Representation and Assisted Negotiation of Textual Agreements

Emmanuel Celestine Ayeleso

A thesis submitted to the Faculty of Engineering in
partial fulfillment of the requirements for the degree
Doctorate in Philosophy (PhD) in Computer Science

uOttawa

Ottawa-Carleton Institute for Computer Science
School of Electrical Engineering and Computer Science
University of Ottawa
Ottawa, Ontario, K1N 6N5
Canada

Abstract

Research into negotiation systems has primarily focused on those for e-commerce and electronic markets, where *quantitative* values such as prices are key to what is being negotiated. However, there is a lack of research into tool support for complex real-life negotiations of documents that contain large amounts of *textual* (qualitative) clauses. Examples of such text-based agreements include international trade and climate-change treaties, as well as labor-management collective agreements. Our goal is to improve the state of the art in textual negotiation technology, so it can be applied to such agreements and their negotiations. In particular, we want to be able to develop technology that can facilitate the delicate give-and-take involving proposed changes, positions, rationale exchange, partial resolutions to disagreements, tracking of notes taken by the negotiators, as well as the ability to search and compare all of the above to facilitate negotiations. We posit that there would be significant societal benefit from the hyper-local to the international level if better technology was available.

We performed literature reviews of existing negotiation systems and systems for representing legal documents to study what has been done in this domain. We also performed a grounded theory study based on interviews with people that have participated in real-life negotiations. An end-user's survey of negotiation systems was also conducted and analyzed. We used the results from the literature review, grounded theory and survey analysis, as the basis for a subsequent phase of design-science research in which we developed use cases, requirements and a comprehensive metamodel for qualitative negotiation tools, as well as a prototype negotiation tool.

Acknowledgements

I would like to express my deepest gratitude to God, individuals, and institutions who have supported me throughout the journey of completing this thesis.

First and foremost, I am immensely thankful to my supervisor, Prof. Timothy Lethbridge, on whose experience of more than two decades at the negotiation tables, hinge the foundation upon which this research work is built. Your exceptional guidance, expertise, motivation, and unwavering support have been instrumental in shaping this research. Your mentorship has not only enriched my academic experience but has also inspired me to become a better researcher.

I am also indebted to the members of my thesis committee, Prof. Daniela Damian (University of Victoria), Prof. Liam Peyton (EDTI, UOttawa), Prof. Mehrdad Sabatzadeh (EECS, UOttawa), and Prof. Ahmed El-Roby (Carleton University) for their valuable insights, constructive feedback, and dedication to ensuring the quality of this work.

I extend my appreciation to the past and present Complexity Reduction in Software Engineering (CRuiSE) team members who have contributed in one way or the other to the success of this work: Dr. Opeyemi Adesina (University of the Fraser Valley), Reyhaneh Kalantari, others. Also, I wish to appreciate the staff of the Faculty of Engineering, at the University of Ottawa, for providing a conducive environment for learning and research. The resources and facilities made available to me were indispensable in the successful completion of this thesis.

I am grateful to my nuclear family: my wife- Olasimbo Opeyemi Ayeleso and our children- Olakanmi, Olatorera and Timon for their unwavering support and encouragement. Their belief in my abilities has been a constant source of motivation throughout this journey.

I would like to acknowledge the financial support provided by the federal government of Nigeria (FGN) my home country, through the Tertiary Education Trust Fund (TETFUND), and my employer back home in Nigeria, the Federal Polytechnic Ede (FPE) that found me eligible and nominated me for a scholarship through a rigorous process to TETFUND. The FGN, TETFUND and FPE's management generosity made it possible for me to focus on my studies without the full burden of financial concerns.

Lastly, I want to express my gratitude to all those whose names may not appear here but have played a significant role in my academic and personal growth. Your support, whether big or small, has not gone unnoticed.

This thesis would not have been possible without the collective support and encouragement of these individuals, institutions, and God. I am deeply thankful for their contributions, and I look forward to contributing all the expertise that I have gathered, during this long process back into making the world a better place.

Emmanuel Celestine Ayeleso
November 2023

Table of Contents

List of Figures

List of Tables

List of Abbreviations

ACM	Association of Computing Machinery
ALR	Additional Literature Review
AR	Action Research
CASCON	Conference on Computer Science and Software Engineering
CLM	Contract Lifecycle Management
CSCW	Computer Supported Cooperative Work
DS	Design Science
DSS	Decision Support System
EBS	Electronic Brainstorming
ENS	Electronic Negotiation Systems
ENT	E-negotiation tables
GT	Grounded Theory
IEEE	Institute of Electrical and Electronics Engineer
IIASA	International Institute for Applied Systems Analysis
MIT	Massachusetts Institute of Technology
NAA	Negotiation Agent Assistant
NSA	Negotiation Software Agent
NSS	Negotiation Support System
QNSS	Qualitative Negotiation Support System
RFP	Request for Proposal
RQ1	Research question 1
RQ2	Research question 2
RQ3	Research question 3
RQ4	Research question 4
RQA	Systemic Literature Review Research question A
RQB	Systemic Literature Review Research question B
SLR	Systemic Literature Review
UML	Unified Modeling Language
UNCLOS	United Nations Conference on the Law of the Sea

XML	Extensible Markup Language

Chapter 1 Introduction

In this thesis, we study the complex textual negotiation domain using multiple research methods including literature review, grounded theory (GT), surveying, design science (DS) and prototyping. We identify challenges facing the domain, and suggest multiple approaches to tackle these challenges. In particular, negotiations require effective communication support and intensive data management that existing tools accessible to negotiators do not provide. Unfortunately, there are no effective open-source tools available to assist negotiators.

We have gathered data about the negotiation domain via various means; analysed the data; and use the analysis results to develop negotiation models and build an open-source prototype to demonstrate how to support both negotiators and the process of negotiation more effectively.

Negotiation is a subject that has gained the attention of researchers from different perspectives and from multidisciplinary angles. Research into existing negotiation systems (NSs) has been largely focused on *quantitative* aspects, such as pricing, and has been limited to negotiations in e-commerce and electronic markets. A high proportion of such negotiations involve binarized; accept-or-reject decision-taking, or proposing numeric amounts such as prices to be paid, both of which can be obtainable algorithmically.

However, a large fraction of real-life negotiations involves repeated give-and-take discussions about wording changes in *textual* or *qualitative* agreements. These types of agreements are critical in many aspects of society: they include international treaties on topics such as trade or climate change, labor-management collective agreements, and many others. Negotiations to create or modify such agreements often call for intricate compromises and trade-offs regarding the acceptable wording, and are hence quite different from agreements in electronic markets.

The textual agreements on which we will focus can be called contracts, agreements, or treaties. We will use the first two of these terms interchangeably. A contract is a document that contains the mutual agreement of two or more parties that have successfully concluded a negotiation.

As mentioned above, the majority of the existing research efforts on negotiation systems, revolve around determining numeric values for products or services (G. E. Kersten & Noronha,

1999; Klenk et al., 2012; Schoop, 2004; Schoop et al., 2003; Schoop & Quix, 2001). Research on quantitative transactions includes auctioning or bargaining over numeric amounts.

The kinds of textual documents on which we will focus instead contain extensive hierarchal sets of textual *clauses*, often with many subtleties in their structure, such as various numbering conventions, bullet lists, cross-referencing, standardized wording, and so on.

New clauses or changes to existing ones are tendered as *proposals* during negotiation sessions. Proposals may involve complex logic (e.g. if proposal A from the other party, is to be accepted, then our proposals B and C must be accepted; or otherwise our proposals D and E must be accepted). Typically, proposals may be accepted by the other party (or parties) as presented, rejected outright, or returned in modified form as *counterproposals*. The parties also have *rationale* for their proposals and counter-proposals, reasoning for their rejection of proposals, and *notes* about their own positions and those of the other parties. The rationale and notes tend to be informally written and may be private to a negotiator or negotiation team, or may be partially revealed in a strategic manner to the other party as the process of making proposals proceeds. The proposing and counter-proposing process is iterative until the negotiating parties are satisfied with the complete final agreement.

1.1 Motivation

As discussed above, negotiated agreements are often critical to society, whether it be at the local, national or international level. Agreements and the negotiation process are often complex.

The process of reaching agreement at the negotiation table is often slow. Sometimes, it takes weeks, months or even years to reach agreement. Whether an agreement can be reached at all is also subject to the complexity of the subject of negotiation.

Despite the fact that most negotiation in the literature focuses on quantitative aspects, we also observed that textual communication is also key to the medium and outcome of almost all negotiations reported in the literature (Carmel et al., 1993; Cellary & Picard, 1998; G. E. Kersten & Noronha, 1999; Schoop & Quix, 2001). Thus, more documented-oriented approaches need to be considered in negotiation system design. For instance, a standardized textual agreement metamodel may spur interoperability across negotiation systems. Such a development may further promote handshaking of technology and continuity in the world of negotiation systems.

Agreements often expire on a set date, necessitating another set of negotiations to review or update the terms enshrined in the document. Such expiration makes negotiation a recurrent exercise. It might have been several years since the previous round of negotiations, and a new set of negotiators may be involved in each party. With the state-of-the-art processes employed by many negotiators today, the only basis for the next round is the final agreement of the previous round. The notes and history of rationale for the previous round tend to be either inaccessible or very hard to access, thus hindering negotiations for the new round.

To summarize: We are motivated to improve textual negotiation technology, in order to simplify it, speed it up, improve its quality, and standardize it, since negotiations are critical to the functioning of society.

1.2 Research Questions

Considering the cognitive and processing demands required to perform negotiations and writing of contracts, we will address the follow questions:

RQ1. What **processes and tools** are currently used by people negotiating qualitative (textual) agreements, and what are the strengths, weaknesses and requirements gaps in these?

RQ2. What would be an appropriate **metamodel** (abstract syntax) that would capture the structure and content of qualitative (textual) agreements and negotiations in general, enabling tools to be created that would in turn enable agreements to be better analyzed and negotiated?

RQ3. How can we **improve the process** of creating, modifying and negotiating qualitative agreements through technology?

RQ4. Why are sophisticated and **existing negotiation tools not being universally used**?

Our goal is that answers to these questions should spur innovations, including the design and development of negotiation models, theories, and systems. This will bridge important gaps in the literature, lead to practical improvements in complex real-life negotiations, and positively affect society.

1.3 Research Design

The research idea in this work originates from experience of the researcher's supervisor, Timothy Lethbridge. Click or tap here to enter text. Prof. Lethbridge participated for over 20 years in different negotiations and observed the challenges faced by the negotiation team. He developed the hypothesis that many challenges could be overcome if suitable negotiation technology were available.

As a first step towards developing such negotiation technology, the author's supervisor reflected about his frustrations and experiences at different negotiation tables, and communicated these experiences and some ideas for improvements to the author of this thesis. These ideas included some rough ideas about requirements for a negotiation tool, and an initial class diagram designed with Umple (Lethbridge et al., 2021) to describe the data needed in such a tool. The requirements ideas were later combined with extensive input from other research processes performed by the author, described below, to create the results presented in Chapter 5. Similarly, the class diagram was iteratively modified to create the metamodel presented in Chapter 7.

The initial input from the supervisor's negotiation experience required extensive additional data gathering analysis, and extension. In particular it was a single source of data from a single researcher. We therefore followed the other research processes described below.

First, we conducted a systematic literature review (discussed in Section 2.2) to understand and document the existing research into technology support for negotiation. As our research evolved, we conducted addition literature review (discussed in Section 2.3) which further deepened our knowledge.

Next, we conducted a grounded theory (GT) study (discussed in Subsection 3.1 and Chapter 4) to understand negotiators' experience with their general process, tools use by negotiators, how to improve the state of the art in textual (qualitative) negotiation, and the technology needed to support it. The result of the GT helped define baseline requirements for a metamodel of negotiation, and tool features that would effectively support negotiation.

Then, we conducted survey (discussed in Subsection 3.2 and Chapter 6) to characterize how negotiation is performed in various sectors, and to learn how a wider sample of negotiators feel about baseline requirements for negotiation software as defined by the GT results.

