A Commitment Reasoning Support System: CRSS

By

Lik Alaric Lau
B.S., The Pennsylvania State University, University Park, 1990

A THESIS SUBMITTED IN THE PARTIAL FULFILMENT OF
THE REQUIREMENTS FOR THE DEGREE OF
MASTER OF SCIENCE

in

THE FACULTY OF GRADUATE STUDIES
MANAGEMENT INFORMATION SYSTEMS
FACULTY OF COMMERCE AND
BUSINESS ADMINISTRATION

We accept this thesis as conforming
to the required standard

THE UNIVERSITY OF BRITISH COLUMBIA

December, 1993

© Lik Alaric Lau, 1993
In presenting this thesis in partial fulfilment of the requirements for an advanced degree at the University of British Columbia, I agree that the Library shall make it freely available for reference and study. I further agree that permission for extensive copying of this thesis for scholarly purposes may be granted by the head of my department or by his or her representatives. It is understood that copying or publication of this thesis for financial gain shall not be allowed without my written permission.

(Signature)

Department of  

The University of British Columbia  
Vancouver, Canada  

Date  

12-15-93
Abstract

The concept of commitment has recently been adopted by researchers to model cooperative organizational activities. A commitment is an agreement by an agent to a requester to carry out a requested action. A Commitment Reasoning Support System (CRSS) is proposed in an attempt to support the reasoning process in deciding whether or not to make a commitment. An implementation of CRSS is constructed on the NeXT Computer System. An empirical study was conducted on the CRSS implementation. The findings show that the system is especially useful for commitment requests with high complexity and/or with important implications. The study also provides important conceptual insights in commitment reasoning for future research.
# Contents

Abstract ......................................................... ii  

Table of Contents ............................................... iii  

List of Figures ................................................... vi  

1 Introduction .................................................... 1  

2 Concepts of Commitment ........................................ 3  

2.1 The Literal Meaning ........................................... 3  

2.2 Commitment from Sociology and Social Psychology ........... 3  

2.2.1 The Side Bet/'Have to’ Notion .............................. 3  

2.2.2 The Sentiment/Flow/'Want to’ Notion ......................... 4  

2.2.3 The Connection between the Two Notions .................... 5  

2.2.4 The Positive Elements, Negative Elements, and the Links ... 5  

2.2.5 Value Systems and Beliefs ................................ 6  

2.3 The Relationships of Decisions and Commitments ............. 6  

2.4 Related Work .................................................. 8  

2.4.1 Distributed Artificial Intelligence .......................... 8  

2.4.2 Multi-Criteria Decision Making Support Systems .......... 9  

2.4.3 Indented Text Issued-Based Information Systems .......... 11  

2.4.4 Contract Theory and Reasoned-Action Theory ............... 12
2.4.5 The Request-Commitment Support Model ...................... 12
2.5 Definition of Commitment in this thesis ......................... 13

3 Factors Related to Commitment Reasoning .......................... 15
  3.1 Resources ............................................................. 15
  3.2 Utility ................................................................. 16
  3.3 Constraints ............................................................ 17
  3.4 Sentiments and Internal Values ................................... 17
  3.5 Connections between Commitments ............................... 18

4 The Reasoning Process for Commitment .............................. 20
  4.1 Reasoning and Logic ................................................ 20
  4.2 Commitment Reasoning as a default reasoning process .......... 21
  4.3 Theorist ............................................................... 23
  4.4 Preferred Subtheories .............................................. 25
    4.4.1 Concepts ....................................................... 25
    4.4.2 The Formal Framework ....................................... 26

5 The Commitment Reasoning Support System ......................... 28
  5.1 Theoretical Foundation ............................................ 28
    5.1.1 Side Bet Analysis ............................................. 30
    5.1.2 Sentimental Factors Analysis ................................ 31
  5.2 System Design ...................................................... 32
    5.2.1 The Main Menu ................................................ 34
    5.2.2 The Side Bet Analysis ........................................ 35
    5.2.3 Sentimental Factors Analysis ................................ 40
    5.2.4 Casebase ....................................................... 42
  5.3 Implementation ..................................................... 44
List of Figures

5.1 Brickman’s Model of Commitment ............................................. 29
5.2 CRSS System Design ................................................................. 33
5.3 CRSS Main Menu ................................................................. 34
5.4 CRSS Side Bet Analysis ............................................................ 36
5.5 CRSS Rule Inspector ................................................................. 37
5.6 The Conceptual Model of the Core Rulebase Used in the Empirical Study . 39
5.7 CRSS Sentimental Factor Analysis ............................................ 41
5.8 CRSS Casebase ................................................................. 43

6.1 Flowchart of the Firing of Rules ................................................. 47
6.2 CRSS Final Decision Screen .................................................. 49
6.3 Purchasing Request ............................................................... 50
6.4 Side Bet Analysis of the Purchasing Request ............................... 51
6.5 Final Decision of the Purchasing Request .................................. 52
Chapter 1

Introduction

The concept of commitment was first widely studied and developed in social psychology and sociology fields. The essential idea from these domains is that a commitment is a bond or a constraint on a person's courses of actions. Recently, the social approach to modelling organizational activities has been studied by distributed artificial intelligence (DAI) researchers [Gasser91] [Hewitt91] and organizational information systems (OIS) researchers [Bond90] [Hewitt86]. In particular, the concept of commitment is incorporated by researchers to model the cooperative activities in organizations. For example, it has been suggested that an organization can be modelled as a network of intelligent agents with mutual commitments [Bond90].

In this thesis, a specific form of cooperative activities are modelled by the concept of commitment: an agreement by an agent\(^1\) to the requester to carry out the requested action [LuWoo92]. In order to support the reasoning process for deciding whether or not to make a commitment request, relevant concepts of commitments from different research areas (artificial intelligence, social psychology and sociology) are incorporated to build a reasoning support system: The Commitment Reasoning Support System (CRSS).

The research reported in this thesis attempts to (1) show the possibility and the applicability of providing computer support in the reasoning process of making commitments,

\(^1\)an agent is an organization worker in this context
(2) to gain insights on the use of such a support, and (3) to find out the applicability of default reasoning to support this type of reasoning process. To attain these objectives, an implementation of CRSS is purposed and an empirical study has been conducted on the implementation.

In the next chapter, the concepts of commitment in different areas of study are explored. The factors related to commitment reasoning process are discussed in Chapter 3. Chapter 4 illustrates how the commitment decision making reasoning process can be modelled by a default reasoning framework. The theoretical foundation, the system design, and the implementation of the CRSS are discussed in Chapter 5. Chapter 6 shows two examples to illustrate the functions of the CRSS implementation, followed by a discussion of the empirical study of CRSS in Chapter 7. Finally, this thesis ends with 'Conclusion and Future Work' in Chapter 8.
Chapter 2

Concepts of Commitment

2.1 The Literal Meaning

In the Webster Dictionary a commitment means 'an agreement or pledge to do something in the future'. According to [Bond90], it is originated from the Latin word *committere*, which conveys the idea of an action to unite individual entities into one whole.

2.2 Commitment from Sociology and Social Psychology

2.2.1 The Side Bet/'Have to' Notion

The concept of commitment was first studied in the field of sociology by Howard Becker [Becker60] [Bond90]. He suggested that commitments were determined by side bets. That is, when an individual participates in multiple settings, his\(^1\) commitment to a course of action in a particular setting is determined by the side bets resulted from his simultaneous participation in other settings (or activities) [Becker60].

A side bet could be any item *that the person has linked to a course of action that was initially related to that item; that the person will lose if he does not follow the course of action in the future; that he has chosen by prior actions to make the linkage (intentionally

\(^1\)For ease of reading, the masculine form of pronoun is used to refer to both males and females.
or unintentionally); and that the person is aware of having made the linkage (when he is considering derailing from the course of action)' (p.33 [Brickman77]). For instance, a person is considering switching from his current cellular phone company to a new one (for which his phone number must be changed). Even though the new company provides better services and greater convenience, he might decide not to switch due to the side bets he has made with his current phone number, namely having the number listed in the Yellow Pages and all his business cards. This person is said to be committed to staying with his current phone company.

Other researchers have also defined 'commitment' in similar ways. In [Gerard67], a commitment is defined as 'any constraints that operate against changing behavior'. While in [Kiesle71], a commitment means 'the pledging or binding of the individual to behavioral acts.' [Brickman77] points out that the term 'commitment' is used sometimes by people to refer to a course of action they 'have to' follow, or else they stand to lose (or not to gain) something valuable to them.

To summarize, the Side Bet or 'Have To' notion of commitments is determined purely by utilitarian considerations [JanMan77]. Utilitarian considerations can be of two types: (1) utilitarian considerations for self, and (2) utilitarian considerations for significant others (for instance, family members, spouse, friends, co-workers, etc.) [JanMan77].

2.2.2 The Sentiment/Flow/‘Want to’ Notion

Another notion of commitment is described as a course of action that a person ‘wants to’ follow [Brickman77]. This type of commitments are purely determined by the internal values (or sentimental factors) assigned to them. [Csikszen75] describes this kind of commitment as a flow, which is an activity that has a person’s total attention, and is self-motivating, appearing to need no external rewards. For example, there are people who commit themselves to volunteer work on weekends which has no apparent utilitarian reward. It is their conscience or other internal values that motivate them for such commitments.

[JanMan77] explains that the actual driving forces behind commitments of this type are:
(1) self-approval considerations (e.g., personal ideals, ethical and moral considerations, and self-images), and (2) social approval considerations (e.g., image to and opinion from the peers, close friends, and family). These considerations may not take as tangible forms as the utilitarian considerations, but depends on the nature of commitment, they sometimes have more dominant influence than utilitarian ones [JanMan77].

2.2.3 The Connection between the Two Notions

These two notions seem to suggest contradictory meanings to the term 'commitment'. The Side Bet (or 'have to') notion has a utilitarian perspective on the concept of commitment, while the Flow (or 'want to') notion has an intrinsic perspective. However, they both suggest a common underlying characteristic of commitment: a binding force which resists temptations and negative factors, to bind a person to a course of action. From this perspective, these two notions are differed only by their binding forces. The Side Bet notion of commitment by [Becker60] is shaped by external values: money, time, material, skill, etc. While the Flow notion is motivated by internal values: personal ideals, self-image, moral considerations, etc.

Furthermore, [Brickman77] points out that in reality commitments usually do not come in extreme forms as the above examples, they are usually somewhere between the continuum of the 'want to' (the Flow notion) and 'have to' (the Side Bet notion). That is, most instances of commitments are driven by both external (utilitarian considerations for self and significant others) and internal (self- and social approvals) forces.

2.2.4 The Positive Elements, Negative Elements, and the Links

In [Brickman77], a commitment is said to contain three sets of elements: positive elements, negative elements, and the links between the two. In other words, for every commitment, there are some elements which attract the agent to make the commitment, and there are some elements that repel him from making it; and the agent's final decision depends on how he evaluates the values of the pros and cons of the commitment (i.e., how he links up the relative importance of the two sets of elements).
This view is also endorsed by many other social scientists [Lewis51] [JanMan77]. Basically, they suggest that a person would only commit to a course of action if he links more\textsuperscript{2} positive factors than negative factors to it.

2.2.5 Value Systems and Beliefs

In the process of making a decision to commit to some course of action, we implicitly assign intrinsic values to that action. The value system of a person can be viewed as a set of beliefs about the world. The same action may have entirely different meanings to two people who hold different beliefs about them. For example, if there is an invitation to attend a social function which conflicts the class schedule; one person might decide to accept the invitation because he believes that socializing opportunities and interpersonal skills are more important than academic knowledge; while another person might refuse to go because he believes that it is 'wrong' to skip class for any reasons other than emergencies. [Becker60] states that the value system of a person is largely determined by his family, educational, and social backgrounds.

