequently refined and captured in additional models. For example, the functional goals (e.g., searching) are refined and captured in a Use Case model and the non-functional goals are refined and captured in a Special Requirements document.

Currently, the representation of the goals in the textual Vision Document seems to be somewhat limited, as visual models can alleviate some of the difficulties in human communication. This gap could be readily addressed by integrating existing visual representation into the Vision Document.

K. Cooper is with The University of Texas at Dallas, Richardson, Tx, USA, (e-mail kcooper@utdallas.edu; phone 972 883 4216, fax 972 883 2349).
L. Chung is with The University of Texas at Dallas, Richardson, Tx, USA, (e-mail chung@utdallas.edu).
S. Courtney is with Tenfold Inc., Carrollton, Texas, USA, (e-mail: samcourtney@hotmail.com)

Although we find little other work specifically on the visual modeling of goals in the vision document within RUP, there are several important advances in the area of early Requirements Engineering that have some relevance to our work. GRAIL/KAOS [4] offers an AND/OR graph [8] for stating functional requirements as goals and refining them through logical conjunction and disjunction. A visual modeling of stakeholders as agents is pioneered in an agent-oriented notation called i* [13], in which an agent can have goals, carry out tasks to achieve the goals, use resources in carrying out the tasks, and depend on other agents to carry out the tasks, etc. The work in [6] describes how to integrate non-functional requirements, as expressed in a SIG, and the informational aspect of functional requirements, as expressed in an entity relationship diagram. A methodology for dealing with non-functional requirements together with use cases is presented in [10].

Here, we present an enhanced template for the Vision Document, which retains the detailed textual descriptions of the stakeholders and their goals, already defined in the Vision Document template, while adding complementary, existing visual representations for the goals. Functional goals are presented using an AND/OR graph; non-functional goals are represented using the Softgoal Interdependency Graph (*SIG*), which is part of the NFR Framework [3].

A Quality Assurance Review Assistant Tool (QARAT) is used as a running example throughout the paper. A QARAT supports capturing and managing the results of conducting quality assurance inspections, or reviews, of various software engineering artifacts (e.g., requirements model, design model, etc.). Reviewers can conduct their inspections remotely and submit their comments, defects identified, etc. using the QARAT.

The remainder of this paper is organized as follows. In Section II, an overview of how early requirements are represented in the RUP using a Vision Document, a Softgoal Interdependency Graph (SIG), and an AND/OR Graph is presented. The Enhanced Vision Document is presented in Section III. Conclusions and Future work are in Section IV.

II. Representing Early Requirements

Early requirements, or goals, have been represented in both text- and graphically-based notations. In this work, we draw upon several existing approaches used to represent early requirements. The first one is the Rational Unified Process'

Vision Document, which is text-based representation. The second one is the Non-functional Requirement (NFR) Framework's Softgoal Interdependency Graph (SIG), which is a graphic notation used to represent non-functional goals. The third is the AND/OR Graph, which is a graphic notation used to represent functional goals. An overview of these is provided in this section; detailed presentations are available elsewhere [1][3][5][8].

A. RUP and the Vision Document

The RUP is a comprehensive software development approach that is iterative, use-case driven, and architecture-centric [5]. It is organized along two dimensions, horizontal and vertical. The horizontal dimension represents the dynamic structure; it is described in terms of phases, iterations, and milestones. The four phases in the RUP are inception, elaboration, construction, and transition. The vertical dimension represents the static structure. Here, disciplines, or high level workflows, such as business modeling, requirements, analysis and design, etc. are described in terms of the roles, activities, artifacts, workflows, and additional process elements (templates, guidelines, concepts, roadmaps, and help on tools).

The early requirements for a system are defined during the inception phase, as part of the Business Modeling discipline. The role responsible for this work is an Analyst. A Vision Document is defined to clarify the business benefits that will be provided by building the application, the problem(s) the application will solve, who the target users would be, and the high level capabilities of the application.

A word template for the Vision Document is available for the Analyst to use, and if necessary, tailor. The template has sections clearly defined for each of these elements and provides guidelines on the content for each section. For example, Section 4 of the Vision Document, called the Product Overview, is organized into the following subsections: Product Perspective, Summary of Capabilities, Assumptions and Dependencies, and Cost and Pricing. Each subsection contains guidelines for the Analyst. For example, the Summary of Capabiltities section contains the following guidelines; part of the example table provided is also shown (refer to Table I):

"[Summarize the major benefits and features the product will provide. For example, a Vision document for a customer support system may use this part to address problem documentation, routing, and status reporting without mentioning the amount of detail each of these functions requires.