Building on the input from the previous steps, we used a design science approach to create the final metamodel described in Chapter 7 and a prototype tool, SoarNego, described in Chapter 8.

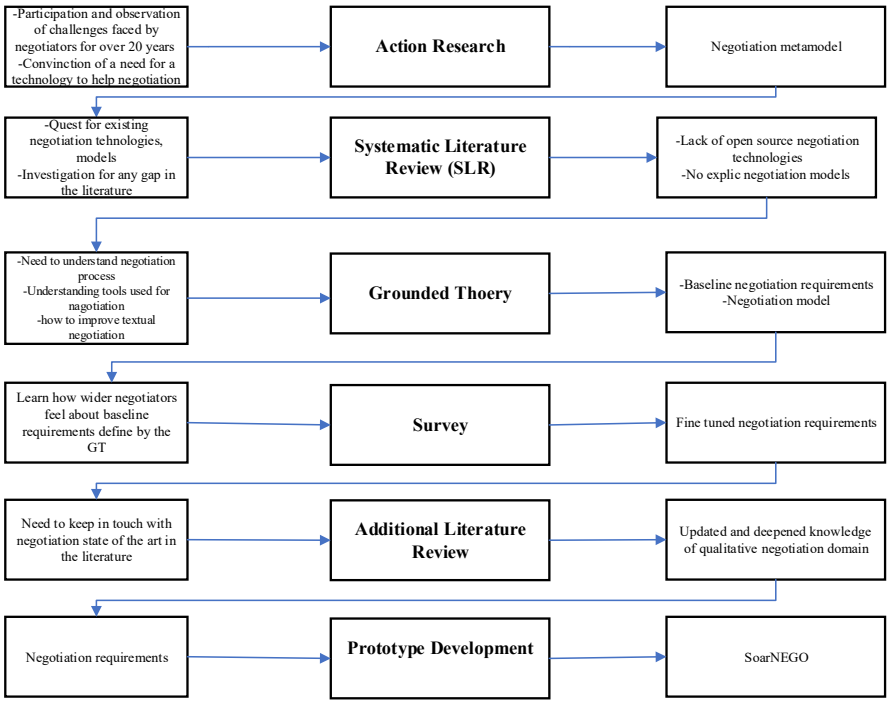

Figure 1: Research Design Framework

1.4 Paper Published so far about this Work

So far, we have published one paper from this work and there are other papers that are undergoing the process of publication. We have adapted major parts of this thesis from the content of these papers. The published paper citation is:

Emmanuel C. Ayeleso and Timothy C. Lethbridge. 2022. Requirements Analysis Using Grounded Theory: A Case Study in the Domain of Textual Negotiation Tools. In Proceedings of the 32nd Annual International Conference on Computer Science and Software Engineering (CASCON '22). IBM Corp., USA, 199–206.

1.5 Thesis Contributions

The key contributions of this thesis are listed next. Full details of these contributions are summarised at the end of the thesis in Section 9.3.

1. **Gap identification:** The field of qualitative negotiation has a lack of research and tools to support it. Evidence for this comes from our literature review as well as from the grounded theory and survey.
2. **Key observation**: Identification of the prevalence of the use of *MS-Word* as the main support tool for qualitative negotiators, despite its limitations. Evidence for this comes from negotiation information provided by the supervisor, and was further confirmed by the grounded theory and survey.
3. **Process model**: Creation of a model detailing the processes and operations in qualitative negotiation. Evidence for this is from the grounded theory.
4. **Use case model**: Definition of a comprehensive set of use cases for software support in the qualitative negotiation process. Evidence for this was generated by all the research methods.
5. **Requirements model**: Formulation of specific requirements for essential features in a qualitative negotiation tool. Evidence for this was derived from the negotiation information provided by the supervisor, grounded theory, and the model was validated through the survey results.
6. **Negotiation metamodel**: This provides insight into qualitative negotiation processes and can be utilized for generating negotiation data models or code. Evidence for this was derived from and cross-validated by all the research processes.
7. **Prototype tool**: Design and implementation of an open-source prototype tool, SoarNego, which incorporates some key features identified in the requirements. Evidence for this came from the negotiation information provided by the supervisor, grounded theory and survey, and served as validation for some of the features, and as a basis for future research.

1.6 Thesis Outline

The rest of the thesis is organised according to the following outline:

- Chapter 2: Background

Discusses the existing research efforts in the domain of negotiation systems, as well as other information needed to understand the thesis.

- Chapter 3: Research Methods
  Discusses the specific procedures that we have used to collect and analyse data during this research.

- Chapter 4: Results of the Grounded Theory Research
  Discusses the results of grounded theory conducted in this research, including the coding we developed, backed up by numerous quotations from participants.

- Chapter 5: Suggested Features for a Qualitative Negotiation Support System:
  Discusses features of a tool that will effectively support qualitative negotiations. The suggested features are part of the output of this research.

- Chapter 6: Results of The Survey
  Discusses the results of the survey we conducted to expand and validate suggested features discussed in Chapter 5.

- Chapter 7: Negotiation Metamodel
  Discusses the negotiation metamodel describing the data that would need to be captured in a fully-functional qualitative negotiation system, derived from numerous aspects of our research

- Chapter 8: SoarNego
  Discusses a qualitative negotiation support tool prototype that was developed during this research. The prototype implemented some of the features discussed in Chapter 7

- Chapter 9: Conclusion and future work
  Discussed the conclusions and the forward-thinking direction of this research.

Chapter 2 Background

In this chapter, we discuss various topics needed to understand the thesis, and research by others on which our thesis builds.

In Section 2.1, we first introduce some definitions of negotiation systems as identified by other authors.

Section 2.2 is a systematic literature review (SLR) that we undertook in 2020 to examine the literature on models and processes related to qualitative negotiation. Section 2.2.1 gives the specific SLR research questions on which we focused; these partially answer the global research questions in Section 1.3. We elected to present the SLR unchanged.

As the research developed, we uncovered a variety of other research threads relevant to our work. These are research works that our SLR search keywords did not catch, due to the facts that terminologies in the negotiation domain are not crisp, unlike in a domain like medicine. Literature about these threads will be discussed in the following sub-sections of this chapter. We cover various categories of negotiation support systems (NSSs) including those that discuss communication management and how negotiators exchange proposals and counterproposals. One particular tool, Neggoist (Schoop, 2020), is considered in both the SLR and subsequent sections, as it is still under active research.

2.1 Negotiation Systems Overview

Negotiation is a decision-making process used to reach agreement in a situation where a single party cannot take a decision alone. It can take a simple form of a dialogue between two people or multiple parties (negotiators) trying to come to terms of agreement on a subject. Sometimes, the process of reaching mutual agreement takes excessive time, or reaches a state of inconclusiveness or even deadlock. Due to the complexity, sometimes decisions are not in the best interest of the negotiating parties.

The success of the third United Nations Conference on the Law of the Sea (UNCLOS III) was one of the many experiences that buttressed the significance of using an early software tool during global-level negotiations. UNCLOS III resulted in consensus decisions by 157 states, constituting 77 groups; its negotiations took place between 1973 and 1982. The treaty produced by UNCLOS III amicably settled a number of the world's critical issues with far reaching

consequences for the participating states (Buzan, 1981). A major contributor to this success is the application of MIT's deep ocean mining model as a simulator to reconcile the wide differences of the developing and developed countries' nations of deep-sea mining. The negotiators were shown the different simulated scenarios in the context of deep-sea mining; the understanding gained from these contributed towards financial agreement reached during the negotiations (G. E. Kersten & Lai, 2008a). Subsequent to UNCLOS III, researchers realized the potential impact of models to support negotiators. This further spurred research efforts to support, aid or automate the process of negotiation.

The idea of using computer-related tools to support negotiators arises from the need to compensate for human limitations when they try to process huge amount of information that may be required during negotiation.

2.1.1 Electronic Negotiation Systems (ENSs)

The advent of the internet in the early 1980s further spurred the development of negotiation systems. Prior to the internet, such systems operated in a stand-alone manner typically on mainframe computers connected using dumb terminals. An example of such an early negotiation system is NEGO (G. E. Kersten, 1985). The internet made it possible for negotiation to be done without any constraint for the negotiators to be at the same location; Kersten and Lai called the resulting systems electronic negotiation systems (ENSs) or e-negotiation systems (Kersten and Lai 2021).

Kirsten and Lai define ENSs as internet-based software systems that are deployed through the web to electronically communicate, collaborate, coordinate, and decide the activities undertaken by the negotiators and any other stakeholder involved in a negotiation (G. E. Kersten & Lai, 2008a). Since the late 1970s, many systems have been developed to support negotiation, aid negotiators, and automate part or full negotiation process (G. E. Kersten & Lai, 2008a). Virtually all the early negotiation support systems have evolved and are now internet based. Therefore, it will be difficult to separate negotiation support systems (NSSs), without talking about e-negotiation systems as the two are now intertwined, and essentially synonymous.

Later in this thesis we will primarily use the more modern term NSS. The NSS topic is a subtopic of computer supported cooperative work (CSCW), which is a research field that focuses on understanding "how computer systems can be instrumental in reducing the complexity of

coordinative cooperative activities, individually conducted and yet interdependent" (Schmidt & Simone, 1996).

Many systems have been designed to solve complex negotiation tasks: decision making; conflict identification and resolution, search for consensus, assessment of agreement stability and equilibrium analysis (Kersten and Lai 2007). These authors divided the types of electronic negotiation systems into the following:

1. E-negotiation tables (ENT) are software systems that provide negotiators with a virtual space, a mimic of the conventional negotiation table and tools that could be used to undertake negotiation activities.
2. Negotiation Software Agents (NSA) are software tools that have the autonomy to perform negotiation activities on behalf of human negotiators or artificial principals.
3. Negotiation Agent Assistants (NAA) are software agents that supply human negotiators and/or third-parties with relevant advice, analysis, and support in the domain of negotiation.
4. Negotiation Support Systems (NSSs) are software tools that assist negotiators to reach mutually satisfactory decision/agreement by providing communication and coordination facilities. Note that this acronym later came to be used as a term that incorporated aspects of the other categories.
5. Decision Support System (DSS) for negotiations are negotiation support systems with addition of decision support capabilities.

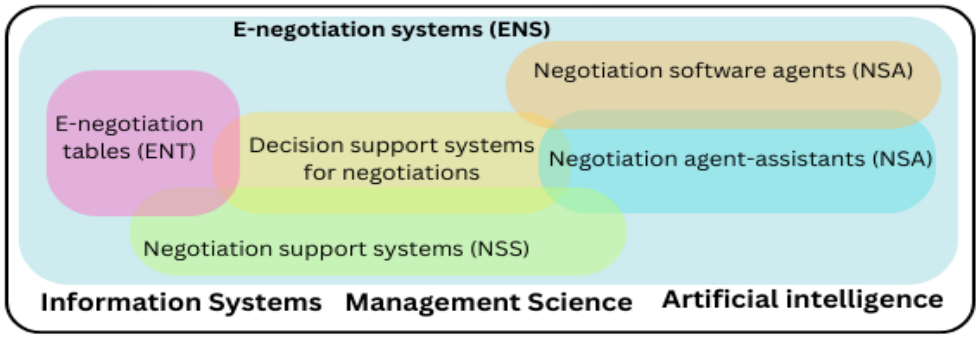

Figure 2: Software systems in negotiation facilitation, support and automation. (Adopted from G. Kersten & Lai, 2021 p. 1054)

2.2 Systematic Literature Review

We carried out a systemic literature review (SLR) in 2020 about negotiation support systems (NSSs). Our interest was to investigate the underlying models used by the existing NSSs. We also investigated user stories and models in the world of NSSs. With the aid of Mendeley and Convidence, we selected 22 relevant papers out of the 460 papers collected for review. No explicit agreement model was mentioned in the reviewed papers.

2.2.1 SLR Research Questions

For this SLR we created some research questions that specialize our global research questions that we presented in Section 1.2.

RQA: What are the underlying **models** of agreements used by negotiation support tools for creating or modifying contracts? (This will help answer RQ1, inquiring into the tools, and RQ2, inquiring into the existing models.)