Hence, how an agent makes commitments is affected by his belief or value system [Becker60] [Bond90] [Brickman77]. From this viewpoint, a commitment reflects how an agent's value system links the external actions with his internal values.

2.3 The Relationships of Decisions and Commitments

The process of decision making is the selection of an act or a course of action from among alternative acts or courses of actions such that it will produce optimal results under some criteria of optimization [JanMan77].

There are three characteristics that distinguish a system as a DSS [BonchHolWhin81]:

1. a model is incorporated in the system.

2. it provides useful information to higher-level management aiming at supporting relatively unstructured decision making situations.

\textsuperscript{2} 'more' can be in qualitative terms as well as quantitative terms.
3. It provides powerful and user-friendly interfacing for problem solving.

In summary, a DSS is a system that provides a user-friendly interface to support the semi-structured decision making processes done by higher-level management.

Recall the meaning of commitment from the Webster's Dictionary: 'an agreement or pledge to do something in the future'. There are very intimate relationships between the concept of commitment and most psychological formulations of the decision-making process [JanMan77]. [Lewis51] suggests that a commitment has the effect of 'freezing' a decision (i.e., commitment is what makes the decision maker sticks with his decision). The concept of commitment affects a decision maker both before and after the decision is made [JanMan77]—before making the decision, the decision maker often considers the consequences of being bound by each alternative acts; while after the decision is made, the decision maker feels obligated to carry out his decision.

In essence, a commitment is a concept that binds a decision maker to his decision. From the definition of a decision making process, a commitment reasoning process can be defined as a decision making process for which the agent selects either to commit or not to commit to a requested action such that it will produce optimal results. The Commitment Reasoning Support System can, therefore, be viewed as a specialization of decision support system with the following characteristics:

- It focuses on handling decision making situations at a personal level, namely situations when the user needs to decide whether or not to make the requested commitment; while much of the DSS research effort is devoted to situations at the corporate level.

- It aims at supporting the making of decisions that are meant to be carried out and followed through. That is, the emphasis is on the accountability of decision making (see Section 5.2.4). The system enhances such accountability by: (1) alerting the agent about the consequence of his decision before he actually decides, (2) providing support for the user to understand his rationale for making the decision, and (3) by keeping records of his decisions and decision rationale in case when they are questioned in the future.
• It attempts to address the sentimental aspect of personal decision making. These sentimental factors (e.g., considerations of self- and social approvals) play important roles on decision making process [JanMan77] but are not emphasized by decision making models.

Based on the characteristics mentioned above, the system proposed in here is called Commitment Reasoning Support System (CRSS) instead of a decision support system (DSS).

2.4 Related Work

2.4.1 Distributed Artificial Intelligence

In the current state of Distributed Artificial Intelligence (DAI) research, researchers are striving to adopt a social approach as the foundation of DAI [Gasser91]. It is believed that properties in a multiple agent system cannot be characterized only by the properties of individual agents, but must be characterized through the social interactions of these agents such as trials of strength, commitments, negotiation, and cooperation [Hewitt91]. [Bond90] views the interactions between agents as a network of commitments — agents make mutual commitments to establish the functionality of the settings they are in. For instance, in order for an agent to calculate the annual budget correctly, he has to assume that other agents have made the commitment to supply accurate information, and in turn when these agents receive the budget report later, they will assume that the agent who calculated the budget has used their information accordingly to produce the budget, and that they can rely on it to make further commitments. [Bond90] also suggests that agents can be characterized by the types of commitments made, the kind of resources used, the set of agents committed to, and the set of beliefs held.

These research projects provide valuable insights on how to model organizational activities with the concept of commitment, which are incorporated into our work (e.g., the utility of resources, the inter-connection between commitments, the interaction between value systems and commitments). However, little work has been done on supporting the decision making...
process in making a commitment within organizations.

2.4.2 Multi-Criteria Decision Making Support Systems

A Multi-Criteria Decision Making (MCDM) problem is a decision making problem with at least two conflicting criteria and at least two alternative solutions [Tabucanon88].

Criteria are measures, rules, and standards that guide decision making. All attributes, objectives, or goals, which have been judged relevant in a given decision making situation are criteria. Criteria are said to be in conflict if the full satisfaction of one will result in impairing or precluding the full satisfaction of the others [Tabucanon88]. Attributes are characteristics used to describe a thing's: (1) objective traits such as profit, loss, production volume, etc., and (2) subjective traits such as prestige, goodwill, beauty, etc. Objectives are aspirations that also indicate directions of improvement of selected attributes such as maximize profit, minimize losses, etc. And Goals are aspirations with given levels of attributes such as to increase profit by 25% [Tabucanon88].

The commitment reasoning process can be considered as a Multi-Criteria Decision Making process because:

• The positive and negative factors present conflicting criteria of the process. If we view criteria as goals to be met, the positive factors are goals projected to be met if committed, and negative factors are goals projected not to be met if committed. These two sets of criteria are in conflict since presumably it is impossible for the agent to benefit from the positive factors while avoiding the negative factors if he commits to the requested action.

• To commit and not to commit are the two alternative solutions.

However, the current research of MCDM differs from our research of commitment reasoning in the following aspects:

• Current MCDM research focuses on solving structured decision making situations such as production planning, allocation decisions, project selections, etc.
Commitment reasoning emphasizes on unstructured decision making situations (i.e., situations which an agent needs to decide whether or not to commit to a requested action). The unstructuredness comes from the sentimental considerations (such as friendship, reputation, and other self-image factors) in these decision making situations. It is believed that sentimental factors place a crucial part in commitment reasoning [Brickman77] [Becker60] [JanMan77] [Lu92].

- Situations considered by MCDM researchers usually have a large set of alternative solutions with objective criteria [Tabucanon88], while situations in commitment reasoning have a mixture of objective and subjective criteria and a set of two alternative solutions: commit or not commit.

- MCDM research emphasizes on building mathematical models to support decision making at a corporate level. Commitment reasoning research, on the other hand, focuses on supporting decision making process at an individual level. The subjectivity and the unstructuredness of the factors involved in the process at this level render the use of those mathematical models impractical.

Despite the different focuses of MCDM and commitment reasoning research, the following insights for commitment reasoning have been gained from the MCDM research:

- It can be viewed as a MCDM process.

- There are conflicts between criteria in a commitment reasoning process.

- Ranking the importance of the criteria is an important first step in solving a MCDM problem.

In response to these findings, the CRSS rulebase design includes a built-in ranking feature to handle the conflicting criteria (see Section 5.1.1).
2.4.3 Indented Text Issued-Based Information Systems

A different but related piece of work was done in [Conklin91]. A system called ‘indented text Issued-Based Information System’ (itIBIS) is described. The idea behind this project is to formalize the documentation of design rationale (DR) (which is defined as ‘the information that explains why an artifact is structured the way that it is and has the behavior that it has’ (p.359 [Conklin91])). Essentially, the system formalize the documentation by allowing the users to state the design problem, all possible solutions, the final decision, and the arguments for adopting the final decisions in a specified text notation.

The field trial of itIBIS shows that it enhances the quality of design decisions by:

- allowing designers to easily refer to DR of previous related design problems recorded in itIBIS.
- allowing team members to clearly understand other members’ positions and arguments which are recorded in itIBIS during the group meeting.
- allowing the designers to more thoroughly consider the factors that are relevant to the design problems. This benefit is achieved by requiring users to explicitly state the design problem, all possible solutions, the final decision, and the arguments for adopting the final decisions. By doing so, a designer can see a clearer picture of the entire design problem, and thus can search his memory for salient information which can lead to a high quality design decision.

Similar decision support facilities are featured in the Decisional Balance Sheet in [JanMan77] and in the Decision Construction Set in [HelMarWoo]. This piece of research is discussed here because we adapted the design rationale idea from itIBIS in our Commitment Reasoning Support System. Detail usage of this work and how it benefits commitment reasoning are discussed in Section 5.1.2, 5.2.3, and 5.2.4.
2.4.4 Contract Theory and Reasoned-Action Theory

The concept of contract is widely studied in law, economics, and sociology [Macneil80] [Lu92]. [Lu92] states that the essential difference between a contract and a commitment is that a contract is usually perceived as having an emphasis on economic exchanges, while a commitment emphasizes more sentimental factors.

Reasoned-Action Theory [FisAjz75] claims that a person's intention to a behavior is determined by his:

1. attitude to the behavior (i.e., the expected behavior consequences and the evaluations of these consequences)
2. subjective norm (i.e., the person's perception of what others think of the behavior)

[Lu92] provides a detailed analysis of the relationship between Reasoned-Action Theory and commitment. Conceptually, if we view making (or not making) a commitment as a behavior, the Reasoned-Action Theory does provide a model for commitment reasoning. However, Reasoned-Action Theory does not provide enough details about commitment for it be useful for our work.

2.4.5 The Request-Commitment Support Model

In [Lu92], a commitment reasoning support system called RECOS (REquest-COmmitment Support model) was proposed. Many of our concepts in this model are adopted from and inspired by [Lu92]. Both the RECOS and the CRSS aim at achieving one goal: to support the reasoning process conducted by an agent in order to decide whether or not to make a commitment. However, the CRSS differs from the RECOS model in the following main ways:

- The RECOS project emphasizes surveying the various studies on the concept of commitment.

The CRSS project utilized the knowledge gained from RECOS to construct an implementation to support the decision making process of commitment making.
• RECOS provides invaluable insights and analysis of the decision making process of commitment making at a conceptual level.

CRSS aims at acquiring insights of the decision making process of commitment making at an implementation level (which may not be found on the conceptual level).

• RECOS provides a general framework for supporting the decision making process of commitment making.

The CRSS project used the empirical study results from RECOS and additional research findings to construct a more precise commitment reasoning framework (based on Brickman's commitment model [Brickman77]).

• Even though the RECOS project suggests a framework which provides precious ideas on the design of such a system, it merely suggests 'what to build'.

The CRSS project addresses many of the implementation issues by providing the solutions for 'how to build'. In particular, much research was done in coming up with a rulebase design (which adopts features from a default reasoning framework) and a set of domain-independent rules for the system (a cole rulebase).

2.5 Definition of Commitment in this thesis

Recall that the main purpose of this project is to model organizational activities in a way that effective computer support can be provided for them. In this section, a precise definition of commitment is given to further clarify the focus of this project:

Definition 2.5.1 A commitment is an agreement by an agent to a requester to carry out a requested action.

In an organization, workers very often agree or promise to carry out tasks that are requested by other workers. For instance, a programmer promises to write a program requested by his supervisor, the purchasing department agrees to order new softwares requested by other
departments, a worker assures to carry out a sick colleague’s work by his request, etc. All these agreements, promises, assurance, and guarantees, according to the above definition, are considered as commitments.

Anyone who has worked in an organization knows that this type of commitment requests occur regularly at different levels of complexity and importance. Every time an agent receives a request (which may appear in hand-written memo form, e-mail form, verbal form, or any other form), he has to go through a reasoning process in order to decide whether to commit to the request. The reasoning process is defined as followed:

**Definition 2.5.2** A commitment reasoning process is the reasoning process conducted by an agent in order to decide whether or not to make a commitment.

The vision of this project is to have a computer system in the future which can support the commitment reasoning process to an extent that (1) some routine requests, such as requests for a weekly meeting or requests issued by certain people, are handled by the system automatically based on knowledge in the rulebase, and that (2) reasoning support is provided for the agent when uncommon or important requests are received. A Commitment Reasoning Support System is proposed to realize this vision.
Chapter 3

Factors Related to Commitment Reasoning

After the discussion of the different concepts of commitment in Chapter 2, one can perceive that a commitment reasoning process is a process which takes various types of factors into consideration. In this chapter, an effort is made to identify these factors.