Organize the functions so the list is understandable to the customer or to anyone else reading the document for the first time. A simple table listing the key benefits and their supporting features might suffice. For example:]"

TABLE I Summary of Capabilities (example) [9]

Customer Benefit	Supporting Features
New support staff can quickly get up to speed.	Knowledge base assists support personnel in quickly identifying known fixes and workarounds.
Customer satisfaction is improved because nothing falls through the cracks.	Problems are uniquely itemized, classified and tracked throughout the resolution process. Automatic notification occurs for any aging issues.
Management can identify problem areas and gauge staff workload.	Trend and distribution reports allow high level review of problem status.
…	

The Vision Document provides a common, high level understanding of what needs to be built, who the stakeholders are, and why the application needs to be built. It is used as a working document to subsequently identify and define the requirements specifications in a use case model (primarily the functional requirements) and supplementary requirements description (non-functional requirements), the analysis and design of the system, implementation, and test as the development moves through the inception, elaboration, and construction phases.

B. AND/OR Graph

Directly borrowed from problem reduction methods in Artificial Intelligence [8], AND/OR graphs may be used to capture goal *refinement* links [7] (refer to Figure 2). Based on classical logical, *AND-refinement* links relate a goal to a set of subgoals (called refinement); this means that satisfying all subgoals in the refinement is sufficient for satisfying the parent goal. *OR-refinement* links relate a goal to an alternative set of refinements; this means that satisfying one of the refinements is sufficient for satisfying the parent goal. A *conflict* link between two goals is introduced when the satisfaction of one of them may prevent the other from being satisfied. AND, OR,

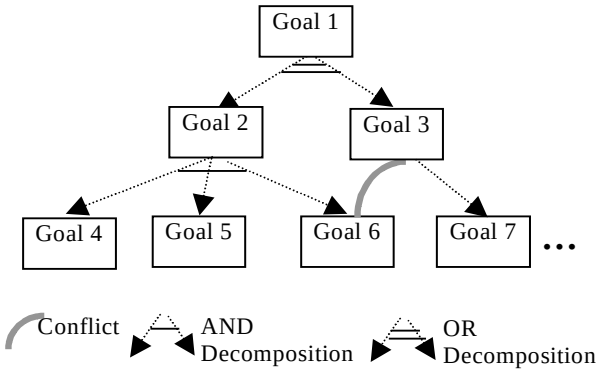

Figure 1 AND/OR Graph

and conflict links are used to capture alternative goal refinements, potential conflicts, and to prove the correctness of goal refinements.

C. *Softgoal Interdependency Graph*

A Softgoal Interdependency Graph (SIG) is much like the AND/OR tree used in problem-solving [8]. Unlike traditional problem-solving and planning frameworks, however, softgoals can rarely be said to be “accomplished” or “satisfied” in a clear-cut sense. Instead, different design decisions contribute positively or negatively towards a particular goal. Softgoals are based on the notion of goal "satisficing" (as defined in [11]) to suggest that generated software is expected to be acceptable, or more simply “good enough”. For non-functional goals, a softgoal is considered to be a closer representation. For example, it is very difficult to answer the questions “Is the system secure?” or “Is the system fast?” with a simple yes or no answer.

The atomic elements of a SIG include softgoals, operationalizations (i.e., possible solutions), and claims (i.e., rationale or justification). These kinds of goals may be augmented with topics (relationships to other elements including functional goals, agents, etc.) and prioritizations.

To represent the complex relationships among goals, softgoals, claims, and operationalizations may be decomposed with n-ary AND and OR relationships. In addition, the contributions from one goal to accomplishing another can be represented. There are five kinds of contributions: “helps” (+), “makes” (++), unknown (?), “hurts” (-), and “breaks” (--). A simple, symbolic example of a SIG is in Figure 1. Mathematical definitions of the interdependencies (helps, hurts, etc.) are available in [1].