RQB: What are the user stories and operational **processes** in NSSs? (This will help answer RQ1 and RQ3.)

2.2.2 Systematic Literature Review Methodology

We adopted the following stages associated with conducting the SLR. Broadly, the processes carried out can be summarised as (Kitchenham et al., 2008):

1. identification of research;
2. selection of primary studies;
3. study quality assessment;
4. data extraction;
5. data synthesis.

2.2.2.1 Search Strategy

The keywords “negotiation” and the phrase “model of agreement” formed the core keywords that were used to search the electronic databases. Words or phrases that represent their synonyms were chosen and combined with the logical expression AND and OR in conformity to

the syntax of the databases searched. We observed that negotiation-based research cut across multiple disciplines with research interest beyond the scope of our study. Some of these external disciplines are management, organizational behavior, social psychology, cognitive psychology, economics, communication studies, sociology and mathematics. This led to high false-positive search results from two of the generic databases used for this study. Therefore, we limited papers considered for selection to the following disciplines: computer science, engineering, and social sciences. We believe these disciplines are most relevant to answering the set research questions than certain other areas where negotiation is sometimes also discussed.

2.2.2.2 Databases Searched

Two generic databases and two field-specific databases were searched. The generic databases are Scopus and Web of Science. The field- specific ones are ACM Digital Library and IEEE Xplore.

2.2.2.3 Search Query Strings

There are syntax variations to the search query used on each of the electronic databases. However, the generic block search query is defined here.

("electronic-negotiation*"
OR "e-negotiation tool*"
OR "electronic contract*"
OR "bargaining system*"
OR "agreement support system*"
OR "agreement negotiation support*"
OR "contract negotiation tool*
OR "contract negotiation system*")

AND

("models of agreement*"
OR "offer*"
OR "process model*"
OR "negotiation process*")

Search fields: title, abstract, keywords.

Publication type: conference papers and journals.

Language: English only.

Subject Areas: Computer Science, Engineering, and Social Sciences

2.2.2.4 Exclusion and Inclusion Criteria

To exclude papers from the selection, we used the following criteria: works were excluded that did not focus on either negotiators' input, mediation, modeling agreements, or documentation of a contract as an outcome of the negotiation. These factors are the critical phases of negotiation in real life. The following specific exclusion criteria were also applied:

1. All works not published in the English language, which is the only language understood by the researcher.
2. All works that do not discuss a tool or model used in reaching an agreement.
3. All works that do not provide assistance such as presentation of new proposals, arguments or facts during the negotiation process.
4. All works that fully automate the process of mediation such that parties involved in the negotiation cannot flexibly re-present their proposals, arguments, or facts if the need arises.

We also snowballed peer-reviewed works not discovered initially, but that were referenced by at least two of the final selected papers, and which also met the other selection criteria.

Only one researcher, the author of this thesis, actively performed every stage of this systematic literature review process. He explored the use of tools such as Mendeley, a citation manager, and Covidence, an SLR paper screening software system, to aid the production of this systematic literature review. However, relevant stakeholders assisted by reviewing the work at various stages. In particular, a draft of the protocol was vetted and peer evaluated by a professor of software engineering who was teaching a course where writing the SLR was part of the curriculum. His constructive suggestions were used to correct and modify the initial protocol. Subsequently, the initial draft of the SLR was also subjected to peer evaluation through presentation to other students in the course. The contributions and suggestions raised by the peer evaluators were used to correct and modify the initial version of the SLR. Finally, the corrected initial version was evaluated by another graduate student with knowledge of SLRs.

2.2.2.5 Search Results

The initial search retrieved 146 papers from Scopus, 134 from Web of Science, 44 from ACM Digital Library and 136 from IEEE Xplore Digital Library. We reviewed the references of all the papers selected, and found 3 additional papers that were referenced at least 2 times, so we included these 3 papers as well.

2.2.2.6 Selection of Primary Studies

Figure 1 presents the processes that was used to prune the number of the papers eventually selected for this study. This process is adapted from (Moher et al., n.d.).

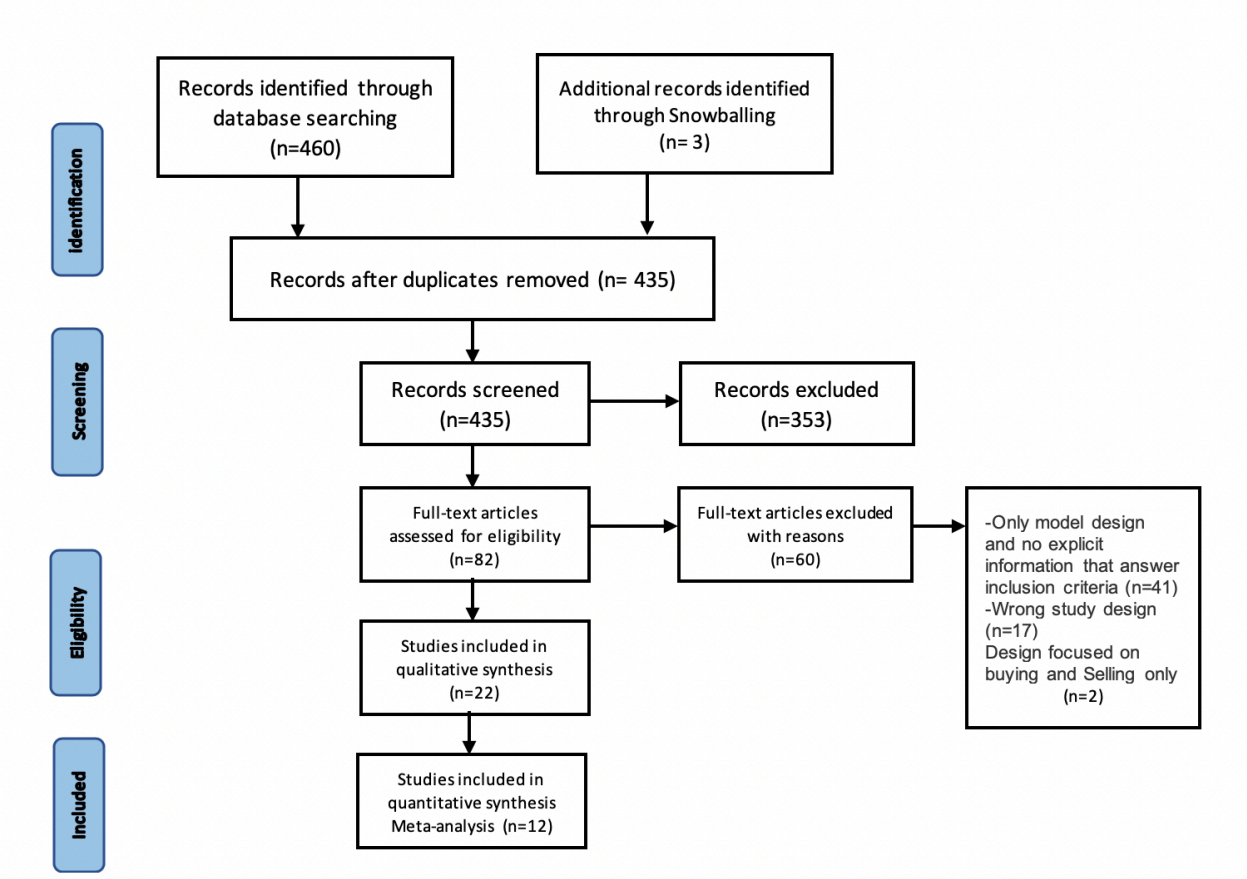

Figure 3: PRISMA flow diagram for Negotiation Support System, a Systematic Literature Review

2.2.2.7 Data Extraction

We perused the selected papers and extracted all data relevant to answering the set research questions in this study, and added this data to a spreadsheet. We performed column-based classification of data in the spreadsheet.

2.2.2.8 Synthesis

Data extracted on the created spreadsheet was examined for patterns by cross matching similarities and differences among studies. Pattern exploration was done as follows: Studies that shared common approaches/frameworks/models were grouped and tabulated together as a category in order to identify patterns within and across studies as shown in Table 1. Sections 2.2.3 and onwards discuss other issues found in the SLR.

2.2.2.9 Threats to Validity

Here we assess the risks to accuracy and correctness of data extracted during this study. The following are the threats to validity observed.

1. Non-availability of any recent literature in the problem domain. This may imply that the extracted data are stale, although it also seems likely that research in the area has simply slowed down.
2. The chosen keywords may not have covered the literature, if certain literature uses vocabulary that differs from the norm.
3. Only one researcher performed this SLR. As a result, data extracted and results may be biased.
4. Creating an SLR is a rigorous exercise with much time required and a vast number of papers to be examined to achieve success. The researcher may have missed information due the volume of data to examine.
5. The conclusion drawn from this study may not be generalizable to NSSs in general.
6. Papers published in English language only were considered. It is possible to have left out other relevant papers published in other languages.

The following mitigation strategies were adopted, such that, to the best of researcher's knowledge, all accessible studies that meet the inclusion and exclusion criteria, and that are accessible were collected for this study.

i. A protocol was written before the commencement of this review which guided the activities of the researcher.
ii. The protocol passed through peer and professor review before being employed.

iii. Technology such as Mendeley and Convidence were used to automate part of the SLR process where possible. This lightened the SLR process, making it more tractable.

iv. Only papers relevant to inclusion and exclusion criteria were selected. The researcher believes these are the papers relevant to answering the research questions.

v. Snowballing was used to track commonly referenced works that never made it to the final selection. This should help mitigate threat 2 in particular.

Table 1: Negotiation support system user stories, models, and contract formats from the SLR

Tool developed	User stories	Model used	Format of contract as NSS output	Support quantitative or qualitative negotiation	Reference
Negoisst (see discussion in Section 2.2.3.1)	Negotiator tags each message for the system to classify it as informal (green) or formal (red). Accepted formal messages constitute what makes the final contract version.	Theories of communication and formal logics	The contract is an XML document containing all attributes of interest and the messages document these attributes.	Quantitative	(Schoop et al., 2003)
SPICA (see discussion in Section 2.2.3.2)	Negotiation is preceded by a setup phase where Negotiators select pattern (bargain, auction or ballot) for the negotiation. The negotiation phase aims to assign values to properties. A verification phase does checks, e.g., whether all properties were negotiated. If OK, an electronic contract is produced and digitally signed by involved negotiators.	UML was used to model the contract instance	XML format	Quantitative	(Baccarin, Madeira, Medeiros, & der Aalst, 2011)
Negotiation assistant	Input provided by the users in the preparation stages may be edited and revised as often as needed during the negotiation phases.	Models used in economics (model name not mentioned), Rule based Decision analysis and Game theory	A stored file containing negotiators' input, negotiation exchanges and post settlements. The format of this file was not stated in the paper.	Quantitative	(Rangaswamy & Shell, 1997)
CBSS – Collective Bargaining Support System	Negotiators keep redefining their text content during negotiation.	Not mentioned	A document showing final agreement. The format of the document was not mentioned in the paper.	Qualitative	(Yuan et al., 1998)
MultiNeg	Recursive nesting as a tree of mathematically structured proposals with Requirements and Offers among more than two parties.	WS-Agreement protocol	A document stored in XML format.	Quantitative	(Klenk et al., 2012)
INSPIRE	Rated Users offers and messages are adjusted and modified iteratively during the stages of the negotiation. History of the adjustment is kept by INSPIRE.	Hybrid conjoint analysis	Presents Pareto-optimal offers that dominate the compromise between the negotiators on INSPIRE. Format of such file not mentioned in the paper.	Quantitative	(G. E. Kersten & Noronha, 1999)
ContractBot	Take input in addition to the contract template from potential buyers and sellers who submit rules that specify their constraints and preferences among alternative negotiation structures.	Rule-based model, implemented with Courteous logic Program (Expert system)	Executable contract that can be fed through inference engine to execute the terms of a deal.	Quantitative	(Reeves et al., 2002)

2.2.3 Negotiation Support System Models Found in the SLR

Models that have been deployed in NSSs can be categorised into three kinds(G. E. Kersten & Lai, 2008):

1. Models of the negotiation *problems*;
2. Models of the *negotiator*; and
3. Models of the negotiation *process*

A typical NSS may incorporate two or more categories of these models.