3.1 Resources

The relationship between resources and commitments has been discussed in various works [Becker60] [Bond90] [Gerson76] [Gasser91] [Hewitt91]. In particular, [Gerson76] expresses commitments as constraints on the use of resources. Similarly, in [Gasser91], commitments are treated as the use of resources. In some literature, these resources include not only time, money, CPU time, etc., but also intangibles such as sentiments, knowledge, opportunities, etc. [Bond90].

In [Gerson76] [Hewitt91] [Lu92], resources are money, time/space/material, skill/technology, authority, and sentiment. Money, time/space/material, and skill/technology are more intuitively perceived as resources in organizations because:

1. their direct involvement in commitments
2. they are quantifiable and/or measurable entities

Authority, in this thesis, refers to the position an agent is holding in the organization (e.g., general manager, administrator, etc.). It can be viewed as a resource because when an agent issues a commitment request, the requestor’s authority (relative to the requestee’s authority) can be used as a resource that influences the outcome of the request. [JanMan77] refers to this type of resource factors as utilitarian factors.

Sentimental factors such as favors, competence, friendship, reputation, and other self-image factors are treated under a separate heading—‘Sentiments and Internal Values’ (in Section 3.4) due to the following reasons:

1. they are not as suitable to be quantified as the tangible factors (e.g., it is hard or even impractical to quantify how much friendship is needed to ask an agent to commit).

2. they play a distinct and important part in the commitment reasoning process [Brickman77] [JanMan77] [Lu92] (also refer to Section 2.2).

3.2 Utility

In economics context, utility stands for subjective pleasure, usefulness, or satisfaction derived from consuming goods [Samuel88]. In making a commitment, there is always an exchange of utility during the process, namely, an agent decides to make a commitment because he believes that he will gain more than he will lose after satisfying the commitment [ThibKell59][LuWoo91]. In [Brickman77], it is claimed that it is how the agent links the positive factors and the negative factors of a commitment that determines whether he will take on or reject the commitment.

The factor of utility is closely related to resources, in that it deals with the agent’s beliefs about the gains and losses of resources in making certain commitment. However, utility is more than merely the difference between the benefit and the cost of an action, utility of an action is an aggregate measure of all dimensions of worth such as cost, benefit, risk and other
properties related to the agent's situation [Doyle92][ThibKell59]. In our Commitment Reasoning Support System, resource utility (e.g., gain/loss of time/space/material, skill/technology) is dealt with by means of rules in the CRSS rulebase. Agents can set up different rules of utility functions for different commitment types. As for sentimental and internal values' utility (i.e., utility related to self-image of helpfulness, loyalty, competence, etc.), users are presented with all the relevant factors in a table form, so that they can analyze factors which may be affected if they were to make or not to make certain commitments.

3.3 Constraints

Constraints are rules and regulations that an agent is bound by in making commitments. Hence, constraints include corporate regulations and other written laws. [JanMan77] explains that constraints could also be social (e.g., approval from peers and family). In Section 3.5, we will see how some connections between commitments can be viewed as constraints as well.

3.4 Sentiments and Internal Values

According to [Hewitt91], 'sentiment deals with how various participants feel about each other: goodwill, reputation, obligation, etc.' In other words, sentiment deals with the beliefs of an agent on things such as self-images, images viewed by other agents, images of commitment, and the intrinsic values of behaviors.

According to [Brickman77], a commitment is a means to link people's internal beliefs to external actions. That is, when people carry out certain actions, they must have sufficient positive internal values attached to those actions to justify the negative features of the actions. Although CRSS has a more rigid definition of commitment, it is certain that our value systems play an important role in the commitment reasoning in our context. For instance, if Agent A receives a request to do a task for Agent B, when Agent A is deciding whether to commit, he will consider his relationship with Agent B, the implication of this commitment to his career, his self-image, and his perceived image seen by other agents. All these are linked to Agent
A's value system in one way or another. If Agent A has a belief that helping a friend is very important, and he thinks Agent B is a good friend, then he is likely to commit. While if he believes that his career is most important, then he will only commit when he thinks it will not interrupt with his career pursuit.

[JanMan77] interprets sentimental factors as self-approval and social approval. Self-approval is how a person interprets the effects of his actions on his morality, personal ideals, and other self-image issues. Social approval is how a person interprets the effects of his actions on how the people around him view him. [JanMan77] also agrees that these sentimental considerations have significant impacts on a person's commitment reasoning process.

Section 2.2 provides a detailed discussion on the sentimental aspect of commitment.

3.5 Connections between Commitments

In [Becker60], an agent’s commitments are constrained and determined by side bets (i.e., the agent’s activities or commitments in other settings). In [Gasser91], commitment is also seen a result of an agent’s participation in multiple settings. [Bond90] states that an organization can be viewed as a 'web of commitments'. Despite the different contexts of the citings, they all agree that commitments are interconnected and are closely related to each other. [Lu92] states that there are direct (a natural and predefined relationship between commitments) and indirect (interlinked by resources) connections. Regardless of the type of connections, it is intuitive that when an agent is considering a commitment request, its connections with his other commitments are a factor that he will not ignore during his commitment reasoning process. For instance, when an organizational worker is asked to relocate to a foreign branch, it is highly unlikely that he does not consider this request's implications on (1) his commitments in current social settings (such as taking care of his parents) and (2) on the future commitments (such as selling part of his properties, not seeing his friends and relatives for sometime, adapting to a new culture, etc.)

At this stage, the CRSS does not handle commitment connections directly. We think that handling commitment connections requires some extensive research in areas like conflict
resolution, and consistency maintenance, which are beyond the scope of our area of the current project.

Nevertheless, connections between commitments can be indirectly handled by providing rules on resource constraints and regulation constraints in the CRSS rulebase. For example, the user can specify constraint rules such as *if commitment type A is made, then commitment type B cannot be made*. Indirect connections can be handled partially through some resource management rules (e.g., if available resources are less than required resources, then agent cannot commit).
Chapter 4

The Reasoning Process for Commitment

In order to incorporate the concept of commitment into organization information systems, the representation of commitment must be dealt with in a formalized manner. [Bond90] points out three representational issues of commitments: (1) expression of commitments (how to represent an entity called a commitment), (2) generation of commitments (i.e., how one decides to make a commitment), and (3) the simple negotiation of commitments (how to decide a reasonable set of commitments for agents to make under given circumstances). In this project, the focus is on the first and second representational issues — the expression and the generation of commitments.

4.1 Reasoning and Logic

The most common way to model human beings' reasoning process is through the use of logic. There are more than one way to characterize what logic is. From the artificial intelligence point of view, logic is about the study of knowledge representation languages in which the notion of entailment can be captured [Ramsay88]. In other words, it is about the study of languages such that when facts and rules are stated, there is a means to infer new facts.
The problem with classical First Order Logic (FOL) for modelling the human reasoning process is its monotonicity. That is, as the number of premises increases, the number of conclusions drawn cannot decrease. For every set of premises \( A \) and formulas \( p, q \):

\[
A \vdash q \Rightarrow A \cup \{p\} \vdash q
\]

This means new additional (and possibly more reliable) information can never invalidate old conclusions [Brewka91]. This posts a serious problem, since [Brewka91] points out that the two most crucial reasoning capabilities every intelligent agent should have are:

- the ability to handle incomplete information in a reasonable way (i.e., to draw sensible conclusions in the absence of complete information).
- the ability to handle inconsistent information.

For these reasons, we must rely on features of a logic that is nonmonotonic to model the process of commitment reasoning.

### 4.2 Commitment Reasoning as a default reasoning process

A reasoning process is said to be nonmonotonic when we draw conclusions based on incomplete information of the world, and upon arrival of more information, we are ready to retract some of the conclusions made previously. In the context of commitment reasoning, when an agent conducts his reasoning process in an open system environment, where information is incomplete and new information may emerge at anytime into the system, an agent must: (1) make default conclusions when information needed to draw definite conclusions is incomplete, and (2) must be able to retract some conclusions when more accurate information has arrived. For example, an agent may receive a request from a co-worker, who is an old friend of the agent, to do certain task for him. The agent may at first decide to help him based on their friendship and his availability of resources, but the agent may later retract this decision after realizing that he will be tied up with other more important commitments.
Default reasoning is a particular type of nonmonotonic reasoning \[\text{[Brewka90]}\] in that default assumptions are made in the presence of incomplete information. Using the popular example of flying birds \[\text{[Reiter87]}\], we want to represent that:

- 'birds normally fly' or
- 'if $x$ is a bird then by default assume $x$ flies'

so that we can use the rule to conclude Tweety flies when we only know Tweety is a bird. In other words, we would like to jump to conclusions in the absence of information to the contrary. Default reasoning is nonmonotonic, because 'inferences sanctioned by default are best viewed as beliefs which may well be modified or rejected by subsequent observations' (p.153 \[\text{[Reiter80]}\]).

The commitment reasoning can be viewed as a type of default reasoning process in that an agent makes certain default assumptions in the absence of complete information about a request. In open system environments such as organizations, complete and accurate information cannot be obtained at any particular point of time. Therefore, organizational workers must make their decisions based on certain assumptions. For instance, they may have a rule like:

\[
\text{If I gain resources more than I lose by making the commitment, then typically I commit}
\]

Hence knowing that a request will provide more gains than losses, the agent will make a default conclusion to commit based on the assumption that he has the required resources to do so. However, this conclusion is retractable (or conditional); that is, if the agent discovers that he actually does not have the required resources to commit, he will retract the conclusion and will decide not to commit.

Two formal frameworks for default reasoning - Poole's Theorist framework \[\text{[Poole88][Poole89]}\] and Brewka's Preferred Subtheories framework \[\text{[Brewka90][Brewka91]}\] will be examined. Part of the CRSS proposed in this thesis is based on the second framework by Brewka. A Descrip-
tion of Poole's Theorist framework (which Preferred Subtheories is extented from) is included here to assist the readers in understanding the rationales of the CRSS design.

4.3 Theorist

Theorist is actually the programming language in an implementation of a logical framework for default reasoning. For the sake of simplicity, we shall refer to this logical framework itself as Theorist from now on. The basic idea is that the system has a set of facts known to be true, and a set of possible hypotheses (or defaults). We want the system to be able to draw correct conclusions and not to imply things that are known to be false based on the set of facts and some subset of possible hypotheses that are consistent with the facts [Poole88].

\( \mathcal{F} \) denotes a set of closed formulas which we called 'facts'. Because facts are known to be true, they should be consistent with each other. For the set of defaults (or possible hypotheses) which is denoted by \( \Delta \) (\( \Delta \) is a set of formulas (possibly inconsistent)), the definitions of scenario and extension are given as followed [Poole88]:

**Definition 4.3.1** A scenario of \( \mathcal{F}, \Delta \) is a set of \( D \cup \mathcal{F} \) where \( D \) is a set of ground instances of elements of \( \Delta \) such that \( D \cup \mathcal{F} \) is consistent.

**Definition 4.3.2** An extension of \( \mathcal{F}, \Delta \) is a set of logical consequences of maximal (with respect to set inclusion) scenario of \( \mathcal{F}, \Delta \).

In other words, a scenario is the union of the set of facts with a subset of instantiated defaults that are consistent with the facts. An extension is a set of all the logical consequences that can be derived from the largest union found in some given sets of facts and defaults.