Previous experience in defining non-functional requirements is captured in re-usable knowledge catalogues. There are three kinds of catalogues in the Framework: type, method, and correlation catalogues. Type catalogues provide the terminology and classification for non-functional concepts, such as a hierarchy including Performance, Cost, Security, etc. Method catalogues encode knowledge by decomposing softgoals and considering operationalizations. Correlation rule catalogues encode knowledge that helps to detect implicit interdependencies among softgoals. For example, increasing security hurts the ease of use and ease of use hurts modest cost. The catalogues may be adopted as is, or tailored as needed. Consequently, the catalogues can be extended to encode new knowledge.

A SIG can also be evaluated by a procedure that propagates the labels of subgoals to their parent goals, where each label indicates the degree to which its associated goal is satisficed. Simple evaluation cases for AND and OR decompositions are discussed here. For example, consider a softgoal that is refined with an AND decomposition. If all of the subgoal labels have been evaluated as being satisficed, then the label propagation rules result in the parent being evaluated as satisficed as well. Alternatively, consider a softgoal that is refined with an OR decomposition. If at least one of the subgoal labels has been

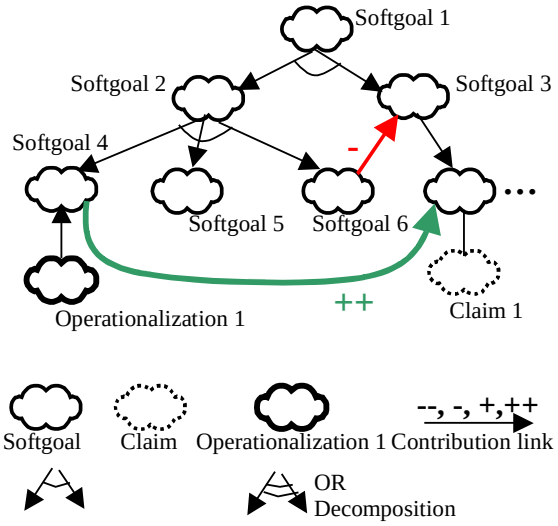

Figure 2 Softgoal Interdependency Graph

evaluated as being satisficed, then the label propagation rules result in the parent being evaluated as satisficed as well. More complex cases of the propagation rules are presented in [3].

III. Enhanced Vision Document

The Vision Document is intended to be a means to communicate a common understanding of the system. In order to improve its usefulness, the linear, textual description of the agents and goals described is enhanced with corresponding visual models. Here, we summarize the enhancements to the Vision Document using a Quality Assurance Review Assistant Tool (QARAT) as an example. To aid the reader in understanding the structural changes to the Vision Document, the Table of Contents of the Enhanced Vision Document is available in Appendix A; the modified and new sections are highlighted in blue font.

A. *Product Overview Section*

The Product Overview (Section 4 in the Vision Document) provides a place to present a brief summary description of the functional (and non-functional) goals called the Summary of Capabilities section. The purpose and content of this section remains the same, however, additional Guidelines are provided to improve the consistency of the document. Here, we propose the following guidelines to ensure the capabilities listed in this section become the top level functional and non-functional goals which are decomposed in later sections:

Each functional capability listed in this section should appear as a subheading in the Functional Product Feature (textual) section.

Each functional capability listed in this section should appear as a goal icon in the Functional Product Feature (visual) section.

Each non- functional capability listed in this section should appear as a subheading in the Non-functional Product Feature (textual) section.

Each non- functional capability listed in this section should appear as a softgoal icon in the Non-functional Product Feature (visual) section.

For example, the QARAT system's summary of capabilities is presented in Table II.

TABLE II QARAT: Summary of Capabilities

Secured access to the system	*QARAT only allows authorized users to access the system, system is available, transactions are secured. Security is needed to support the intellectual property of the development house.*
Distributed support teams can work together to solve problems.	*Replication server allows current database information to be shared across the enterprise.*
Convenient, flexible access to the system.	*Database can be accessed over the Internet, which also includes hypertext search capabilities and graphical query engine.*
Assist in reviewing SEAs to provide high quality work product.	***Manage User Accounts*** *QARAT provides capabilities to manage User Accounts – Add User Account, Update User Account, Delete User Account.* ***Manage SEA reviews*** *The QARAT provides capabilities to read all software engineering artifacts (SEA) and write reviews of SEAs and delete or edit reviews. The QARAT provide capabilities to baseline the reviews. Reviews can be edited until they are baselined. It cannot be deleted or changed after a version is baselined* …

The purpose of the Product Features section in the original Vision Document is to provide a place to describe the functional capabilities of the system in text. This section has been organized into a Product Features (textual) and a Product Features (visual) section in the Enhanced Vision Document (Sections 5 and 6)

B. *Functional Product Features*

The purpose of the Product Features section in the original Vision Document is to provide a place to describe the functional capabilities of the system in text. This section has been organized into a Product Features (textual) and a Product Features (visual) section in the Enhanced Vision Document (Sections 5 and 6). These functional goals can subsequently be refined into use cases as the software development progresses.