NSS that incorporate such models can be used in tools referred to as *intermediary analysts*, which take the wants and needs of the negotiators and generate suggestions, solutions, or scenarios from such data.

Example of such models include the MIT deep ocean mining model and IIASA RAINS model for cross-boundary air pollution (see the introductory discussion of this in Section 2.1). The MIT deep ocean mining model was a financial model of a deep-sea mining operation system (Nyhart et al., 1978) that helped UNCLOS III to reconcile widely different positions of the countries that participated. Section 2.3 further talks about this model.

Using a similar approach, IIASA (International Institute for Applied System Analysis) created a system called RAINS to support negotiations for a convention on long-range transboundary air pollution (G. E. Kersten & Lai, 2008b). Extensions of RAINS have produced models that combine formal representation of both the problem (objectives of the negotiator, wants and needs as regards the domain of negotiation) and the negotiator (their preferences) (G. E. Kersten & Lai, 2008a).

NSSs are a subtype of tools called group decision support systems; one way to sub-classify NSSs is as follows:

- **Solution-driven NSSs**: These provide solution alternatives as suggestions that could aid agreement by the negotiating parties. Such suggestions are derived from different models such as social judgement theory models, hyper game decision models, bargaining models, multi-objective linear programming, expert systems, neural networks, genetic algorithms, rule-based models, and logic models (G. E. Kersten & Lai, 2008a; Xie et al., 2003; Yuan et al., 1998). Table 1 showcases these systems and their implemented model(s).

- **Process support NSSs**: These enrich the communication channels, co-operative work and support the process of negotiation at any of the stages from preparation to the contract signing (Schoop et al., 2003; Xie et al., 2003; Yuan et al., 1998). This is the subcategory on which we are focusing in this thesis.

Much NSS research is focused on systems designed to support one of the above categories or the other. These prototypes and implementations seem not to be based on any common, precise, explicit model of the phenomenon of negotiation (Xie et al., 2003) that is relatable as regards possible roles, actions and reaction of the stakeholders involved in a negotiation and its processes.

In reference to our SLR RQA in section 2.2.1, we discovered that two NSSs found in the literature, Negoisst and SPICA explicitly showcase a negotiation model that is used in creating and modifying contracts. We discuss both next.

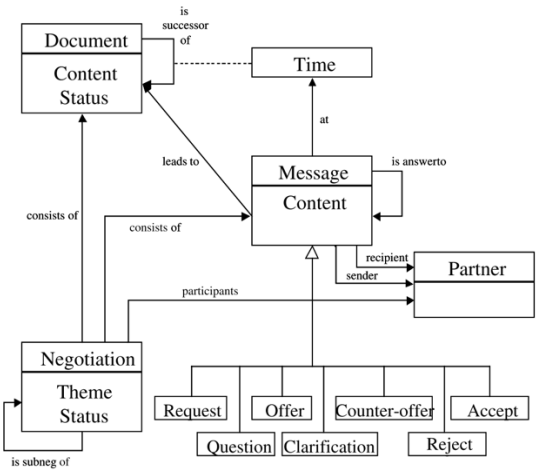

Figure 4**: Negotiation model adopted from Negoisst** (Schoop et al., 2003)

2.2.3.1 The Negoisst Contract Model

Figure **4** presents the Negoisst negotiation model. The design integrates communication data (messages exchanged by the negotiators during the synchronous or asynchronous session), and documentation data (including the contract document that evolves during the negotiation session). The model creates links between messages and the document; messages lead to a new contract version.

Negoisst is limited to only electronic interaction. Face-to-face interaction is key in negotiations, as it gives negotiators room for observing gestures, body language and way of speaking of the parties. Such non-verbal cues help provide quick feedback regarding the negotiating parties stands as regards the issues being discussed during negotiation.

Electronic interaction in a negotiation without face-to-face communication may be susceptible to ambiguity. A situation where there may be a gap between the intentions of the writer (a negotiator) and discernment of the receiver (another negotiator). To bridge such gaps, "there must be a way to transfer semantics and pragmatics to avoid unwanted ambiguities and misunderstandings" (Schoop, 2020b). Negoisst provides semantic and pragmatic enrichment as a solution to bridge such gaps by building on the theory of speech acts by Searle (1969).

Searle posits that a word or a sentence is not the smallest unit of an utterance, but a speech act. Searle further explains that each speech act consists of, "(1) the propositional content (what the utterance is about) and (2) the illocutionary force (the mode of utterance)" (Schoop, 2020a; Searle, 1969a).

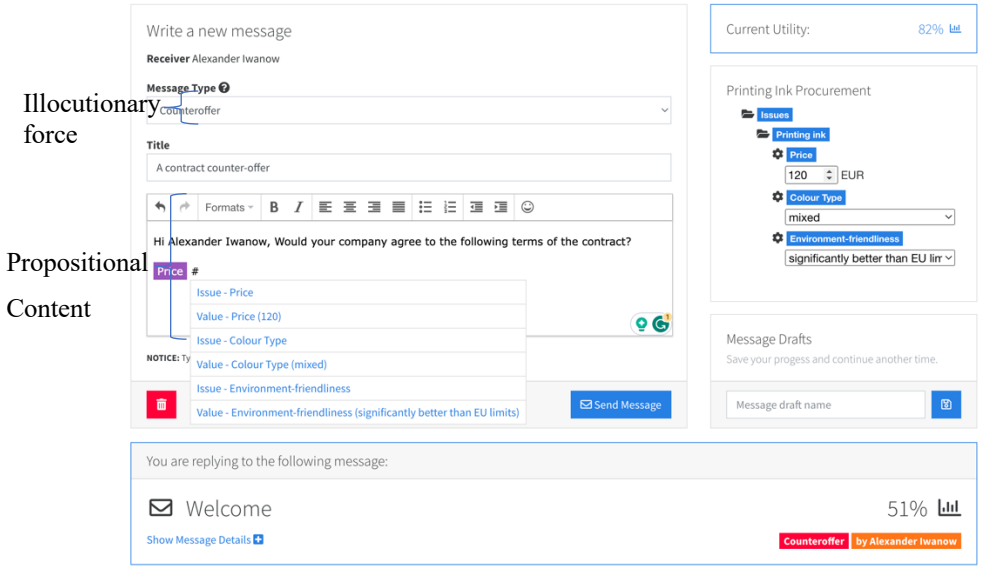

Figure 5: Message elements and composition in Negoisst (Screenshot from Negoisst interface)

To aid understanding, Negoisst semantically enriches natural language text (that is, propositional content) exchanged by negotiators using a negotiation ontology. Negoisst makes use of negotiation issues to define an agenda based on the ontology that clearly represents shared meaning to the negotiators. Texts with purple background in Figure 5 are semantically enriched

text in Negoisst. Semantically enriched texts are automatically updated and subject to the agreement of the details on the right pane of the figure (captioned "Printing Ink Procurement") where the negotiation agenda is listed. The negotiation agenda may be negotiated by changing values during negotiations, as offers or counteroffers, until agreement is reached by the negotiating parties. The utility function is also updated automatically, each time a negotiation agenda item is changed by a negotiator.

Also, to compensate for the missing cues (the illocutionary force) in written electronic negotiation, Negoisst expects negotiators to tag input messages using Searle's five illocutionary force classification (Searle, 1969b) . These are:

1. *Assertive* speech acts represent a state of affairs e.g. describing, telling, insisting, boasting, suggesting, putting, concluding.
2. *Commissive* speech acts represents the intention of the speaker to do something e.g. promising, opposing, vowing.
3. *Directive* speech acts attempt to make the addressee perform an action e.g. asking, ordering, requesting, begging, inviting, ordering.
4. *Expressive* speech acts express how speaker feels about the situation e.g. apologising, welcoming, deploring.
5. *Declarative* speech acts change the state of the world in an immediate way e.g. firing, bidding, passing sentence, excommunicating, blessing.

Negoisst defines seven message types according to the illocutionary types. The message types are attached to the contractual text content in Negoisst and are classified into five classes that Searle introduces (Schoop et al., 2003). Negoisst message types and their equivalent illocutionary points are the following (Schoop et al., 2003):

1. *Request*: Message used to express interest in buying a selected product. It is classified as directive point.
2. *Offer*: It can be used to present some product including a quantitative context such as its unit price and quantity available for sale to a potential customer. An offer is classified as a commissive point.
3. *Counteroffer*: This can be issued by either supplier or buyer to reply to a request or an offer. The counteroffer is either a directive or a commissive illocutionary point subject to

the role of the sender. For instance, a counteroffer sent by a buyer is a directive illocutionary point, whereas, if a counteroffer is sent by a supplier, it is a commissive point.

4. *Accept*: This indicates that the sender agrees with the already-exchanged message. An acceptance message is a declarative illocutionary point. Such a message marks the end of a successful negotiation and terminates the session by producing a legally binding contract.
5. *Reject*: This means the opposite of accept. It denotes disagreement and marks the end of an unsuccessful negotiation. It terminates the session without producing a contract. It is a declarative illocutionary point.
6. *Question*: This may be used during negotiation to request for clarification. It is a directive point.
7. *Clarification*: This indicates the answer to a question asked. It represents an assertive point.

Negoisst further classifies message types into either formal (request, offer, counter-offer, accept, and reject) or informal (question and clarification). Only formal messages lead to new contract versions. The content of the contract is automatically deduced from semantically-enriched message content and the message type (Schoop, 2020a). The message type implies the role of the partners (e.g delivery by the supplier, payment by the customer). The message content with semantic enrichment provides the structure, the issues, and their values in a contract version (Schoop, 2020a).

A contract version is produced after the negotiation and is strictly based on the messages exchanged by the negotiators without any room for modification once the negotiation is completed. During the negotiation, or when completed, negotiators can view the thread of contract versions deduced from formal messages exchanged during negotiation. Also, negotiators can view each issue discussed that led to each contract version.

Negoisst has been used around the world for over two decades to teach digital negotiation, as well as to enable international negotiation experiments and competitions. We observe that the Negoisst system design is focused on *quantitative* negotiation.

We will now critically review Negoisst's design. Our intention is to assess whether the design is sufficient or not to handle *qualitative* negotiation. Our key question is, what are the Negoisst components that would be useful in a system to successfully support qualitative negotiation? The following five points summarize our analysis:

1. **Qualitative negotiation-based operations outside Negoisst's defined message types:** Since Negoisst is focused on quantitative negotiation, we note some scenarios that occur in qualitative negotiation, which Negoisst's defined message types do not represent. Also, we highlight some possible recurrent operations that may follow a message type in qualitative negotiation that would generally mark the end of negotiation in Negoisst.
    i. *Request*: This Negoisst message type implies too much specificity. In a more general negotiation context, a request may not necessarily be about a particular product: it could be about a service, a request for negotiation, a request to change a rule, etc.
    ii. *Offer*: An offer may not necessarily mean a product and its value as implied by Negoisst. It could mean a proposal to change some rules, or service with terms of reference where monetary values are not applicable. The use of the word 'proposal' would be more appropriate in the context of qualitative negotiation. For instance, in a negotiation between management and a labour union, a proposal may be a modification to the wording of a rule (i.e. a clause), or a schedule of duties of staff. A proposal could contain multiple clauses. Also, a party may present multiple proposals at a time.
    iii. *Counteroffer*: The world counterproposal would be more appropriate in a qualitative negotiation context. Multiple proposals may be countered at a time by another party in a negotiation.
    iv. *Accept*: The message, 'accept' means an agreement to the terms of a proposal. Yet in qualitative negotiation there could be multiple such acceptances of different proposals in a single negotiation session. Also, there could be a conditional accept, whereby a proposal is conditionally accepted subject to the acceptance/withdrawal of other proposal(s). For instance, party ABC will accept proposal A, if party BCD will accept proposal B and ABC jointly rewrite proposal C with BCD. Negoisst cannot handle such a conditional acceptances.

v. *Reject*: This does not necessarily mark the end of a qualitative negotiation, as there could be further proposals or counterproposals and negotiations on other clauses may continue. There could be multiple rejects in a single negotiation session. It means a proposal, possibly a clause or a set of clause is/are:

- *Outrightly rejected*- If whole clauses were rejected, the proposal would be deleted from the new contract version, but where only a single clause is rejected, such a clause would be deleted while other clause(s) of the same proposal may still make it to the new contract version.
- *Partly Rejected*- The affected clause(s) may be edited or reworded by the initiator of the proposal, or the receiver of it and re-presented by either party as counterproposals. Also, there may be a collaborative editing of a rejected proposal until agreement is reached.

vi. *Withdraw (not present in Negoisst):* A party could present a proposal and later withdraw it, or the negotiating parties could jointly agree that a proposal should be removed from the agenda if they have reason to take such decision. No Negoisst message made provision for this. In fact, Negoisst messages are all treated as unidirectional. That is, each message must be acted on without reversal. There is a need for both unidirectional and multidirectional communication in qualitative negotiation. The complexity of the domain necessitates dynamism.

vii. *Question:* This could be asked regarding any communication, including conditional accept, reject, proposal, counterproposal and withdraw.