This framework captures the semantics of default reasoning by providing a set of defaults. They are assumptions of the world which we prepare to accept as being true as long as they do not conflict with the set of known facts. And we are ready to give up some defaults as soon as they become inconsistent with the current set of facts. This can be illustrated by the following example:
Example 4.3.1 Assume we have the following facts and defaults concerning flying animals:

\[ F = \{ \forall x \ bat(x) \rightarrow mammal(x), \forall x \ bat(x) \rightarrow fly(x), \ mammal(wayne) \} \]

\[ \Delta = \{ \ mammal(x) \rightarrow \neg fly(x) \} \]

With the given facts we can apply the default (in this case we have only one), and conclude that \{\neg fly(wayne)\}. However, if we later add \(bat(wayne)\) into the set of facts, then the default is inconsistent with the facts and now the system will conclude that \{fly(wayne)\}.

In [Poole88], an implementation called Theorist is proposed to capture this framework. It is a programming language that allows users to program the facts and defaults into the system, and it will generate the logical consequences entailed from the facts and defaults according to users' instructions.

The following is a description of the main Theorists syntax [Poole88]:

- **fact** \(w\), where \(w\) is a formula, means "\(\forall w\) \(\in F\)."

- **default** \(n\), where \(n\) is a name (predicate with only free variables as arguments) means \(n \in \Delta\).

- **default** \(n : w\), where \(w\) is a formula, and \(n\) is a name with the same free variables as \(w\), means that \(w\) is a default, with name \(n\). Formally this means that \(n \in \Delta\) and "\(\forall n \Rightarrow w\) \(\in F\)."

- **explain** \(g\), where \(g\) is a formula, asks whether we can explain \(\exists g\) from the \(F\) and \(\Delta\). A consistent explanation is returned.

Definition 4.3.3 If \(g\) is a closed formula, then an explanation of \(g\) from \(F\), \(\Delta\) is a scenario of \(F\), \(\Delta\) which implies \(g\).

Hence, when a user types in **explain** \(g\), he is asking the system if \(g\) is a logical consequence of any of the scenarios of \(F\), \(\Delta\), if so show him/her the defaults used in coming up with such consequence. The details of this system is not within the scope of our current interest. More explanation is provided in [Poole88] and [Poole89].

---

1\(\forall w\) is the universal closure of \(w\). That is, if \(w\) is a formula with free variables \(v\), then \(\forall w\) means \(\forall v w\).

Similarly \(\exists w\) is the existential closure of \(w\), that is \(\exists w\).
4.4 Preferred Subtheories

4.4.1 Concepts

The Preferred Subtheories framework is proposed by Brewka [Brewka90]. It is considered to be the extension of Poole's framework for default reasoning. In Poole's framework, it has a set of facts ($\mathcal{F}$), which are closed formulas that are consistent and are known to be true; and a set of defaults ($\Delta$), which are possible hypotheses. The way the system handles inconsistencies is to use the maximal consistent subsets of the formulas. More specifically, the framework will use the maximal subset of $\Delta \cup \mathcal{F}$, which is consistent, to entail logical consequences (i.e., an extension). However, there could be multiple maximal subsets which include different sets of formulas which entail inconsistent logical consequences. The following example illustrates the problem:

**Example 4.4.1** Consider the following facts and defaults of 'bird flies':

- **Fact**: `penguin(tweety)`
- **Default**: `penguin(X) → bird(X)`
- **Default**: `penguin(X) → ¬flies(X)`
- **Default**: `bird(X) → flies(X)`

We have two maximal subsets of consistent formulas, namely

- $(penguin(tweety), penguin(X) → bird(X), bird(X) → flies(X))$
- $(penguin(tweety), penguin(X) → bird(X), penguin(X) → ¬flies(X))$

The problem with Poole's framework is it does not provide preference of which maximal subset of formulas to use. In the above example, an intelligent system should choose the set of formulas that can conclude that *tweety doesn't fly*.

The Preferred Subtheories framework solves this problem by claiming that instead of two levels of reliability, which are the facts and the defaults, we could have multiple levels of
reliability. Therefore, when inconsistencies arise, the more reliable defaults are preferred over the less reliable ones. Using this method, Example 4.4.1 can be expressed as follows:

\[ \text{T1: } \text{penguin(tweety)} \]
\[ \text{T2: } \text{penguin}(X) \rightarrow \text{bird}(X) \]
\[ \text{T2: } \text{penguin}(X) \rightarrow \neg \text{flies}(X) \]
\[ \text{T3: } \text{bird}(X) \rightarrow \text{flies}(X) \]

With T1 being the most reliable level, this theory concludes that \( \text{bird(tweety)} \) and \( \neg \text{flies(tweety)} \).

Consider the above theory without: \( \text{penguin}(X) \rightarrow \neg \text{flies}(X) \). The theory would conclude \( \text{bird(tweety)} \) and \( \text{flies(tweety)} \). By inserting a more reliable rule \( \text{penguin}(X) \rightarrow \neg \text{flies}(X) \), we say that the system is able to retract the former conclusion \( \text{flies(tweety)} \) and draw a new conclusion: \( \neg \text{flies(tweety)} \).

4.4.2 The Formal Framework

Here are some formal definitions for the Preferred Subtheories Framework from [Brewka90]:

**Definition 4.4.1** A default theory \( T \) is a tuple \( (T_1, \ldots, T_n) \), where each \( T_i \) is a set of classical first order formulas.

For any two \( T_i \) and \( T_j \), such that \( i < j \), we say that information in \( T_i \) is more reliable than that of \( T_j \)'s. Although the definition of the default theory per se does not indicate any sort of leveling in reliability, the leveling feature will become clear as soon as the preferred subtheory is defined:

**Definition 4.4.2** Let \( T = (T_1, \ldots, T_n) \) be a default theory. \( S = S_1 \cup \ldots \cup S_n \) is a preferred subtheory of \( T \) iff for all \( k(1 \leq k \leq n) \) \( S_1 \cup \ldots \cup S_k \) is a maximal consistent subset of \( T_1 \cup \ldots \cup T_k \).

This definition says that to obtain a preferred subtheory, we start with \( T_1 \), get its maximal consistent subset, and then we go to \( T_2 \) and obtain maximal number of formulas from \( T_2 \).
which are consistent with the ones acquired in T1. Then we go on to the next level and so on until we arrive at Tn.

There are two notions of provability:

**Definition 4.4.3** A formula $p$ is weakly provable from $T$ iff there is a preferred subtheory $S$ of $T$ such that $S \vdash p$.

**Definition 4.4.4** A formula $p$ is strongly provable from $T$ iff for all preferred subtheories $S$ of $T$ such that $S \vdash p$.

The next chapter will describe how some of the features from the Preferred Subtheories framework are adopted in the CRSS rulebase.
Chapter 5

The Commitment Reasoning Support System

5.1 Theoretical Foundation

The Commitment Reasoning Support System (CRSS) proposed here is based on the concept of commitment described in [Brickman77] (Sections 2.2.1, 2.2.2, 2.2.3, and 2.2.4). Essentially, [Brickman77] says that a commitment consists of a ‘have to’ (utilitarian or side bet) aspect and a ‘want to’ (sentimental or flow) aspect, and each aspect has some positive factors and negative factors. The linkage between the positive and negative factors is made according to the value system of the person that makes the commitment. Our diagrammatic interpretation of this concept is given in Figure 5.1.

Brickman’s model is chosen because it points to the importance of the sentimental aspect of a commitment which is not emphasized by many researchers [JanMan77] [Brickman77] [Lu92]. Furthermore, Brickman’s model is a result of thorough analysis of the different studies conducted on the concept of commitment. By consolidating relevant findings from other studies with Brickman’s own insights, a more complete and precise model of commitment is resulted.

To operationalize Brickman’s model, CRSS consists of two subsystems to handle the
A Commitment

<table>
<thead>
<tr>
<th>Have To/Side Bet</th>
<th>Want To/Flow</th>
</tr>
</thead>
<tbody>
<tr>
<td>Positive Elements</td>
<td>Positive Elements</td>
</tr>
<tr>
<td>Linkage</td>
<td>Linkage</td>
</tr>
<tr>
<td>Negative Elements</td>
<td>Negative Elements</td>
</tr>
</tbody>
</table>

Figure 5.1: Brickman’s Model of Commitment

'have to' part and the 'want to' components of a commitment. Basically, CRSS assists users to determine the positive and negative aspects of both components of the commitment. The linkage of the 'have to' is determined by the rulebase reasoning in the Side Bet Analysis, while the linkage of the 'want to' is left for the user to determine by showing a table of sentimental factors. And upon making the final decision on whether or not to make the commitment, the user is required to write a short paragraph to explain and clarify his decision rationale (which corresponds to the linkage between the 'have to' and 'want to' parts of the commitment).
5.1.1 Side Bet Analysis

A Side Bet Analysis is provided in the CRSS to support the reasoning process concerning the utilitarian factors and rules [JanMan77] (i.e., the 'have to' or the side bet portion of a commitment). This subsystem uses rulebase reasoning to handle the utilitarian factors such as resource allocation (Section 3.1) and utility (Section 3.2), constraints (Section 3.3), and connections between commitments (Section 3.5). The Analysis generates a recommendation of whether or not to commit according to the rules of utilitarian factors (i.e., the recommendation is the operationalization of the linkage of the 'have to' portion).

The rulebase in the Side Bet Analysis adopts features from a default reasoning framework called Preferred Sutheories described in Chapter 4. The basic idea is to group the reasoning rules such as, if \[\text{not enough resource to commit}\] then \[\text{do not commit}\], in levels of priority, with the more important rules in higher levels of priority. Conclusions drawn from rules in higher priority override contradictory conclusions drawn from lower priority rules. This feature allows some former conclusions to be retracted if new rules, which refute those former conclusions, are inserted into higher levels.

The reasons for adopting such a framework for our CRSS rulebase are:

- In a monotonic rulebase system when new rules are entered, the system integrity must be maintained manually or by some dedicated consistency maintenance procedures [Buchanan83]. As the number of rules increases in a monotonic system, maintaining consistency could become very complicated (e.g., checking and modifying the existing rules in order to accommodate a new rule).

The nonmonotonicity of the proposed framework effectively decreases the chances of inconsistency (i.e., two or more rules drawing conflicting conclusions) by allowing users to put rules into multiple levels. By cranking the rulebase through the inference engine, users can find out what rules are fired and the levels they reside in. If conflicting rules need to be added, they can simply be put into a separate level without any modifications on the existing rules. In other words, the task of maintaining the consistency of the
rulebase is much simpler. This is highly suitable for an open system environment where new and possibly conflicting information and rules may emerge at anytime.

- It more naturally resembles the reasoning capabilities of an intelligent agent than a monotonic system. According to [Brewka91], an intelligent agent uses subsets of an entire set of possibly inconsistent knowledge to draw sensible conclusions in different situations. The prioritization of rules in this framework provides a natural mechanism to choose relevant subsets of rules to draw sensible conclusions, while in a monotonic system such mechanism is imbedded in the rules themselves (in the condition clauses of the rules). This means when new rules are required to reflect the emerging new information in a monotonic system some existing rules may need to be modified in order to generate the new subsets needed for sensible conclusions. While in the proposed framework, the new subsets are created by inserting new rules in the appropriate level.

- It allows users to prioritize their reasoning rules explicitly, so that they can have a clearer idea of how certain conclusions are reached, which in turn will enhance the decision making process in the future.

- It is pointed out from Multi-Criteria Decision Making (MCDM) research that ranking the importance of the criteria is an important first step in solving a MCDM problem [Tabucanon88]. Since a commitment reasoning process is also a MCDM process (refer to Section 2.4.2), this framework can provide support for ranking the criteria.