1) *Functional Product Features (visual)*

The new visual product feature section is intended to provide a place to describe the functional capabilities in an easy to read and understand, graphical representation. The Guidelines in this section are:

Each functional capability appearing as an icon in this section should appear in a subheading in the Functional Product Feature (textual) section.

Describe why each functional capability is needed by the stakeholders (this is a claim on the visual model)

The functional capabilities should have a priority ranking

The level of decomposition in the graph should be represented by the level of subheading used in the textual section. The level of decomposition should be consistent with the level of decomposition of the textual representation (i.e., second level subheading should have an icon in the second level of the graph in the visual section)

For example, in the QARAT system, part of the Functional Product Features (visual) is illustrated in Figure 3. To represent the high priority nature of the goals, we borrow the "!" syntax from the SIG.

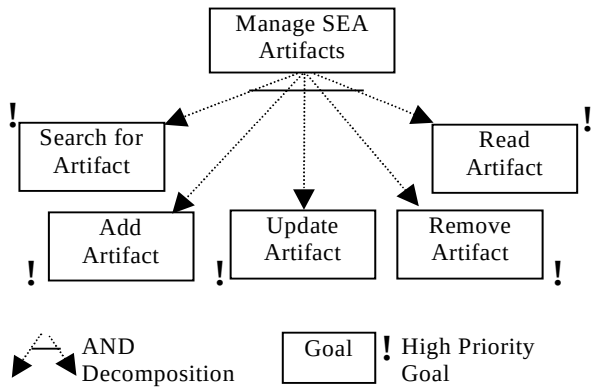

Figure 3 QARAT Functional Features

2) *Functional Product Features (textual)*

The textual product feature section is intended to provide a place to describe the functional capabilities in more detail. This can alleviate the need to include everything on a visual model, while still providing a complete description. This allows the visual model to be easier to read and understand. New guidelines added into the textual section are:

Each functional capability listed in this section should appear as a goal icon in the Functional Product Feature (visual) section.

The functional capabilities should have a priority ranking

Describe why each functional capability is needed by the stakeholders (this is a claim on the visual model)

The functional capabilities should have a priority ranking

When decomposed, the type of decomposition (and, or) should

be described with the keyword "AND" or "OR"

The level of decomposition in the graph (visual section) should be represented by the level of subheading used. The level of decomposition should be consistent with the level of decomposition of the visual representation (i.e., second level subheading should have an icon in the second level of the graph in the visual section)

For example, in the QARAT system, the following functional product features are described. They are a refinement of the summary capability, Manage User Accounts:

…

```
6.3 The following features relate to
    manage software engineering artifact
    (SEA) reviews
```

Decomposition type: AND

```
 6.3.1 Search for Artifact
Search is done with SEA ID as the
Keyword. A list of documents shows up in
order of relevance.
Claim: Fundamental capability needed by
 the reviewers.
 High priority goal.

 6.3.2 Read Artifact
User retrieves the artifacts he or she
wants to review.
 Claim: Fundamental capability needed by
the reviewers.
High priority goal.

 6.3.3 Add Artifact
SEA is uploaded by the Author and is
displayed in the QARAT SEA list.
 Claim: Fundamental capability needed by
the reviewers.
High priority goal.

 6.3.4 Update Artifact
Author selects the artifacts to update
and save changes to the system
 Claim: Fundamental capability needed by
the reviewers.
High priority goal.

 6.3.5 Remove Artifact
Author permanently removes the SEA from
the QARAT system.
 Claim: Fundamental capability needed by
the reviewers.
High priority goal.
```

…

C. *Non-functional Product Features*

The non-functional features in the original Vision Document are described in specific sections for constraints, quality ranges, precedence and priority, other product requirements, and documentation requirements. To simplify the high level structure of the Vision Document, we propose two major subsections for the non-functional features: one for the visual representation (Section 7) and one for the textual (Section 8). These non-functional goals can subsequently be refined into the Special Requirements document as the software development progresses.