2. **Inability to deal with complex text:** Figure **4** does not have any information about the internal structure of the clauses, sub-clauses and wording differences: There is a significant difference between the structure of the version of contracts produced by Negoisst and what is typical of qualitative negotiation. Working towards an agreement using Negoisst's formal message types will possibly produce a contract version in Negoisst, whereas agreement of the wording of a message (proposal/counterproposal) may only produce new version of a single clause, that is sub-part of an agreement. As a result, there is clearly a need for mechanism to keep track of the internals of clauses and sub-clauses from one contract version to the next.

3. **Version view control**: Figure **4** is silent about visibility control of content which may not be necessary in the context of quantitative negotiation but is key in the process of qualitative negotiation. For instance, negotiators within a party may want to limit viewing of the alternative wording that they would present as a proposal only at a later stage of the negotiation; it would initially be visible to only their team members.

4. **Difficulty to tag every message exchange:** Feasibility of tagging message exchange in a text intensive negotiation will put excessive cognitive demands on negotiators.

5. **No multiplicity:** Figure **4** is limited to two-party negotiation. Live qualitative negotiations may involve more than two parties.

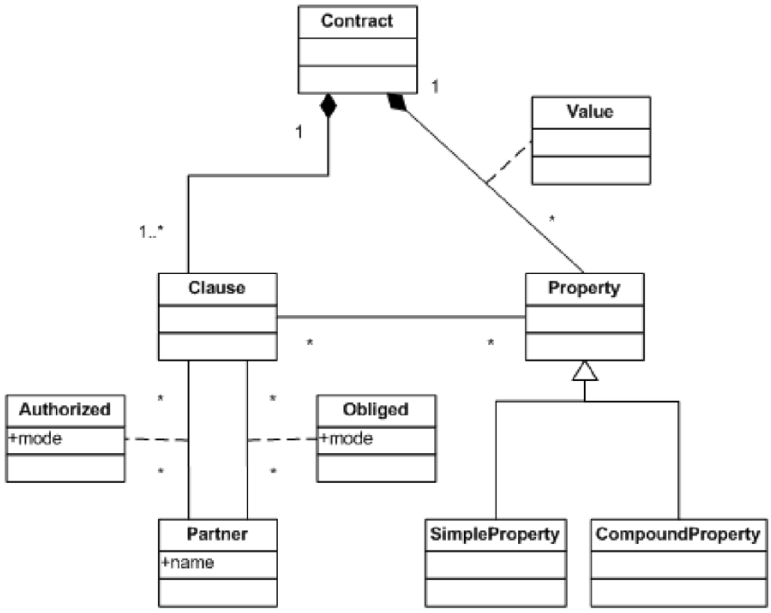

Figure 6: Contract (Adopted from (Baccarin, Madeira, Medeiros, & Van Der Aalst, 2011))

2.2.3.2 SPICA Contract Model

SPICA is another quantitative negotiation tool. Figure 6 presents the class diagram used to model contracts in SPICA. "A contract is an instance of a contract model" (Baccarin, Madeira, Medeiros, & Van Der Aalst, 2011). A *contract* consists of one or more *clauses*. A clause has one or more properties. A property is an attribute (numeric value) that is expected to be negotiated. A

property may be a *SimpleProperty* with a scalar value or *CompoundProperty* with a vector of values.

A clause may have two sets of *partners* (the negotiators), each identified by unique names. Subject to the role played during negotiation, a partner would assume the mode of either *authorised* or *obliged*. "An obliged partner must perform some action to produce the intended result. An authorised partner has the right to receive such a result" (Baccarin, Madeira, Medeiros, & Van Der Aalst, 2011). In the text of the clause, a simple property name is preceded by $ and the name of a partner is preceded by @. For example, a clause may say, "Party @OBLIGED agrees to deliver $QC number of computer systems at a unit price of $PC per system."

The model focuses on negotiation within a supply chain. The design targets assigning values to the properties in a clause. It also holds negotiating partners accountable for the proposals (obliged) or offers (authorized) make during negotiation as regards a clause. A contract's properties to be negotiated are described by means of the following message types:

1. *RFP (request for proposal)*: It indicates invitation for negotiation. For instance, Party A sending an RFP to inquire what Party B is willing to propose for the value of one or more properties. The RFP may also contain some predefined properties.
2. *Offer*: This is a commitment and answer to an RFP. A negotiator sends an offer back as a reply to an RFP. Such an offer may confirm the values of predefined properties in the RFP or propose new values for the desired properties. Sometimes, an RFP indicates restrictions; an offer must comply with such restrictions.
3. *RFI (request for information)*: This requests for information about values of properties. For instance, it may request for the possible lower and upper bounds of such values.
4. *Information*: It is the answer to an RFI and it does not imply any commitment, unlike in the situation of an offer.

SPICA supports several kinds of negotiation patterns (bargain, ballot, auction). After negotiation setup, sending negotiation messages to a participant invokes operation of a service (checking protocol rules) and delivers them to the intended receiver. The back-and-forth sending of messages during the negotiation phase aims at assigning values to the properties. SPICA logs all negotiation messages on its interface, but the paper is not explicit that a contract as document would be created after the negotiation. There is no information about the type of the messages that will lead to such a final contract.

SPICA focuses on supply-chain quantitative negotiation. It shares some commonalities and differences with Negoisst's design. We talk about these next:

Partner: Both designs use the word *partner* to represent negotiators creating and exchanging or receiving clauses/messages during negotiation.

Message Type: Both designs use message types to hold negotiators (partners) accountable for their actions during negotiation. For instance, if a negotiator presents an offer during negotiation, that negotiator is obliged to conform to this offer when the contract is finalized after negotiation.

We will revisit Negoisst further in Section 2.3.1.6.

2.2.3.3 MultiNeg Contract Model

MultiNeg (Klenk et al., 2012) departs from the Negoisst and SPICA in several ways. It explicitly focuses on enabling negotiations among *more than two* parties to arrange for a network of services where parties are obligated to provide services to other parties, in return for the fulfilment of their own requirements being fulfilled.

The paper does not provide a class diagram, but provides several instance models, such as what we have reproduced in Figure 7.

Like Negoisst and SPICA, MultiNeg employs sequences of messages to iteratively build a contract, but the contracts are much more sophisticated than in the aforementioned tools. Multineg automates the process of seeking to satisfy the various requirements of all the parties. Much of this automation is quantitative, although some is qualitative; the qualitative aspects are not explained in sufficient detail.

A lot of the discussion of MultiNeg focuses on the processes, which are represented using state diagrams (e.g. Figure 7) and there is also considerable formal mathematical reasoning regarding how the processes lead automatically to contract negotiation.

2.2.4 Results and Conclusions from the SLR

We discovered that many times the models mentioned above in Section 2.2.3 are implemented in the programming logic of the developed prototype or system and mathematical

notations are used to represent the details of the models. This makes the implemented models in NSSs to be essentially black boxes.

Table 1 presents the user stories, models, and contract formats of all reviewed NSSs; they all focus on quantitative negotiation. Input to these NSSs takes the form of structured and unstructured text with required values to be negotiated. The outputs are agreements where a compromise has been reached; they are prepared and stored as contracts binding the negotiators. We observe that inputs into such NSSs are either in XML format or are converted to it. Likewise, the output contract is commonly stored in XML. The XML format of messages exchanged during and after negotiation in the world of NSSs allows machine readability, content adjustment, message coding and reproducibility that is typical of quantitative negotiation.

2.2.4.1 Back to the SLR Research Question

We now revisit some of the research questions we originally posed in Section 2.2.1.

RQA: What are the underlying **models** of agreements used by negotiation support tools for creating or modifying contracts?

Yuan et al and indicated that these models should be based on Social Judgement Theory Models, Hypergame Decision Models, Bargaining Models, Multi-objective Linear Programming, and Expert Systems (Yuan et al., 1998). However, these types of models are not referenced explicitly in the papers relating to the tools we reviewed.

UML class diagrams dominate as the format for models used by the reviewed papers to model users, data, metadata and activities in the world of NSSs. Instances of the models tend to be transmitted using XML and contain structured or unstructured data, with *embedded variables* to be negotiated.

RQB: What are the user stories and operational **processes** in NSSs?

User input comes from either the NSS interface or a workflow communication interface. Often, these inputs are iteratively processed (rated, weighed, or classified) in order to determine suitable values of the variables; the notion of utility function(s) may be part of the process of

determining the values. Such processing commonly ends up in a document that can be called a contract.

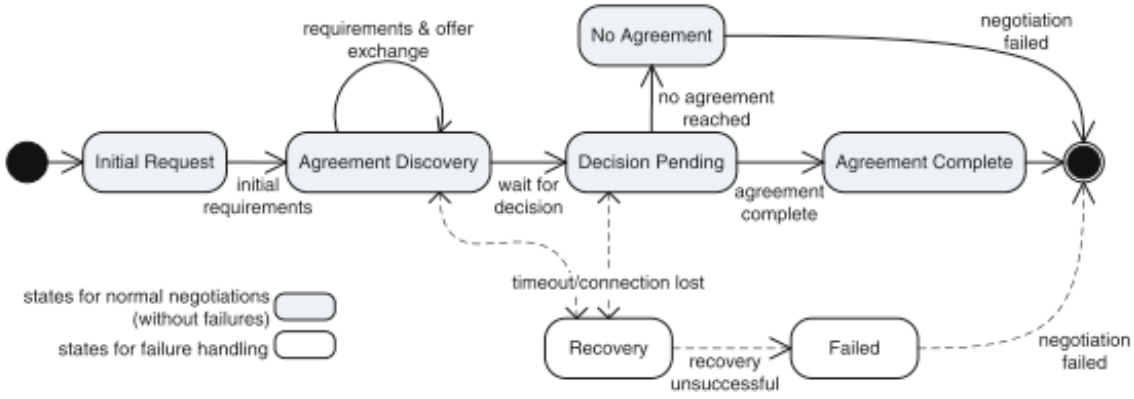

Figure 7: Message states during negotiation (Adopted from (Klenk et al., 2012)**)**

2.2.4.2 Further thoughts from the SLR

We feel it would be important to investigate the possibility of developing a NSS that can be implemented outside the scope of services and supply chains that the research efforts discussed in this SLR have been focused on. There is a need for NSS that can be used to negotiate and reach agreement between employer and labor union, and to negotiate complex issues in international relations and draft treaties between countries. The aim of our research is to focus on these.

2.3 Additional Literature Review

Following the SLR, we uncovered additional literature as our research progressed, and reviewed it in this section. We refer to this material as our additional literature review (ALR).

2.3.1 Teaching and Research Negotiation Systems

There are many negotiation systems developed as a product of research. Negotiation itself interfaces with many other disciplines such as: social science, law, sociology etc. These are fields that are driven by well-established theories to postulate concepts and their components. As a result, many of models and theories that have been investigated or implemented in negotiation systems can be traced to these fields.