The rulebase language used in CRSS will be discussed in Section 5.2.2.

5.1.2 Sentimental Factors Analysis

The Sentimental Factors Analysis supports the aspect of commitment reasoning that deals with all the sentimental factors (as discussed in Section 3.4). As pointed out in [Hewitt91], sentiment is how people feel about each other's behavior. In other words, people have different interpretations (according to their value systems) about different behaviors. Also pointed
out in [JanMan77], sentimental considerations such as self-approval and social approval play important roles in the kind of commitments a person makes.

In CRSS, we feel that it is impractical and ineffective to quantify sentimental factors and put them in rule form. For example, it would not be sensible to quantify how much the agent thinks of himself as being helpful by committing to help a friend. A list of sentimental factors are extracted from various literature in commitment studies [Gasser91] [Hewitt91] [Bond90] [Brickman77] (e.g., guilty, helpfulness, willingness to learn, superiority, etc.) with the intent to bring these factors to the agent's awareness, and let the agent interpret the 'meaning' of the commitment according to his value system (i.e., linkage of the 'want to' portion). Bringing the relevant factors to the user's awareness is a crucial factor to good decision making [Conklin91] [JanMan77] (Section 2.4.3) — research has shown that people tend to make only satisfactory decisions but not optimal decisions due to the overlooking of some important factors [Brickman77].

At the end of the Analysis, the user has to enter his final decision on whether or not to make the commitment, and to write a short paragraph to explain and clarify his decision rationale (the linkage between the 'have to' and 'want to' parts of the commitment). Research has shown that explicitly stating the decision rationale can improve the quality of the decisions made [Conklin91] [JanMan77] [HelMarWoo] (Section 2.4.3).

5.2 System Design

There are three components in the CRSS (indicated as ovals in Figure 5.2): (1) Main Menu, (2) Side Bet Analysis (interacts with a CRSS Rulebase), and (3) Sentimental Factors Analysis (interacts with a CRSS Casebase).

The three components of CRSS are displayed sequentially. In other words, users will first see the Main Menu where the request is displayed and then pressing [Process Request] will lead the user to Side Bet Analysis where the information for the request is converted to facts in the rulebase. Pressing the [Sentiment] button takes the user to the Sentimental Factors Analysis, where relevant sentimental factors can be examined, and the final decision and the
Figure 5.2: CRSS System Design

In-coming Commitment Request

Receiving a request from another agent

CRSS Main Screen

CRSS Rulebase

- Convert Request Info. to facts in rulebase
- Process request using the rulebase. Allow user to edit rulebase

CRSS Casebase

- Old cases can be retrieved. Current case can be saved with final decision

Sentimental Factor Analysis

- User can go back and forth between the two subsystems before making the final decision
- Decision Support via the use of table of sentimental factors

Display Request to User

User Action

Recommendation to User

User Action

Display Table of Factors to User

User Action

Decision is made. The request and all results from CRSS are saved in Casebase. The entire reasoning support process is terminated.
decision rationale are recorded.

The sequential ordering of the three subsystems is strictly an implementation limitation (the screen cannot fit all 3 subsystems simultaneously), no precedence is assigned to them in our design. The user can access any of the subsystems by clicking the corresponding buttons whenever he finds necessary. The functions of each component are described below.
5.2.1 The Main Menu

In this component, a commitment request is presented in a special electronic mail message form called Commitment Request Form (see Figure 5.3). The first three fields on the Form indicate the sender, the receiver, the received date of the request respectively. The Task Type field indicates the type of commitment (used by CRSS Casebase for case retrieval). The Task Description is a detailed description of the request task. The Task Requirement/Rewards is where the rewards, resource requirements (including date and period of the commitment) are indicated.

At the current stage of implementation, this subsystem is responsible for the following functions:

- Allow users to browse new requests.
- Give users the options to go to the other two parts of the system.
- Process a request — convert the information of the current commitment request into facts in the rulebase so that the system can perform the Side Bet Analysis.

5.2.2 The Side Bet Analysis

This part of CRSS is based on the Preferred Subtheories framework.

The Side Bet Analysis has the following capabilities (Figure 5.4):

1. Allow user to alter (add, delete, & edit) rules and facts by pressing the [Edit Rulebase] button on the top right corner of the window in Figure 5.4. As indicated in Figure 5.5, a rule-construction panel is used to assist users to construct rules.

2. Allow user to re-process the request by the modified rulebase if necessary.

3. Display the result of the process (i.e., a recommendation and the rules that are fired). This is the bottom window in Figure 5.4.

4. Allow users to go to Sentimental Factor Analysis (the icon with an eye on the right of the window in Figure 5.4) when they are satisfied with the rulebase recommendation.
The following is the rulebase language adopted for the Side Bet Analysis:

- **Predicates** are the main form of representation (always start with lowercase alphabetes). They describe the relationships between entities. For example the predicate:

  \[
  \text{authority}(\text{robert}, \text{advisor})
  \]

  means that *Robert's authority* is an *advisor*. In other words, authority is the relationship between the entity Robert and advisor.
In the CRSS, a set of **Domain-Independent Predicates** and a core rulebase are provided (see Appendices A and B). Users are allowed to add, edit, or delete any domain-dependent predicates according to their specifications.

- **Variable names** always start with uppercase alphabetes (e.g., V1, Agent, TASK), while **constants** start with lowercase alphabetes (e.g., robert, advisor).

- A **fact** is stated as a predicate with a period (e.g., `cmt_type(design_project)`).

- A **rule** is stated as follow:

  \[ \text{R##.##}: \text{IF condition THEN action} \]

Each rule begins with a rule number: R##.##. The ## before the period is the level number where the rule is resided in, the other ## is a number that is unique to that rule in that level (the convention is to use the number of rules). For example R2.12, means it is the twelfth rule at level 2.

A **condition** can be:

![Figure 5.5: CRSS Rule Inspector](image-url)
- conjunctives (AND) and/or disjunctives (OR) of predicates
- conjunctives and/or disjunctives of comparison of variables or mathematic expressions of variables. For example:
  \[ Var_1 = Var_2 \text{ OR } Var_3 = Var_2/Var_1 \]

An action can be a conjunctive of the following:

- COMMIT or NOT COMMIT
- NOT R##.##. This action is for blocking the application of a rule.
- predicate. This is used to create new predicates.
- REPLACE predicate WITH predicate. This is for changing the values of the arguments in the predicate. For example, REPLACE timeavail(40) WITH timeavail(30) changes the argument from 40 to 30.

An example of a legitimate rule:

\[ R1.1: \text{IF } constraint(T1) \text{ AND } cmType(T2) \text{ AND } T1=T2 \text{ THEN NOT COMMIT} \]

This rule says that if there is a constraint on committing to the commitment type T2, then conclude NOT COMMIT.

There are three possible recommendations from the Side Bet Analysis: COMMIT, NOT COMMIT, or UNDECIDED. UNDECIDED means that none of the rules that conclude COMMIT or NOT COMMIT are fired. Regardless of the recommendation, this subsystem will show all rules that have been fired to reach a particular recommendation (see the bottom window of Figure 5.4).

A Core Rulebase

A core rulebase is included in the Side Bet Analysis to provide some basic knowledge for determining the availability of required resources, the importance of the task and the requestor,
Figure 5.6: The Conceptual Model of the Core Rulebase Used in the Empirical Study

The utilitarian gain/loss of the requested commitment, etc. The provision of a core rulebase is an attempt to allow the Side Bet Analysis to draw sensible conclusions based on utilitarian information without the users entering any rules at the initial stage of using the system. The conceptual model of the core rulebase used in the empirical study is shown in Figure 5.6. The core rulebase is constructed based on the research of commitment detailed in Chapters 2 and 3.

It is well known that two different experts of the same domain may have entirely different conceptual models for an expert system, and that getting experts and users to agree on one
conceptual model is a vast research problem by itself [TricDavi93]. Our assumption is that the conceptual model of the CRSS core rulebase is a commonly understandable, agreeable model in the commitment reasoning context, for which most users can build their domain knowledge upon without major difficulties in understanding and agreeing with the model. For example, the core rulebase in Figure 5.6 contains rules which subtract the number of occupied-hours by the total number of working hours to determine the time available for a commitment. It is unlikely that there are many very different ways to determine the availability of time.

In other words, we assume that the core rulebase is readily agreeable and understandable enough that in most cases it does not suffer the consensus problems in domain-dependent expert systems mentioned in [TricDavi93]. Hence, the limitation of the usefulness of this core rulebase exists when meaningful conclusions cannot be drawn in domains where (1) one common conceptual model cannot be extracted or (2) where the conceptual model of the user being incompatible with the core rulebase's. One viable solution to the latter limitation is to provide users with a choice of multiple core rulebases with different conceptual models (see Chapter 8).

5.2.3 Sentimental Factors Analysis

The Sentimental Factors Analysis provides a table that shows sentimental factors that could be affected if committed and if not committed to request. The user can choose whether any of the sentiments are increased, decreased, or unchanged, and he can also alter the list of sentiments according to his context (Figure 5.7). By listing the relevant sentimental factors in a table format, the agent can easily see how some factors could be affected if he was to make a certain decision.

The Sentimental Factors Analysis provides the following functions:

1. display the table of sentimental factors, allow the user to choose whether a factor would be increased, decreased, or unchanged, and to put in short comments to explain his choice.
2. allow users to add/delete sentimental factors according to their needs (i.e., the table of sentimental factors are changable).

3. provides access to casebase which allows the retrieval of previous cases of the same commitment type. The retrieved information included (Figure 5.8):

- the rules fired and recommendation generated.
- the sentimental factors in the Sentimental Factor Analysis.
- the final decision and the justification for making the decision. It is in a format of
a small paragraph where the agent describes in his own words about why he made that decision. This paragraph is called the Linkage, since the agent makes his final decision according to how his value system links up the pros and cons of all factors related to the request. In other words, the agent makes his final decision by weighing the importance of the positive and negative effects of the utilitarian and the sentimental aspects, as well as the relative importance between the two aspects.

4. allow users to save the current request into casebase.

5. allow users to enter the final decision of the commitment request (which is to be saved into the case base). As mentioned above, the user needs to enter a short paragraph (called Linkage) of his justification for making the decision.

5.2.4 Casebase

In the CRSS, a simple casebase is implemented. The main purposes of the casebase are:

- provide decision support by allowing users to refer to analyses of previous commitment requests of similar type.

- keep track of all requests processed by the system in case decisions are questioned in the future (this is to emphasize the accountability of commitment making mentioned in Section 2.3).

- make debugging and upgrading of CRSS in the future easier (since previous cases can be used as a means to detect errors).

For the above purposes, the simple casebase in CRSS is implemented as a database with records of a predefined format. Each record represents a case and contains the following information:
Figure 5.8: CRSS Casebase

- the original Commitment Request Form with Commitment Type as the key for each record.

- the recommendation and rules fired from Side Bet Analysis.

- the table of sentimental factors in Sentimental Factor Analysis.

- the final decision made by the user, and a short explanation for why the decision was made.
The current version of the casebase only allows the stored cases to be retrieved for display in text form on the Casebase screen of CRSS (see Figure 5.8). This means that no information from the current version of casebase can be retrieved and used by other parts of CRSS. Furthermore, the only matching mechanism is the manual selection of Commitment Type from the list of cases shown in Casebase menu. Future work can be done to expand the capability of the casebase in these two aspects (see Chapter 8).