1) *Non-functional Product Features (visual)*

The new visual non-functional product feature section is intended to provide a place to describe the non-functional functional capabilities in an easy to read and understand, graphical representation. The Guidelines in this section are:

Each non- functional capability appearing as an icon in this section should appear in a subheading in the Non-functional Product Feature (textual) section.

The non- functional capabilities should have a priority ranking

To reduce complexity of the visual model, create subgraphs for substantial or complex decompositions

Describe why each non- functional capability is needed by the stakeholders (this is a claim on the visual model)

Describe why each non- functional capability is needed by the stakeholders (this is a claim on the visual model). To reduce the complexity of the visual model, consider describing claims only in the textual representation

The level of decomposition in the graph should be represented by the level of subheading used in the textual section. The level of decomposition should be consistent with the level of decomposition of the textual representation (i.e., second level subheading should have an icon in the second level of the graph in the visual section)

For example, in the QARAT system, part of the Non-functional Product Features (visual) for security is illustrated in Figure 4.

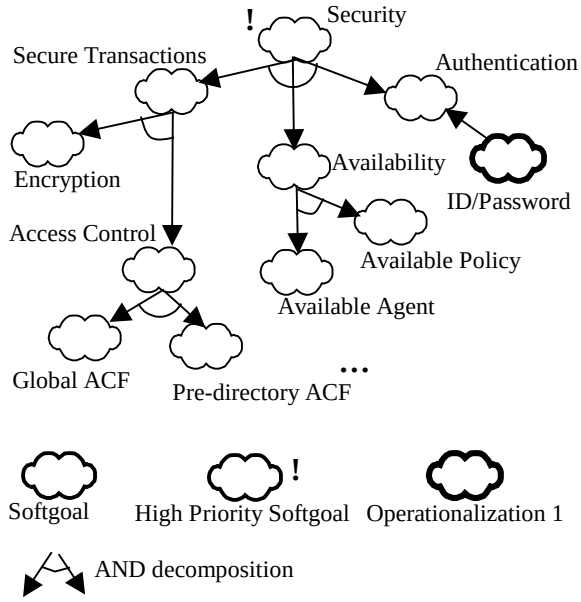

Figure 4 QARAT Non-functional Product Features

2) *Non-functional Product Features (textual)*

The textual non-functional product feature section is intended to provide a place to describe the non-functional capabilities in more detail. This can alleviate the need to

include everything on a visual model, while still providing a complete description. This allows the visual model to be easier to read and understand. New guidelines added into the textual section are:

Each non-functional capability listed in a subheading should appear as a softgoal, operationalization, or claim icon in the Non-functional Product Feature (visual) section.

The non-functional capabilities should have a priority ranking

When decomposed, the type of decomposition (and, or) should be described with the keyword "AND" or "OR"

The level of decomposition in the graph (visual section) should be represented by the level of subheading used

The level of decomposition should be consistent with the level of decomposition of the visual representation (i.e., second level subheading should have an icon in the second level of the graph in the visual section)

The very negative (break), negative (hurt), positive (help), and very positive (make) associations with other non-functional goals should appear. For example, Authentication hurts Response time performance)

For example, in the QARAT system, the following non-functional product features for security are described:

```
8.1 Security
It is a highly prioritized softgoal, to
ensure the overall security of the
system. Security is decomposed to
Secure Transaction, Availability and
Authentication. These are important
features to implement in QARAT.

8.1.1 Availability
Claim: Availability of agents and
policies is crucial to QARAT as the
system should be available to users all
the time.

8.1.2 Authentication
Claim: It is a high priority softgoal
since users cannot operate in the
system without proper authorization.

Operationalization: ID/Password is used
to authenticate user access to the
system and ensure security. User logs
in and out of the application.

Authentication hurts the softgoal
Usability.

8.1.3 Secure Transaction
…
```

IV. CONCLUSIONS AND FUTURE WORK

An Enhanced Vision Document is presented in this work, which provides visual models for the functional and non-functional goals of the system. The functional goals are presented with AND/OR Graphs; non-functional goals are presented with Softgoal Interdependency Graphs.