In the following, we review some commonly cited negotiation systems that have been used for training, teaching, or simulation of negotiation. Surprisingly, only one, Negoisst has the evidence of being used to the present time.

2.3.1.1 MEDIATOR

MEDIATOR (Jarke et al., 1987) was one of the earliest quantitative NSS that supported negotiation by seeking consensus through exchange of information between the negotiators and, where impossible, settle for a compromise by a human mediator who is also supported by MEDIATOR. It was a data-based micro-mainframe NSS that supported negotiators and human mediator in multi-issue decision taking.

Each negotiator using MEDIATOR is empowered to use his/her own micro decision support system to exchange information that can lead towards consensus. A sequence of menus provides decision alternatives or decision criteria derived from the underlying database. Such input from the database produces a decision matrix, which is a table where rows correspond to alternatives and the columns correspond to criteria defined by the participants that express their preferences. Decision matrices from negotiating parties are merged to establish a category relation and utility function through aggregation and disaggregation of preferences by a human mediator and with the use of MEDIATOR. This enables group joint problem representation of the negotiation that can be graphically displayed.

MEDIATOR allows the human mediator to perform what-if analyses of the possible suggestion about the utility thresholds that the negotiators may have to adjust. Such suggestions are broadcast to the negotiators, who may generate additional alternatives for discussion and hopefully reach agreement on changes to make. Such changes can usually be affected without recomputing the whole decision matrix or re-accessing the mainframe databases. Suppressing a criterion may even be done by the dialog manager, without changing the internal representation at all by the human mediator. Where consensus can still not be reached, a human mediator can support reaching compromise through the use of concession-making procedures in the NSS model base (Jarke et al., 1987). The research team demonstrated the use of MEDIATOR for group car buying decisions.

2.3.1.2 Carmel's Case Study

In 1993 Carmel et al published a case study report of a labor-management qualitative negotiation by their team, in a report of action research (Carmel et al., 1993). The labor-management negotiation reported by this study took 57 hours (13 sessions) in the electronic meeting room that resulted in a contract ratified by both sides (Carmel et al., 1993). The events were in three types of sessions: strategy sessions, issue sessions and bargaining sessions; two of the researchers played the role of mediators.

The strategic sessions involved training the negotiators and their staff on both the labor and management sides how to explore the underlining issues by presenting them to the meeting room using toolsets called Electronic Brainstorming (EBS), Topic Commenter, and Vote Selection/Ranking. At the issue sessions, the two sides compiled their lists with an editor and alternately presented an issue on a screen for all to see and accompany each issue with verbal justification. At the bargaining sessions, the mediators introduced the negotiators to the ground rules:

Firstly, negotiators had to type all proposals into the workstation in the caucus room by the use of a Proposal Editor, a simple editor available to each team on a networked workstation. The paper does not specify if the proposal editor permits text markup. At this time, each proposal was only visible to members of the team of each party meeting in caucus. After a proposal was deliberated, reviewed, and adopted by the caucus, it could then be displayed on the public screen for discussion and insertion into both an electronic bargaining book and contract log.

Secondly, all verbal agreements were read aloud by the mediator, dated, marked "AGREED" and accordingly captured in the contract log.

Negotiators had access to two set of tools during the sessions:

The first set supported Win-Win techniques and included tools applied to the strategy and session issues. They helped to elicit ideas, alternatives, and solutions from the negotiators that could be organised, brainstormed, and ranked by these tools. These first set were:

1. **Electronic Brainstorming** (EBS) permitted negotiators to exchange electronic sheets of paper containing their ideas electronically within the same team. A Participant could answer a question already on the sheet (electronic file), make comments on responses already made by other participants, or include entirely new topics. This tool hence enabled brainstorming within a negotiating team.

2. **Topic Commenter** permitted participants to comment on multiple issues at a time unlike the case of electronic brainstorming that treated one issue on an electronic sheet of paper at a time. Participants were presented with multiple color cards with each card inscribed with an issue. Participants could select each card color and reply to issues. A card represents the sharing of thoughts with others via this tool.
3. **Vote Selection** was a tool that allows negotiating parties to indicate their priorities or opinions (Yes/no, multiple choice, 10-point rating scale, and ranking)

A second set of tools were used to store and make available all pieces of information (proposals, current contracts, letter of agreement) shared by the sides during the negotiation sessions. The idea is creating shared memory representation of the negotiating parties' interests. These tools are updated by the mediator.

1. The **electronic bargaining book** served as a computer-based public notebook of the negotiation and was controlled by the mediator. It contained the article list and checklist, current contracts, letters of agreement, proposals, addenda, and anything else important to the negotiation session.
2. The **contract log** was a computerized audit trail that turned into the definitive document of the negotiation. Respective proposals discussed, rephrased if need be, and agreed on were inserted into the contract log by the mediator after being read by both sides. The contract log was presented on the public screen all through the negotiation session and edited using a word processor on the mediator's workstation. Both sides: union and management representatives were given soft (diskette-based) and hard (paper-based) copies of the new contract at the end of the session.

2.3.1.3 Agora

Agora (Cellary & Picard, 1998) was another quantitative NSS tool that contributed improvements to contract negotiation technology. It focused on two things: virtual table negotiation and an integrated tool for collaborative development of the contract by the negotiators (Cellary & Picard, 1998). An Agora user, intending to negotiate with a business entity or multiple entities could create one or more virtual tables, and add the intended negotiators with the possibility of a negotiator being on a single or multiple negotiation tables. Text messages exchanged by the

negotiators could be seen by all the negotiators that join the negotiation table after invitation by the initiator of the negotiation.

Agora kept all the documents in the database and accessible to the negotiators through the network. A document stored in the central Agora database was called the global copy and the ones shared with each negotiator were called local copies. Agora sees a document being made up of paragraphs, where paragraphs are terminated by a new-line character. A paragraph can be a single sentence, a header, a title, a mathematical formula, or a figure. A negotiator can read, edit or refresh their local copy of a document and commit or cancel changes to the global copy of the document version via Agora's interface. A modified paragraph would be annotated by a color.

Where two negotiators desire to commit a paragraph modified at the same time, Agora used two possible resolutions to resolve such a conflict. The first resolution was for the negotiators to use the virtual table to exchange text messages and decide which of the modification would stay; they would wait for each other until the decision was finalized. The second possible resolution was to derive a new version of the document. Then the negotiators would continue their work in parallel and latter choose a version that would be adopted as their agreement.

The history of exchanges was permanently available through scrolling up and down in the virtual table; for each entry, the name of the negotiator, the time, subject abstracts, text messages, and attachments could all be seen. Agora drafts contracts derived from the messages exchanged by the negotiators as a linear set of paragraphs, based on the sequential order of the negotiators' messages. Agora was also enhanced with a contract version control mechanism.

We will refer to Agora again later, in Section 9.1, since it has early versions of ideas that our current research identified as important.

2.3.1.4 INSPIRE

INSPIRE (G. E. Kersten & Noronha, 1999) was a web-based quantitative tool that was developed for research and training purposes to facilitate a cross-cultural study of decision making and negotiations. INSPIRE broke the negotiation process into three phases: preparation, conduct of negotiation and post-settlement re-negotiation. INSPIRE made use of *hybrid conjoint analysis* to construct a utility function that ordered the preferences of its users during the preparation phase. In order to do so, users had to compare packages that represented possible offers in which all issues were stated. The rating assigned to each issue was seen as a component of a package's total utility.

During the conduct of the negotiation phase, users could exchange their structured offers and free-text messages via the INSPIRE interface. They were also supported by access to a graph displaying negotiation dynamics, including offers made by the parties over time with the users' rating scales. Such information permitted users to review and revise their ratings via the same interface.

Once a compromise was reached during the negotiation conduct phase, INSPIRE acted as a mediator and checked if the agreement made by the negotiating parties was in their best interest. In other words, it ensured that it was impossible to make any party better off without making some other party worse off, a situation referred to as *parato-optimality*. Negotiation ended if the compromise was efficient. Otherwise, INSPIRE computed efficient packages and displayed such information to the negotiators for selection (G. E. Kersten & Noronha, 1999).

2.3.1.5 DOC.COM

The negotiation framework called DOC.COM combines "document and communication management" for business-to-business (B2B) (Schoop & Quix, 2001). DOC.COM uses XML format to create and store negotiators' messages and the contract documents produced after a negotiation.

Negotiators enter the content of their message (proposal, argument or comments) as free text through the DOC.COM interface. Parts of the text can be marked and linked to an ontology of concepts. The ontology of concepts is organised in a hierarchical structure and represents contract elements such as products and their attributes, delivery date, etc.

Typically, the first message sent by the initiator of a negotiation acts as a medium of negotiation and as a document (possibly the first contract version). The content of such a document is reviewed, rephrased, edited and corrected as opinions or changes are suggested by the negotiators if need be. While DOC.COM stores every exchanged message, the actual document evolves until the final version is agreed on by the negotiating parties. The final contract produced by DOC.Com formalizes the obligation of the negotiating parties for post negotiation monitoring (Schoop & Quix, 2001).

2.3.1.6 More About Negoisst

We discussed Negoisst earlier in the context of the SLR, but the following provides additional information that we discovered subsequent to the SLR.

Negoisst is based on the DOC.COM framework, enhanced with communication management and cooperative document management (Schoop et al., 2003). It is still in use as of 2023; it is mostly used in the negotiation research and training community. Negoisst targets efficient and unambiguous quantitative electronic negotiations between human negotiators.

For flexibility of give-and-take chat that may go with human negotiation and the strict intentions of the negotiators, Negoisst offers two workspaces:

- The formal workspace (red area) where each message of a negotiator is marked with a message type: request, offer, counteroffer, accept, reject, question or clarification as the explicit intention of the negotiator.
- The informal workspace (green area) that can be used for discussion and preliminary exchanges.

Negoisst also performs some data analysis and checks on the existing documents and messages exchanged by the negotiators. It uses satisfiability checks to reconcile the obligation terms of messages exchanged and the general terms of the business, such as available stock level and current financial budgets.

Satisfiability checks are used to audit the feasibility of the obligatory terms expressed by the negotiators. “Negoisst offers tracing for the negotiation phase to provide the complete view of a negotiation process including the history behind an agreement” (Schoop et al., 2003). Monitor is another component that checks the obligations that arise during negotiation. For instance, monitor checks the current state of the business in terms of already-defined stock and the terms of supplies being negotiated. Documents created are stored in XML format.

2.3.2 Negotiation Software Agent

In order to develop tools and infrastructure that can support or conduct negotiations, use of software agents was introduced by a large number of research efforts into negotiation systems (Choi et al., 2005; Fei & Chen, 2007; Griffel et al., 1997). An agent is a computer program that is situated in a particular environment with capability to autonomously take actions proactively or reactively on behalf of its user (G. Kersten & Lo, 2001). Therefore, software that is able to perform

a significant part of negotiation activities on behalf of its user or another program and with some degree of autonomy is called a Negotiation Software Agent (NSA). While an NSA gives a sense of automation to negotiations, such is typically limited to well-defined and structured activities. Such condition of operation make NSAs more applicable to the negotiation systems developed, evaluated and used in e-commerce and electronic markets. As discussed previously, these fields feature a form of quantitative negotiation about goods, services and prices that favor the use of NSAs.

However, the roles of NSS and NSA in negotiation systems are complementary. Typically, NSSs are *process-oriented* and support the negotiators. Such tasks ranges from: supporting the cognitive demands of negotiation in the interest of the negotiating party; rendering communication workflow; setting negotiation protocol (such as bargaining or auctioning); generation of utility functions; and generation of options for mutual gain (G. Kersten & Lo, 2001). On the other hand, NSAs are *goal-oriented*; their object is to perform tasks and meet objectives such as: exchange of negotiating text messages on behalf of negotiators, analysis of exchanged message in the context of the configured purpose such as price of an item, and determination of best prospective buyer of the item.