5.3 Implementation

The implementation of CRSS is done on the NeXT computer system. This platform was chosen for its support for building mouse-driven user interface. The Interface Builder makes programming the user interface a relatively simply task. Most of the programming is done in Objective-C, the NeXT version of Object-Oriented-C, while a part of the rulebase interpreter is written in LEX and YACC.

Both the CRSS repositories, the casebase and the rulebase, are kept in text files. They both have fixed internal formats which are recognized by corresponding read/write subroutines.

The rules in the CRSS rulebase are translated into Prolog statements and then executed by a Prolog interpreter. The sophisticated inferencing capability of the Prolog interpreter justifies the extra work of converting the rulebase into Prolog statements; it would be much more complicated to program the entire inference engine from scratch. Outputs from the Prolog interpreter are interpreted by CRSS’s own interpreter (i.e., the CRSS rulebase interpreter interprets things such as the correct leveling of rules, the blocking of rules, the conflict of conclusions, etc.) in order to provide the correct outputs to the user. The part of CRSS’s own interpreter that is responsible for translating the rulebase language into Prolog statements is written in LEX and YACC. LEX and YACC are programming languages specially designed for writing lexical analyzers and parsers respectively. These two programming languages can
easily be translated into C statements\(^1\) which can be integrated with the rest of the codes. This approach is less complicated than writing the interpreter in Objective-C.

In principle, the CRSS's interpreter goes from the rules of the highest priority to the lowest. It records the level of the first rule which concludes COMMIT or NOT COMMIT and overrides all the contradictory conclusions made by lower level rules, but all fired rules (including the overridden rules) are recorded and shown to the user. A list of rules that are blocked are kept in a file, procedures are called to check that blocked rules will not be fired and that higher priority rules cannot be blocked by lower priority rules. The end product is a conclusion (COMMIT, NOT COMMIT, or UNDECIDED) and a list of all rules that are fired.

\(^1\)C is a subset of Objective-C
Chapter 6

Two CRSS Examples

The following are two examples to illustrate the features in CRSS. The first example is one of the scenarios used in the CRSS empirical study. The second example is a real life purchasing commitment problem with a substantial financial implication encountered by the author.

6.1 Presentation in a Faculty Meeting

Example 6.1.1 The agent is a master's student in an academic institution. He receives a request coming from Bob Smith, the Associate Dean of his Faculty, asking him to represent the Graduate Students' Association (GSA) in the bi-monthly Faculty Meeting. The meeting will last 2 hours and the agent needs to conduct a 30-minute presentation in front of the Faculty members (Figure 5.3). The presentation is to summarize the activities and the financial status of GSA.

1. The request comes in the form in the Main Menu of CRSS in Figure 5.3. When the user presses [Process Request] at this point, the Side Bet Analysis screen will show up like Figure 5.4. All information from the request form is already converted to facts in the rulebase (in the current version of CRSS, the request information is converted to predicates in the rulebase manually), and the request has already been processed at this point.
2. As indicated in Figure 5.4, CRSS suggests COMMIT. The user can go through the information provided in the bottom window of Figure 5.4 to find out how CRSS arrived at this recommendation:

- the requestor is an important person,
- the task type is public in nature, and thus is an important task type,
- and the user will gain presentation skill after committing to the task.

The rulebase recommends COMMIT despite the fact that there is a time conflict, and that the commitment does not provide any monetary compensation for the time the
agent will spend. Although not implemented in the current CRSS, this window could potentially display the use of rules in a graphical form as shown in Figure 6.1.

3. The user has the options to edit the rulebase and then re-process the request. A Rule Inspector is provided to support rule editing (Figure 5.5). After editing the rulebase, one can re-process the request by clicking the **[Process Request]** button on the Side Bet Screen.

The nonmonotonicity of this rulebase effectively decreases the chances of inconsistency (i.e., two or more rules drawing conflicting conclusions) by allowing users to put rules into multiple levels. By cranking the rulebase through the inference engine, users will find out the fired rules and the levels they reside in, hence he can decide what level to put the new rules in order to have the result he desires. The process can be repeated until the desirable result is reached. This nonmonotonic feature effectively simplifies the task of maintaining consistency of the rulebase by reducing the need for tempering with the rest of the rulebase.

4. After the user has obtained the result in the Side Bet Analysis, he can click on the **[Sentiment]** to proceed to the Sentimental Factor Analysis screen (Figure 5.7). On this screen, the agent can indicate effects he thinks the current commitment request would have if he decided to commit or not commit. He can retrieve cases with the same commitment type from casebase (Figure 5.8) for references.

5. At the end, the user is to input his final decision as whether to commit or not (in the Sentimental Factor Analysis section), and he is to write down a short explanation of the rationale of his decision (Figure 6.2). This explanation has the following two purposes:

- to ensure that the user understands his decision rationale (it is the linkages between positive and negative factors, and between the 'have to' and 'want to' aspect).
- to make the casebase more informative for future use (i.e., future users will obtain more information on how the final decision was reached from this paragraph).
Finally, the user can click on the [End CRSS] button to end the entire CRSS session.

6.2 Purchasing Car Stereo

Example 6.2.1 The agent wants to decide whether he should purchase an automobile compact disc/radio stereo unit. It costs about $500-$600 which is a large amount of money for the agent's financial status. On one hand, the agent does not want to spend that kind of money on a luxurious item, but on the other hand, he has a strong desire to acquire the unit as a means to improve the sound quality of his car stereo. Therefore, he issues himself a commitment request in CRSS hoping that CRSS can provide him some reasoning support.

1. The agent sends a request to himself as shown in Figure 6.3.

2. After the user has pressed [Process Request] at this point, then the Side Bet Analysis screen will show up like Figure 6.4. The CRSS rulebase suggests that NOT COMMIT for the following reasons:
3. Since this request concerns little sentimental factors, the agent skips the sentimental factor analysis and goes on to make his final decision (Figure 6.5). He decides to commit to buying the unit only if it costs less than $550.
What makes this example interesting is that it was actually a real-life problem faced by the author himself. Before using the CRSS, the author could not make up his mind on whether or not he should buy it. In fact, he had changed his mind several times before he decided to seek help from the CRSS. After using the CRSS, the author not only felt that the CRSS clarified his reasoning process by making him understand that he was basically spending $550 in return for the satisfaction of owning and listening to a stereo unit, he also felt that by recording his final decision and justification ('linkage') in CRSS, his decision was made 'official' and he no longer had the intention to change his mind. At the end, the author
did not buy the unit since he could not get a deal below $550.

This example illustrates that (1) CRSS does help the user to clarify his thoughts (sometimes by explicitly stating the obvious but overlooked facts) in his reasoning process, and (2) that it helps to bind the user to his final decision by requiring the user to put his decision and decision rationale in writing. The second point is an especially important contribution because it is the binding effect that characterizes the concept of commitment which inspired the CRSS research. As pointed out in Section 2.3, this binding effect augments the accountability of decision making which is not emphasized in the current DSS research. After all, a DSS is of little use if the user never implements his decisions.
Chapter 7

Empirical Study of CRSS

A series of semi-structured empirical testings were conducted on CRSS implementation. Three subjects were used in the testings. All subjects were master’s students of the Management Information Systems Division of the Faculty of Commerce. One of them is a second year student, and the others are third year students. Each testing lasted approximately 2.5 to 3 hours and were all conducted in separate days.

Before conducting the actual testing with the subjects, a pilot study was conducted with a full-time research assistant in the Faculty of Commerce. The intention was to detect any implementation problems as well as testing procedure problems.

The basic structure of the study is to let subjects use CRSS on a set of provided commitment request scenarios, and then collect their feedbacks (details of the testing procedure are provided in Section 7.2 and Appendix C).

7.1 Objective

The objective of the testing is twofold:

1. **Gain implementation insights:** the usefulness of the CRSS, as a system to support commitment reasoning. In particular, we hope to find out:

   - situations when CRSS is most practical, and situations when it is not as practical.
- the limitations of CRSS. For instance, the limitations on the expressiveness of the rulebase.
- the possibilities for implementation improvements (that is, how information can be presented on screen to the user to provide better support).
- the possibility of providing a core rulebase in the Side Bet Analysis. The intention is to make CRSS a functional system without the painstaking effort of entering the rules on the user side.

2. **Gain conceptual insights**: in particular, the commitment reasoning process of the subjects vs. the commitment reasoning model of CRSS, which can:

   - provide valuable knowledge for the advance of commitment reasoning research in the future.
   - provide ideas for conceptual improvements of CRSS. For instance, there might have been factors in the reasoning process for which the conceptual model of CRSS does not cover.
   - act as an indicator of the usefulness and the limitations of CRSS.
   - indicate the usefulness and problems of the application of default reasoning.

### 7.2 Testing Procedure

The assumption for the testing is that subjects are being rational\(^1\) or at least attempting to use CRSS as a tool to enhance their reasoning rationally. A person makes a decision based on what he 'feels like' is an example of irrational reasoning.

Since CRSS is designed to be used in organizational environments, we decided to choose our subjects from the organization that we are most closely related to — our Faculty of Commerce. A series of academic commitment request scenarios were used for which subjects could easily relate to:

---

\(^1\)To be rational means to have well-understood and well-supported reasons for every decision made.
1. Request by an associate dean to present in a Faculty meeting (Example 6.1.1).

2. Request by a staff member of the Job Placement Office for an interview on how to improve the Office's service.

3. Request by a master's student to help out in a new student orientation.

4. Request by the thesis supervisor to become a marker.

5. Request by another master's student to be a subject for an empirical study.

6. Request by yet another master's student to comment on his research paper. This request is carried out three times in order to study the change in the reasoning process when the subject repeatedly receives similar requests from the same requestor.

A core rulebase, 6 sets of sentimental factors, and a hypothetical weekly class schedule are provided for the testing. The core rulebase consists of rules for resource allocation, scheduling, and various cost/benefit analysis (see Section 5.2.2, and Appendix A and B). The sentimental factors are based on the pilot studies' results and research results (refer to Chapters 2 and 3).

At the beginning of the testing, the structure of the CRSS and the purpose of the testing are briefly explained to the subject. During each testing, the subject is to process all the request scenarios consecutively with the CRSS.

Subjects are allowed to alter the core rulebase and the sentimental factors to fit their needs. Assistance is provided when called upon by the subject. During and after the processing of each request, the subject's comments are recorded. Questions are raised to the subject whenever necessary to elicit feedbacks. A discussion at the end of each testing session is carried out to summarize the subject's overall comments concerning both the implementation and conceptual aspects of CRSS. For a detail description of the testing procedure see Appendix C.
7.3 Testing Results

The results are categorized into implementation and conceptual:

1. Implementation Results:

- All subjects found the core rulebase of the Side Bet Analysis useful in providing resources analysis such as locating conflicts in schedule, calculating the availability of required resource, and calculating the utilitarian gain/loss of the commitment. From the practicality prospective, this feature of CRSS is highly desirable because very few users are willing to spend a long period of time entering their rules before using a new system.

  This result confirms with the assumption that the core rulebase normally can provide reasonable amount of support without major customization. However, we do not eliminate the possibility that there may be domains where such a core rulebase is not applicable.

- No major difficulties were encountered when subjects needed to put in their own rules in the rulebase (with the researcher’s assistance), although some rules needed to be expressed slightly different in the rulebase. Subjects found the leveling feature of the rulebase very practical for prioritizing their rules.

  The integrity of rulebase was unaffected after the entering of new rules, meaning that the rest of the rulebase did not need to be changed when entering new rules.

- The rulebase language was perceived as ‘programming language-like’ by some subjects. It was suggested that it should be translated into a more ‘natural language’ format or presented in a form which users can easily understand the reasoning of the rulebase.