The Enhanced Vision Document is organized to capture the answers to a rich collection of *who*, *what*, *why*, and *how* questions, with intuitive visual representations in addition to the textual representations, which can provide more detail. We have observed the following limitations of the current Enhanced Vision Document:

- As a representation of early requirements, there is a need to extend this work to consider enterprise goals, to provide a business view.
- The relationships among the stakeholders are not clearly captured in a visual representation. An Agent-oriented approach is needed for this.
- The manual creation and revision of the visual representations is time-consuming and also error-prone. Strong tool support is essential to assist the Analyst.

There are several lines of future research. The set of guidelines presented in this paper for mapping the Vision Document into a goal model needs to be extended to create an Enhanced Vision Document that includes the visualization of enterprise goals as well as agents (in the spirit of i*), which would more thoroughly describe the interactions among the stakeholders. Subsequently, the approach then needs to be extended again in order to generate a use case model. Improvements to traceability techniques to support visual requirements models also need to be investigated. Tool support also needs to be improved to support the Enhanced Vision Document that can assist the Analyst to capture the goals and check them for consistency errors. This can be accomplished in part by refining the new guidelines into rules and encoding them. Catalogs of non-functional goals, represented in SIGs, can be provided to the Analyst for browsing, adopting, and integrating softgoal decompositions in whole or part into the Enhanced Vision Document for their system under development.

It will be interesting to see if these research activities would result in a new paradigm in which early requirements engineering starts, evolves, and ends with visual modeling.

V. REFERENCES

[1] Chung, L., *Representing and Using Non-functional Requirements: A Process-Oriented Approach*, DKBS-TR-93-1, June 1993, Department of Computer Science, The University of Toronto, Canada.

[2] Chung, Cooper, K. and Courtney, S., *RUP Vision Document for the Home Appliance Control System: Defining Stakeholders, Goals, and COTS Components*, Technical Report UTDCS-17-04, 2004, Department of Computer Science, The University of Texas at Dallas, USA.

[3] Chung, L., Nixon, B., Yu, E., and Mylopoulos, J., *Non-Functional Requirements in Software Engineering*, Kluwer Academic Publishers, 2000.

[4] Dardenne, A., van Lamsweerde, A., and Fickas, S., "Goal-Directed Requirements Acquisition", *Science of Computer Programmin*g, Vol. 20, 1993, pp. 3-50.

[5] Kroll, P. and Krutchten, P., *The Rational Unified Process Made Easy*, Addison-Wesley, USA, 2003.

[6] Cysneiros, L. M. and Julio Leite, J. C. S. P., Nonfunctional Requirements: From Elicitation to Conceptual Models. *IEEE Trans. Software Eng*. 30(5): 328-350 (2004)

[7] Dardenne, A., van Lamsweerde, A., and Fickas, S., "Goal-Directed Requirements Acquisition", *Science of Computer Programmin*g, Vol. 20, 1993, pp. 3-50.

[8] Nilsson, N.J., *Problem Solving Methods in Artificial Intelligence*, McGraw Hill, USA, 1971.

[9] Rational Unified Process Workbench, available at http://www-306.ibm.com/software/rational

[10] Rolland, C., Grosz, C. and Kla, R., Experience with Goal-Scenario Coupling in Requirements Engineering. *Requirements Engineering*, 1999: 74-

[11] Simon, H., *The Science of the Artificial*, Second edition, Cambridge, MA., The MIT Press, 1981.

[12] Darimont, R., Delor, E., Massonet, P. and van Lamsweerde, A., GRAIL/KAOS: An Environment for Goal-Driven Requirements Engineering, *ICSE* 1997: 612-613

[13] Yu, E. *Modelling Strategic Relationships For Process Reengineering*, DKBS-TR-94-6, December 1994, The University of Toronto, Canada

[14] van Lamsweerde, A., "Goal-oriented requirements engineering: a guided tour", Proceedings of the 5th Int. Symposium on Requirements Engineering, 2001, pp. 249 –262.

APPENDIX A. ENHANCED VISION DOCUMENT: TABLE OF CONTENTS

A simplified Table of Contents for the Enhanced Vision Document is presented here. Elements from the original Vision Document are in black font; the new and modified sections for the enhanced version are in blue font. For example, new sections have been added for the visual representations of the functional and non-functional goals (Sections 5 and 7). Section 4.2 has been modified with additional Guidelines to improve consistency between the representations. These guidelines are being implemented as rules in the tool support under development, which will automate this aspect of consistency checking within the Enhanced Vision Document.

TABLE OF CONTENTS