Table 2: Comparison of Negotiation tools reviewed

Negotiation Tool	Features	Usage	Date	Negotiation Mode
MEDIATOR	-Building problem joint representation	Research tool	1985	Mainframe/file server accessed through communication protocol by the players
Carmel's Case Study	-Win-win techniques -Group memory -Contract log -Electronic bargaining book -Proposal editor	Research tool; seems to have been used in a single real-world negotiation context in their institution	1993	Electronic meeting room equipped with network
Agora	-Virtual negotiation table -Collaborative document writing of contract	Made accessible on the internet to interested users.	2003	Virtual table through the internet
INSPIRE	-Communication through plain messages or structured offers how the score attached to an offer helps select a good offer.	Teaching and research training	1999	Over the internet
DOC.COM	-Integration of communication and document management	-Research tool -Successfully used for negotiations between architects and different trades such as roofers, mason, and window manufacturers	2001 and improved on to develop Negoisst	Over the internet
Negoisst	-Bilateral, asynchronous negotiations (like email) -Decision support (scoring model for offers, live update during message writing, and easy to trace)	-Negotiation training in higher institutions -Negotiation experiments -Business negotiation training	2003 and until now	Virtually over the internet

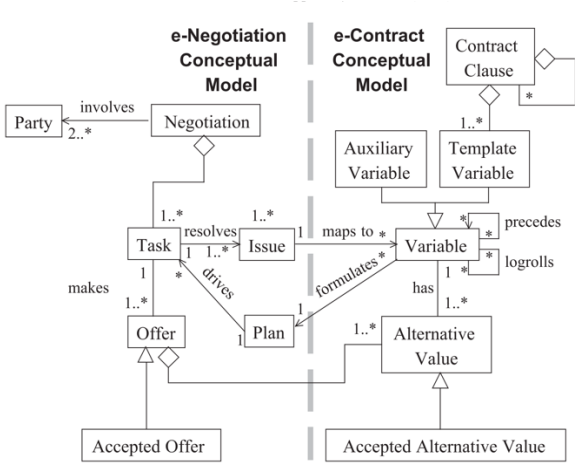

Figure 8: Conceptual model of e-negotiation and e-contract adopted from (Chiu et al., 2005)

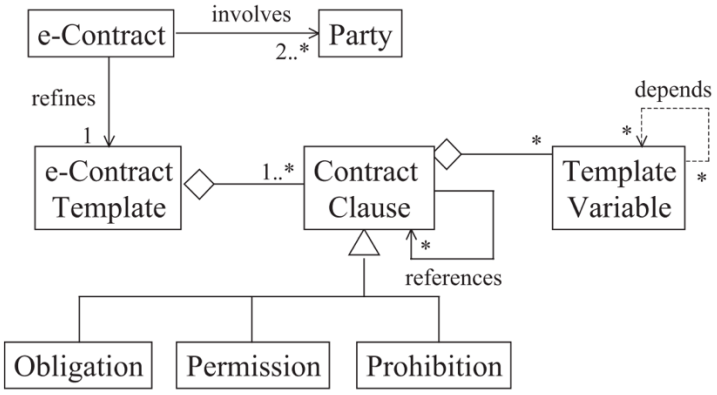

Figure 9: Meta-model of an e-Contract template adopted from (Chiu et al., 2005)

2.3.3 Negotiation Systems Models

The work of (Chiu et al., 2005) proposes e-negotiation of contracts based on the negotiation of template variables. The design attempts to avoid uncontrolled openness of issues and discussion during negotiation, so as to promote effectiveness and efficiency during negotiation. The work is focused on quantitative negotiation.

Contract templates define business interaction. "A contract template is a reference document, based on which new contract can be negotiated". A contract template consists of clauses, which are discussed during negotiations. A clause contains of one or more template variables whose values (such as product, price or quantity) are to be negotiated.

Figure 9 presents the e-contract template meta-model design of (Chiu et al., 2005). An *E-Contract* involves two or more parties. A *template* consists of one or more contract clauses. The variables in a clause correspond to issues of importance to the negotiating parties. Clauses can be of three types: *obligation, permission, or prohibition*; the parties are held accountable to what each clause represents.

Figure 8 presents another view of the model from the same author's research, focusing on the tasks during negotiation and the variables in the clauses. A task may have one of more issues to be resolved and represents some work that needs to be executed by the negotiating parties or the NSS. For instance, a party may have a task to know what it will cost a company to buy a certain quantity of fertilizer. From the instance given, such party has the task to buy fertilizers. The possible issues to resolve from such task are the quantity to buy and the amount. Issues are mapped to a set of variables. An E-negotiation plan can be formulated subject to the relationships among

variables. Offers and counteroffers may be made in a task and until an agreement is reached or otherwise. A final E-contract will only by created if the negotiation is successful.

Similarly to other tools we have examined, the design of (Chiu et al., 2005) models are focused on quantitative negotiation and their attempts to avoid arbitrary modification of text in the templates are the opposite of what is needed in qualitative negotiation. Therefore, this design cannot be used for real life complex negotiation where there are needs for openness of ideas and suggestions from the negotiators. Creating a contract from a rigid contract template will not also be suitable in the context of qualitative negotiation because, a typical qualitative contract tends to be densely textual and have few numeric values.

2.3.4 Some Commercial Tools for Negotiation

There are many commercial tools that can be used to manage contracts throughout their lifecycle, not just in the negotiation phase. They are popularly called contract lifecycle management (CLM) software. While the vendors of these systems claim their products could do sophisticated things to make life easy for their clients, some researchers believe these tools are overhyped, and their details often not explicitly described by their vendors to protect trade secrets (G. Kersten & Lai, 2021). We focus here on some of these tools that have contract negotiation capability.

We gathered limited information about the features of these tools by watching their demonstration and training videos, reading blogs and signing up for free trials where applicable. We took note of the use cases the tools support. As our research progressed, as discussed in subsequent chapters, we uncovered more details of these use cases as well as additional ones needed by negotiators. In Section 5.3 and Table 9 we will present the extent to which the commercial tools support the use cases.

2.3.5 General Comparison of Qualitative vs. Quantitative Negotiation Support Systems

As we have discussed, the electronic systems in the literature have mostly focused on business negotiations with a focus on agreeing on numeric amounts, in other words *quantitative* negotiation. There has been little research into real-life negotiations where the primary interest is settling wording in agreements, i.e. *qualitative* negotiation. This does not mean that numerical settlement is not useful in this kind of negotiation as well, but it is that other issues tend to dominate

the discussion. For instance, a typical treaty negotiation or labor-management negotiation would focus mostly on text wording.

Existing quantitative negotiation systems have, in general, three major components: communication management, document management, and decision management. The decision management module attempts to rate the various offers, that is the interests of the negotiators, and use mathematical techniques to convert such ratings to a joint co-analysis representing what might be a fair compromise that the parties might accept. It would appear that quantifying negotiators' interests in *qualitative* negotiation, or automatically suggesting compromise wording, would not be feasible, since there are too many subtleties about what text connotes, and the interests of the negotiators may be opaque. It is possible that AI could help tackle this challenge, but we leave that to future work.

Table 3 attempts to detail the difference in the intricacies of qualitative negotiation and quantitative negotiation.

Table 3: Comparison of quantitative negotiation support system and qualitative negotiation support system

	Quantitative Negotiation Support Systems	**Qualitative Negotiation Support Systems**
Components	Communication management + Document management + Decision support	Communication management + Document management + *Change management*
Input	Structured (Form) input + arbitrary text (Textual file, text entered from the system interface)	Arbitrary textual input (typically shared files)
Principle of operation	Convert negotiator's interests using utility functions and statistical methods to present joint interests of the parties	Provides mechanism to propose, accept, counter-propose, annotate and/or reject the changes to the wording of text
Context of negotiation issues	The nature of issues to negotiate are relatively structured, repetitive, and less open. That is why it is normal to be able to create a contract from a standardized template, where most of the template text will appear unchanged in the final document, Except for blanks that are filled in with values.	Negotiations have very limited structure, as negotiators are free to use arbitrary natural language; every word in documents is open to change. Templates may be used only as starting points in simple negotiations.

2.3.6 Research into Requirements and Features of NSS

Adler-Nissen and Drieschova (Adler-Nissen and Drieschova 2019) provide one of the few studies of technology used in qualitative negotiation and its potential improvement. They coined the term "track change diplomacy", to highlight that most negotiation of international treaties is done in Microsoft Word and uses tracking of changes as a key technical capability. They discuss several general categories of improvement to how technology might be able to improve negotiation, which they group into three types of affordances: shareability, visualization and

immediacy. An affordance means a capability available to users; the following are how they characterize these three categories:

1. Shareability: Features for shared authorship, drafting and editing of negotiation text by multiple people that may be across diverse locations. Also included would be email and other technology to share files among the involved actors.
2. Visualization: Features to visualize proposals and counterproposal for edits of contractual text. Examples of visualizations currently used include ~~strikethrough~~ to represent deletion; **bold**, or *italics* to represent addition, and [square brackets] to represent suggestions or highlight areas of controversy. Colouring can be used to highlight who is proposing which change. Word's track-changes feature is widely used to automatically help with such visualization.
3. Immediacy: Features enabling concerned stakeholders to undertake instant or near-instant back-and-forth, proposals (e.g. edits) and counter-proposals, or related interactions.

2.4 Summary of Observed Gaps from the Literature

The following summarizes key gaps identified in this chapter:

1. While automated approaches have dominated the negotiation systems world, the majority of the developed negotiation systems in the literature have been limited to e-business and electronic markets application (supply and service chain related). This is simply because the nature of the negotiations involved in these fields can be relatively well defined and structured.
2. There is no evidence of any standardized textual agreement model that could be used across negotiation systems for the purpose of interoperability and other possible demands of successful qualitative negotiation.
3. Many of the reported models and theories in the literature are borrowed and postulated by those outside the computer science field and were implemented in the programming logic of the systems. This makes it difficult to clearly state the model(s) and procedures used by the existing negotiation systems due to the opacity in the description by the literature.

4. There is no negotiation system in the literature that focuses on domain-specific challenges found in a qualitative negotiations such as for treaties and for labor union and management negotiations.

Chapter 3 Research Methods

In this chapter, we describe the research methods we used and explain the rationale behind the methodological choices we took.

As explained in the introduction, the purpose of our research is to have in-depth understanding of technology used in support of qualitative negotiation and to contribute to improvement of such technology. The previous chapter presented our study of previous technology. Our next steps involved studying the process and the tools used by the negotiators during negotiation, to design a negotiation metamodel, and to design a prototype negotiation tool. The methods for these steps are described in this chapter, and the results are presented in subsequent chapters.

Our research ideas originate from the frustration experienced by the researcher's supervisor and other negotiators he has been working with at various negotiation tables over many years. He sought to bring together his reflections on the negotiation experiences to pursue a practical solution to address various challenges he recorded. Negotiation information provided an informal notes by the supervisor therefore forms one of the inputs to our work.

Our first step was the systematic literature review described in Section 2.2. The outcome of the SLR (Section 2.2.4) revealed in part that qualitative NSS models tend to be implemented in the programming logic of the system, making them black boxes that cannot be analysed easily. Very few papers describe their implemented models in narrative form or with diagrams.

We followed the SLR with additional literature review (Section 2.3) to increase our depth of understanding of existing research in the domain. This further reconfirmed that very few research efforts have focused on qualitative negotiation that is our focus in this research. We described gaps in the literature in Section 2.3.6.

To fill in the gaps identified in Section 2.3.6 and to more completely answer the global research questions listed in Section 1.2, we decided to follow an *exploratory mixed-methods* research design. "Exploratory sequential is a three-phase study in which a researcher works from the constructivist principle" (Dawadi et al., 2021). This research design posits collection and analysis of qualitative data, then quantitative data collection and analysis (Dawadi et al., 2021; Schoonenboom & Johnson, 2017; Shorten & Smith, 2017). We adopted a connecting data integration approach, whereby our qualitative data analysis results in one phase define our

quantitative data collection in the following phase. Our choice of a mixed method allows us to combine at least one qualitative and at least one quantitative research approach for us to gain greater breadth and depth of understanding of the textual negotiation domain (Schoonenboom & Johnson, 2017) and to have the methods cross-validate each other to some extent. The combination of research methods consumes more time but increases confidence in the results.