  We have adopted the suggestion from the first subject to present the reasoning process of the rulebase in a flowchart form (Figure 6.1). The subjects followed had found the flowcharts easier to understand than a list of rules.
• All subjects found the Sentimental Factors Analysis useful in reminding them the sentimental factors they need to consider. However, all subjects were unsure of the significance of Sentimental Factors Analysis in relation to Side Bet Analysis. One subject perceived the Sentimental Factors Analysis as more important, while the other two thought the Sentimental Factors Analysis was a supplement of the Side Bet Analysis.

The confusion was caused by the order that the two parts were presented to the subjects — Side Bet Analysis came before Sentimental Factors Analysis. The design of CRSS does not assign precedence to them, however the screen was not large enough to hold both parts side-by-side simultaneously.

We compensated the problem by stating to the subjects in the subsequent sessions that the two parts had no precedence over each other, they should be considered as two attributes of a commitment.

• In terms of overall performance of CRSS, subjects in general felt that the system was still a research version, but they expressed that they would use CRSS when the request was ‘difficult’ to decide. Summarizing the feedbacks from all subjects, the difficulty of a request lies in the complexity and the importance of it. The complexity of a request is proportional to the number of factors needs to be considered, while the importance of a request is proportional to the magnitude of its implications to the agent. Therefore, a ‘difficult’ request is one with high complexity and/or importance implications.

• Since the scenarios used in the empirical study are hypothetical, the system's effect on the accountability of the decision making could not be tested directly. However, the fact that the subjects expressed their willingness to use CRSS on ‘important’ commitment requests gives some indication that they felt that CRSS could provide a finalizing effect on their decisions. That is, they felt that they would be more likely to stick with their decisions after using CRSS.

2. Conceptual Results:
All subjects agreed that the CRSS rulebase was an appropriate model for their reasoning processes concerning resource allocation and scheduling under most situations. However, it was noticed that subjects did not always think like the CRSS model of reasoning. Although in most cases it was possible to convert the way the subjects reasoned into the model, CRSS was not able to support effectively in the following situations:

- When all factors in CRSS suggested commit, yet the subjects still insisted on 'not commit', yet they could not provide explanations for doing so. This could be caused by subjects' reluctance to reveal the real reasons, or by the fact that subjects simply did not understand their own reasoning process at the moment.

One possible explanation of why the subjects were unaware of the important factors that shaped their decision is 'the operation of repression and other psychological defense mechanisms' (p. 140, [JanMan77]).

- When subjects let some psychological factors, such as bad mood, laziness, etc., to decide whether or not to commit.

Under these situations, the subjects were considered to be irrational and were therefore considered to be outside the boundary of the application of CRSS. However, these phenomena should inspire interesting research ideas in the future.

One of the subjects expressed concern that the CRSS rulebase result might have a leading effect in her reasoning process, and that its conceptual model might not be as commonly understandable and agreeable as it was assumed to be. To address the concern, we decided to ask subjects to go through certain requests with their own reasoning process prior the use of CRSS, so that they could compare their conceptual model with the rulebase's conceptual model. At the end, one subject agreed the rulebase's conceptual model usually coincided with her own model, and the other two subjects expressed that its model could easily be understood although they might not necessarily always think that way. The two subjects also
stated that they sometimes used one single most important factor (or rule) to make
decisions. For example, one subject decided she would not commit to any request
as long as there was a time conflict with a class, regardless of the requestor or
the nature of the task. CRSS provides support in such a reasoning process in the
following ways:

- The single most important factor can be entered as a rule (at a high level) in
  the rulebase. This way, all other less important rules will be overridden.
- CRSS can help identify that determining factor. In several occasions, subjects
  indicated the determining factors would not have been determined if they had
  not used CRSS. CRSS supported the reasoning process by reminding the user
  of relevant factors and rules.

- Sentimental Factors Analysis was found especially useful when:
  - the requestor and requestee had close organizational and/or social relationship
  - the commitment was a public one (that is, the carrying out of the commitment
    will be seen by the public).

- As pointed out in the implementation results, the CRSS rulebase was easy to mod-
  ify and provided sensible conclusions. However, both the subjects and the author
  felt that some of the sentimental factors, such as self-image and social approval
  factors, could not be sensibly represented in rule-form. Both the subjects and the
  author indicated that the ‘table’ format used in Sentimental Factor Analysis was
  a more effective representation. This observation reflects a possible limitation of
  applying default reasoning in DSS that needs to be compensated by other means.
Chapter 8

Conclusion and Future Work

In this thesis, the concept of commitment is studied. Our focus is in studying the characteristics of commitment so as to develop a system that can support the process of commitment reasoning. We have proposed a system called the Commitment Reasoning Support System (CRSS), and have developed an implementation of it on the NeXT Computer System. The system is based on Brickman's concept of commitment which is characterized by the interactions between the utilitarian and sentimental factors.

The semi-formal empirical study conducted on CRSS shows encouraging results and interesting findings. In the practicality aspects, results show that CRSS can provide reasoning support by:

- providing a core rulebase to handle scheduling, resource allocation, and other quantifiable factors in commitment reasoning; the system also allows users to modify the rulebase to fit their needs.

- reminding the user to consider relevant sentimental factors in the process.

- finalizing the reasoning process and the final decision by requiring the user to record his final decision and his reasoning process. That is, it increases the accountability of decision making.
CRSS is most useful for requests with **high complexity** and/or **important implications** since it helps to clarify and reveal the relevant factors need to be considered, and to finalize the decision made (i.e., to make the user feel committed to the decision by providing him with a thorough analysis of the request).

In terms of conceptual aspects, we observed that psychological factors played an important role in the reasoning process, that subjects sometimes used one single determining factor to make their decisions, and that resistance occurred when subjects were asked to explain their reasoning processes.

Combining the findings of the two aspects, the following are the possible extensions of CRSS such that it may become a useful component in organizational environments (one which supports complicated and important requests as well as routine requests):

- have a more natural interface for the rulebase, so that the non-technical users can easily understand the inputs and the outputs of the rulebase. One possible approach is to use a diagrammatic interface as suggested by one of the subjects (see Figure 6.1).

- further research to incorporate the **psychological aspects** of human reasoning process into CRSS.

- provide a core rulebase which contains organizational knowledge. Although CRSS is intended to support commitment reasoning at an individual level, organizational level knowledge such as codes of conduct, administrative procedures, conventional way of doing things, etc., needs to be included in the rulebase in order to provide sensible support. One possible way to obtain organizational knowledge is to connect CRSS with various knowledge bases and databases in the organization.

- provide multiple core rulebases (with different conceptual models) to choose from (1) when one common conceptual model cannot be agreed on within a domain, or (2) when the CRSS is used in different domains within the same organization (e.g., the finance department, the engineering department, etc.). The provision of multiple core rulebases requires different ways of interaction between the rulebases and organizational
knowledge bases. A possible way to select the appropriate core rulebase is through the use of a casebase. That is, users can retrieve the core rulebase used in a matched case.

- expand the capability of the casebase (1) by providing a more sophisticated case matching mechanism so that matching can be achieved in a variety of ways, and (2) by enabling core rulebases and sentimental factors used in previous cases to be applied to the current case.

- incorporate CRSS into the organizational electronic mail network so that commitment requests can sent and received by workers via electronic mail.

OASIS [MartWoo92] [WooLoc92] could be a possible implementation platform for these extensions.

In summary, this project has not only shown the possibility of providing computer support for commitment reasoning, it has also shown the applicability of default reasoning on commitment reasoning (via the CRSS rulebase). The leveling feature is not only an intuitive way for the user to represent his reasoning process, it also simplifies the task of consistency maintenance which can be very complicated in conventional rulebase systems.

Furthermore, the empirical study results demonstrate the usefulness of CRSS in handling highly complicated and important commitment requests, as well as providing new insights in commitment reasoning. Although the current version of CRSS is still far from being a system that can be as practical as spreadsheets and database systems in organizations, it nevertheless has provided a building block for a promising research area which may benefit many organizational workers in the future.
Bibliography


Appendix A

Domain-Independent Predicates in CRSS Rulebase

- Tangible Resources Predicates

resource(TYPE, NAME, QUANTITY)

TYPE: money, space, material, skill/technology

NAME: specific types of money, space, material, skill, and technology

QUANTITY: the available quantity of the resource

- money — in dollars and cents
- space — in square meters
- material — in appropriate units (e.g. paper in number of sheets, gasoline in liters, etc.)
- skill — in skill levels (depends on the domain of application, for the purpose of the empirical study, +1 means the agent has the skill, -1 means he does not have the skill).
- technology — the availability of software and hardware. A particular technology can have two status: +1 (available), -1 (not available).

resource_value(TYPE, NAME, QUANTITY) — the value (in dollars) per unit of a kind of resource (defined by the agent; same format as the resource predicate).
occupied(DAY,BEGIN,END,EVENT) — on DAY of each week from BEGIN to END hours is occupied by EVENT.

timeavail(X) — the amount of hours available for each week.

- Authority Predicates

  authority(X,pos) — Agent X’s position is pos in the organization

  important(requestor) — the requestor is an important person

- Commitment Type Predicates

  cmt_type(TYPE) — the commitment is of type TYPE (e.g., presentation, meeting, commenting, etc.).

  cmt_nature(NATURE) — NATURE of a commitment can be private, public, dangerous, or other adjectives to describe the nature of the commitment.

  important(task) — the commitment request is important

- Requestor Predicates

  requestor(X) — Agent X is the requestor of the commitment

- Constraints Predicate

  constraint(T) — commitment type T is prohibited to be committed to

- Commitment Requirements Predicates

  required_resource(TYPE, NAME, QUANTITY) — same format as the resource predicate. QUANTITY — positive value means the amount is rewarded to the requestee, negative value means the amount is required from the requestee when carrying out the commitment.

  cmt_date(DAY,BEGIN,END,DATE) — meaning that the commitment needs to be carried out on DAY of the week from BEGIN to END hours, on exactly DATE.
\textit{cmt\_time}(Y) — meaning that Y hours are needed for carrying out the commitment.

- \textbf{Resource Utility Predicates}

\textit{no\_resource}(TYPE, NAME, QUANTITY) — indicates that there is a shortage of a certain required resource (short by QUANTITY amount). This predicate is usually generated by a rule that calculates the difference between the required resource and the available resource, while \textit{resource}(TYPE, NAME, QUANTITY) is a predicate to record the amount of resource available to the agent.

\textit{cmt\_value}(Z) — the calculated value of commitment in terms of dollars.

\textit{time\_conflict}(DAY, CB, CE) — there is a time conflict with the commitment on DAY of the week between CB to CE in hours.

\textit{will\_gain\_resource}(TYPE, NAME, QUANTITY) — agent will gain resource if committed (same format as the \textit{resource} predicate).

\textit{should\_not\_cmt}(REASON) — meaning that the agent should consider not to commit due to REASON.
Appendix B

CRSS Core Rulebase

authority(self,msc).
authority(smith,asso.dean).
authority(malcolm,msc).
authority(fu,msc).
authority(wong,placement).
authority(edwards,advisor).
instructor(comm525,smith).
resource(money,grant,200).
resource_value(material,coffee_cookies,3).
resource_value(material,t_shirt,20).
resource_value(material,luncheon,20).
resource_value(time,cmt_time,11).
resource_value(money,,1).
time_avail(40).
cmt_value(0).
cmt_time(0).
occupied(fri,1200,1400,gsa.meeting).
occupied(mon,900,1100,comm522).
occupied(wed,900,1100,comm522).
occupied(mon,1400,1600,comm580).
occupied(wed,1400,1600,comm580).
occupied(mon,1100,1300,mis.workshop).
exam(comm525,1200,1400,9/5/93).
occupied(tues,1200,1400,comm525).
occupied(thur,1200,1400,comm525).
occupied(tues,1400,1600,comm635).
occupied(thur,1400,1600,comm635).
occupied(sun,1200,2200,study).

R1.0: IF required.resource(TYPE,NAME,VALUE1) AND resource(TYPE,NAME,VALUE2) AND VALUE3 =VALUE1+VALUE2 AND VALUE3 < 0 THEN no.resource(TYPE,NAME,VALUE3)

R1.11: IF required.resource(A,B,C) AND C < 0 AND NOT resource(A,B,.) THEN no.resource(A,B,C)

R1.12: IF required.resource(A,B,C) AND C > 0 THEN will_gain.resource(A,B,C)

R1.2: IF cmt_type(presentation) THEN cmt.nature(public)

R1.3: IF cmt_type(meeting) THEN cmt.nature(public)

R1.4: IF requestor(X) AND authority(X,asso.dean) THEN important(requestor)

R1.41: IF requestor(X) AND authority(X,advisor) THEN important(requestor)

R1.5: IF occupied(DAY,B,E,_) AND cmt_date(DAY,CB,CE,.) AND CB >= B AND CB < E THEN timeconflict(DAY,CB,CE)

R1.51: IF exam(COURSE,B,E,DATE) AND cmt_date(.,CB,CE,DATE) AND CB >= B AND CB < E THEN examconflict(COURSE,CB,CE,DATE)
R1.6: IF occupied(DAY, B, E,) AND cmt_date(DAY, CB, CE,) AND CE > B AND
       CE <= E THEN timeconflict(DAY, CB, CE)
R1.7: IF requestor(X) AND instructor(COURSE, X) AND examconflict(COURSE, CB, CE,) THEN NOT R2.11
R1.8: IF cmt_date(_, B, E,) AND X = (E-B)/100 AND cmt_time(Y) AND Z =
       X+Y THEN REPLACE cmt_time(Y) cmt_time(Z)
R1.9: IF occupied(_, B, E,) AND X = (E-B)/100 AND time_avail(Y) AND Z
       = Y-X THEN REPLACE time_avail(Y) time_avail(Z)
R1.91: IF required_resource(X, K, Y) AND resource_value(X, K, Z) AND cmt_value(A)
       AND T = Z*Y AND T2= A*T THEN REPLACE cmt_value(A) cmt_value(T2)
R1.92: IF cmt_time(X) AND resource_value(time, _, Z) AND cmt_value(A) AND
       T = (Z*X) AND T2= A-T THEN REPLACE cmt_value(A) cmt_value(T2)
R2.0: IF cmt.nature(public) THEN important(task)
R2.1: IF timeconflict(DAY, CB, CE) THEN NOT R8.2
R2.11: IF examconflict(_, _, _, _) THEN NOT COMMIT
R2.12: IF no_resource(skill, B, C) THEN willGain_resource(skill, B, -C)
R2.2: IF timeconflict(DAY, CB, CE) THEN should.not_cmt(timeconflict)
R2.3: IF time_avail(X) AND cmt_time(Y) AND X<Y THEN should.not_cmt(timeshortage)
R2.4: IF cmt_value(X) AND X<0 THEN should.not_cmt(loss)
R6.0: IF important(task) AND important(requestor) THEN COMMIT
R6.1: IF will_gain_resource(_, _, _) AND important(task) THEN COMMIT
R7.0: IF no_resource(TYPE, NAME, VALUE) AND TYPE <> skill THEN NOT COMMIT
R8.1: IF important(task) OR important(requestor) THEN COMMIT
R8.2: IF will_gain_resource(_, _, _) THEN COMMIT
R9.0: IF should.not_cmt(X) THEN NOT COMMIT
R10.0: IF NOT should_not_cmt(X) THEN COMMIT
Appendix C

Empirical Testing Procedure and Data

C.1 Testing Procedure

Before the testing begins:

1. Explain to the subject the purpose of the empirical study.
2. Briefly describe the components of CRSS.
3. Explain to the subject that in the empirical study his role is a master’s student at the Faculty of Commerce of UBC. Given a hypothetical weekly class schedule, he needs to decide on whether to commit to six commitment request scenarios (see Section C.2).
4. Ask the subject to raise any questions before the testing begins.

The following procedure is carried out for each scenario:

1. Show the subject the commitment request in the Commitment Request Form of the Main Menu. Ask the subject to raise any questions if he does not understand any parts of the request.
2. When the subject indicates complete understand of the request, he is told to analyze the request by himself before proceeding to use CRSS.
3. When the subject is ready, he is told to first process the request by the Side Bet Analysis by pressing the [Process Request].

4. At the Side Bet Analysis, the subject is directed to read the lower window which contains the result generated from the rulebase. The subject is encouraged to ask questions about the rulebase and results generated until he is clear about how the rulebase functions.

5. Ask the subject to comment on the following concerning the Side Bet Analysis:

   (a) The ease of understanding of the rulebase language. If the subject expresses difficulty in understanding the language, he is asked to suggest possible amendments.

   (b) The ease of understanding of the reasoning process of the rulebase (i.e., the conceptual model of the rulebase).

   (c) The reasoning process of the rulebase in comparison with his own reasoning process. If they are different, the subject is asked to identity the difference as precise as possible. The facilitator will try to resolve the difference by modifying the rulebase.

6. If rulebase is modified, the request will be re-processed to make sure the rulebase can generate the desired result.

7. The subject is asked to address any other concerns of the Side Bet Analysis before moving on to the Sentimental Factors Analysis.

8. At the Sentimental Factors Analysis, a list of sentimental factors (per scenario) is loaded on to the screen. The subject is asked to think of sentimental factors as ‘things concerning how one expects oneself and others to perceive oneself if a commitment is made or not made.’ The facilitator will then describe how the Analysis works.

9. The subject is asked if further explanation is needed on how the Sentimental Factors Analysis functions.

10. The subject is allowed to alter the list of sentimental factors to suit his needs, and the alterations made are recorded.
11. Then the subject is instructed to indicate the effect of the commitment on the sentimental factors: increased, unchanged, or decreased.

12. Upon completing the Sentimental Factors Analysis, the subject is required to comment on the usefulness of this analysis in supporting the reasoning process:

   (a) How it is useful.
   (b) When it is useful.
   (c) The usefulness in comparison with the Side Bet Analysis.

13. The subject is then told to combine the results from the two analyses and make a decision on whether or not to commit to the request. When the decision is made, the subject is instructed to press [Final Decision] button to record his final decision and his decision rationale (in the Linkage field).

14. When finished, the entire analysis is saved into a casebase. The subject is then introduced to the Casebase component of CRSS, and he is reminded that the casebase can be used to support the analytical process in the future use of CRSS.

15. At the end of each session, the following questions are asked regarding the scenario just processed with CRSS:

   (a) Do you think the system is useful in this case?
   (b) How do you think it would be more useful in this case?
   (c) Do you think CRSS helps you to solidify your decision? That is, do you think it helps you to feel more committed to your decision?
   (d) Which part(s) of the system do you like most in this case?
   (e) Which part(s) of the system do you dislike most in this case?
   (f) Which part(s) confuses you the most?

16. The subject is instructed to ask any questions before proceeding to the next scenario.
When all scenarios are completed, these questions are asked before ending the session to get the subject’s overall impression of CRSS:

1. How useful do you think CRSS is in supporting commitment reasoning?
2. In what ways does CRSS coincide and complement your commitment reasoning process?
3. What kind of commitment requests do you think CRSS would be most useful for?
4. Would you use it in real life situations? If so, when?
5. Which part(s) of the system do you like most based on the six scenarios you have processed with CRSS?
6. Which part(s) of the system do you dislike most based on the six scenarios you have processed with CRSS?
7. What improvement(s) would you like to see in CRSS?
8. Please give a summary about the overall impression of CRSS.

C.2 Testing Data

Six request scenarios:

1. **Requestee:** Subject's name
   **Requestor:** Bob Smith
   **Received Date:** 21st August, 1993
   **Task Type:** Presentation
   **Task Description:** You are invited to be the representative of the Graduate Student Association (GSA) in the bi-monthly Commerce Faculty meeting. You would be asked to do a 30-minute presentation in front of the Faculty during the meeting. The presentation will be a summary of the activities of GSA for the previous 2-month period.
Task Requirements:

skill required: presentation skill

knowledge required: GSA activities

meeting date: 9/15/93, 14:00-16:00

2. Requestee: Subject's name

Requestor: John Lee

Received Date: 27th August, 1993

Task Type: Presentation

Task Description: You have been chosen to be one of the 'Orientation Leaders' for 'UBC Orientation 94'. You would be assigned with a group of newly enrolled graduate students for a 3-day period. And your responsibilities include holding a campus tour, helping the students to register, coordinating various social events for the orientation, and helping the new students get acquainted with the new environment.

Task Requirements:

skill required: presentation skill

skill required: leadership

reward: T-shirt, luncheon with Faculty Dean at Pan Pacific.

meeting date: 8-30-93 to 9-1-93, 7am to 6pm.

3. Requestee: Subject's name

Requestor: Grace Wong

Received Date: 27th August, 1993

Task Type: Interview

Task Description: Faculty of Commerce's Placement Office is conducting a series of interviews with Commerce students. The purpose of the interviews is to collect
students' comments on our placement services in order to improve them in the future. You have been chosen to be one of interviewees. You would be interviewed by one of our staff members for a 3-hour period. You would be asked about your family, social and academic backgrounds, your impression of the Faculty of Commerce and the Placement Office, and things to improve for the Office. Your participation would be greatly appreciated by the Placement Office.

**Task Requirements:**

reward: free coffee and cookies

meeting date: 9-8-93, 1pm - 4pm.

4. **Requestee:** Subject’s name

**Requestor:** Steve Edwards

**Received Date:** 30th August, 1993

**Task Type:** marker

**Task Description:** UBC needs YOU as a marker for COMM 336. The responsibilities include marking of 8 homework assignments, in which 1-2 of them will be group projects. Also, amongst the individual assignments, 1-2 of them will be spreadsheet/database/expert system assignments, while the others are case studies. The marker will be paid at a 50 hrs subjective/ 50 hrs objective rate, which turns out to be $1500/term. The marker is expected to receive a marking assignment every 2 weeks, and each assignment will take about 12-16 hrs to finish.

**Task Requirements:**

time required: 12-16 every 2 weeks through out the term.

skill required: mis knowledge (with respect to COMM 336's material).

skill required: unbiased judgement.

5. **Requestee:** Subject’s name
**Requestor:** Philip Fu  
**Received Date:** 30th August, 1993  
**Task Type:** commenting  
**Task Description:** You are asked by one of your fellow Phd students to comment on his paper. Your comments should emphasize on both the strength and the weakness of paper. This job would probably provide you interesting research ideas as well as broaden your knowledge in MIS.  
**Task Requirements:**  
- time required: 2 hours  
- reward: research ideas  
- reward: MIS knowledge

6. **Requestee:** Subject's name  
**Requestor:** Nigel Malcolm  
**Received Date:** 30th August, 1993  
**Task Type:** testing subject  
**Task Description:** You are asked by another MSc student to be a subject for the empirical study of his thesis project. As a subject, you need to work on his 'Group Support System' with 3 other subjects from the Faculty of Liberal Arts. The testing will require much interaction (directly as well as via computers) amongst subjects. Then you need to comment on your experience with the system. The entire study will take 2 to 3 hrs. You will be paid $15 for this task.  
**Task Requirements:**  
- reward: $15  
- testing time: Tuesday, 1:00pm - 4:00pm, 9/15/93