As our major qualitative phase, we followed the grounded theory (GT) approach for data collection and analysis. We used semi-structured interviews to gather data from real-life negotiators. Our goal was to understand the subtle details of negotiation processes, the roles played by negotiators, and the requirements that should or must be present in a tool to effectively support negotiation. The exercise helped us to gain more insight into the negotiation process, common recurrent type of operations required during negotiation, limitations posed by the available support tools that are used by the negotiators and their general frustration during negotiation. The researcher's supervisor's experience at the negotiation table, the SLR as well as additional literature review conducted earlier, guided the interview questions.

As our major quantitative phase, we conducted a survey. A larger number of negotiators participated in our survey. The survey questions were defined by the responses received during the interviews and the results of the GT analysis.

In parallel with the above two phases, as well as subsequent to them, we used a *design science* approach to develop a comprehensive metamodel for the domain as well as a prototype solution to the requirements the above phases brought to light (Hevner & Chatterjee, 2010).

We applied and received ethical approval for both the interview and the survey from the Research Ethics Board of the University of Ottawa. We also sought and received consent of every negotiator that participated in either the interview or the survey. We protect the anonymity confidentiality of all the data gathered from both interviews and the survey.

In the next two subsections, we describe qualitative (Section 3.1) and quantitative (Section 3.2) research methods in detail. We will explain the procedures we followed to collect and analyse data. Then, in Section 3.3, we describe how we used the results from the other research methods to design our metamodel for qualitative negotiation and our prototype negotiation technology.

3.1 Our Primary Qualitative Research: Grounded Theory

We adopted grounded theory (GT) to understand negotiators' experience with their general process and their tools. Grounded theory is a qualitative research method that enables studying a particular phenomenon and discovering new theories from collection and analysis of real-world data (Adolph et al., 2012; Coleman & O'Connor, 2007a). GT is a qualitative research method based on induction and is unlike common hypothetico-deductive research (Stol et al., 2016). It uses a systematic set of procedures to inductively uncover theory that is grounded in a phenomenon (Halaweh, 2012). The idea is to discover the concepts and hypotheses that are relevant in the domain of study (Glaser & Strauss, 1967; Stol et al., 2016). GT is a deep dive into the data systematically collected in the research domain, rigorously analysed, and synthesised to uncover the theories buried in the research area. GT was invented by social scientists and initially applied to social phenomenon in the late 1960s (Glaser & Strauss, 1967).

GT has been applied across many other fields, including medical (Leyva-Moral et al., 2021), education (Chun Tie et al., 2019), and nursing (Ohta et al., 2021). GT has also been used to study human and social aspects in software engineering in various studies (Hoda, 2022). Topics of such studies include understanding software process improvement (Coleman & O'Connor, 2007a), how people manage the process of software development (Adolph et al., 2012), roles of software product managers (Maglyas et al., 2013), how agile teams organize themselves (Hoda et al., 2013), agile architecture (Sedano et al., 2017; Waterman et al., 2015), identifying different types of waste in software development(Masood et al., 2022; Sedano et al., 2017), and variations in the real world scrum practice (Masood et al., 2022).

GT is most suitable where there are no strongly defined research questions; this is applicable in our case (Hoda & Noble, 2017). In GT, the iterative and rigorous data collection and analysis processes answer the how and why related to the emergent theories and themes of the subject of study.

We conducted ten semi-structured interviews to gather data. We talked to people involved in various types of negotiation, including labor-management negotiations (both sides) and trade negotiations. We attempted to interview people who use special-purpose commercial negotiation tools, but could not find any. Although we thought this lack might bias our research, it probably did not: we heard that negotiators network with many colleagues, and that use of such special-

purpose tools is not only rare, but that most negotiators in the domains we studied have not even heard that special-purpose tools exist.

3.1.1 Variants of Grounded Theory

There are at least three main streams of grounded theory: Glaser's GT (classic or Glaserian GT) (Glaser & Strauss, 2017); Strauss and Corbin's GT (Straussian GT) (Corbin & Strauss, 2012); and Charmaz's constructivist GT (Charmaz, 2006). We adapted the Straussian GT variant for the following reasons:

1. It allows the researcher to have prior knowledge of the domain to help interpret and analyze data. In our case, we had conducted a literature review and have negotiation experience. Classic GT, on the other hand, would have insisted that the theory arise only from analysis of the observational data.
2. Our objective is to holistically understand qualitative negotiation and how it might be better supported by software. We believe triangulating our conclusions from multiple sources of information will give the best results; this aligns with pragmatism and symbolic interactionism, following the philosophical influence of Straussian GT (Kenny & Fourie, 2015).
3. Charmaz's constructivist GT supports the researcher constructing rather than both discovering and constructing.

3.1.2 Obtaining the Set of Participants

We followed a purposive sampling method to find participants. We sought people with experience in labor negotiation on both union and management sides, negotiation of inter-governmental agreements on topics such as trade and climate change, negotiation among other public-sector organizations, commercial negotiation as both buyer and vendor, and negotiation of legal agreements not included above, such as among litigants, for family law, or for intellectual property protection. We did not end up finding willing participants for the GT to *uniformly* cover each of these groups; however, we increased coverage through a survey (see Section 3.2), and discovered that we did not need to go back to add further GT interviews since the survey confirmed the GT results.

Table 4: Interview sessions

P#	Negotiation Role	Negotiation Area	Duration (min)
P1	Chief Negotiator	Labor (Management side)	36:10
P2	Chief Negotiator	International Trade	51:56
P3	Chief Negotiator	Labor (Union side)	65:59
P4	Negotiator	Labor (Union side)	54:54
P5	Negotiator	Labor (Union side)	53:54
P6	Chief Negotiator	Labor (Management side)	48:06
P7	Chief Negotiator	Labor (Management side)	49:10
P8	Chief Negotiator	Public sector (Buyer side)	37:49
P9	Negotiator	Labor (Union side)	32:38
P10	Negotiator	Commercial (Vendor side)	48:32

Negotiators were contacted by email, WhatsApp, and LinkedIn InMail messaging. We gathered some participants' email addresses by searching for potential negotiators on websites of universities in Canada, relevant Canadian government departments, and treaty-producing international organizations. We used keywords: "Negotiator", "Lead Negotiator", "Chief Negotiator" and "Collective Agreement" to search in the mentioned organizations' websites. We also contacted personal contacts in Canada and Nigeria. We also snowballed by asking participants to share our information sheet with negotiator colleagues within and outside their organization.

We did not initially set an upper bound on the number of interviews, since the GT process requires that data gathering ends when saturation is reached, i.e., that no new relevant information is being found that would substantively alter the theory being developed.

3.1.3 Interview Process

Participants were briefed that discussion of *specific* negotiations would be completely avoided during the session, so as not to reveal confidential information. We emphasized that we were only interested in the general process of negotiation, and to learn how negotiation might be better supported by software.

Virtual interviews with negotiators were conducted on Zoom and took an average of 47 minutes. Table 4 presents interview session data. All but one participant was from Canada, the other was from Nigeria. We have not revealed which is which to preserve anonymity. Each interview was recorded, and interviewers (researcher and his supervisor) took live notes during

each session. Questions were asked from a pre-prepared list (Appendix 1), derived from gaps noted in our SLR, additional literature review, and by the supervisor's experience at the negotiation table. Interviewees deviated from the prepared questions when new or interesting points were raised. The questions were adapted for subsequent interviews as new themes emerged. The questions were not sent to the participants ahead of the interviews.

3.1.4 Theoretical Sampling

"Theoretical sampling is seeking and collecting pertinent data to elaborate and refine categories in the emerging theory" (Charmaz, 2006). We drafted our initial set of interview questions that seek to answer the global research questions posed in Section 1.2 and the gaps from the literature listed in Section 2.3.6.

We conducted the first set of our interviews with negotiators selected randomly from our initial contacts. We used the emerged categories from the data collected in the early stage of our interviews to refine the interview questions, as well as to decide on type of negotiators that we would seek next to interview.

Appendix 1 lists two sets of our interview questions rubric. Such refinement of interview questions and seeking negotiators that would address the emerged categories at every stage of the interview process is considered best practice, as expressed by Charmaz: "Initial sampling in grounded theory is where you start, whereas theoretical sampling directs you where you go" (Charmaz, 2006).

As the stages of our interviews continued, iteratively, we confirmed existing concepts or categories that we had noted before and added emerging ones. At the same time, we were constantly comparing the analysed data with aspects of the emerging theory (Coleman & O'Connor, 2007a). We stopped the interviewing at a stage when there were no new concepts in the data collected during the final interviews. This is a state known as *theoretical saturation*.

3.1.5 Transcription of Video

Transcription of the recorded interview videos was performed manually by iteratively watching them. Only the relevant subject matter was transcribed; material such as the exchange of pleasantries was ignored. Notes taken by the researchers during the interviews were used to cross check the manual transcriptions where needed.

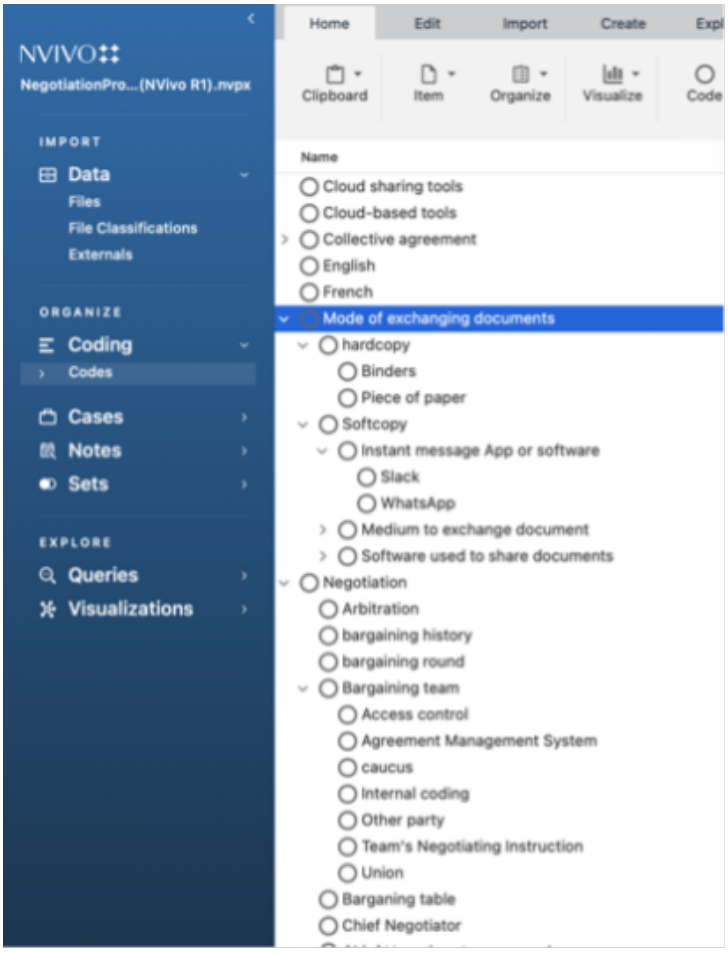

Figure 10: Part of Nvivo screenshot of low-level negotiation coding performed in the open coding step

3.1.6 Open Coding

We used Nvivo, a qualitative data analysis software tool, to label line-by-line in the data gathered, various concepts as thematic codes (Hoda, 2022). Concepts were identified as "incidents" and their associated "properties" (Stol et al., 2016). This conforms with the GT approach of deeply diving into why and how a set of activities or concepts arise in a domain of study. We continuously compared and discovered the relationships among codes to create new ones or sub-categorize them. For a typical code, we attempted to answer the following questions:

- How: How does it relate to a series of events, actions, reactions, components that make up a theme (categorized codes, operations, tools)? We categorized codes based on their shared themes and relationship with another themes.
- Why: Why is it important to the process of negotiation and the negotiators?

The researcher took the lead in the coding, with the supervisor cross-checking and improving the results. The supervisor participated in all but one of the interviews, but the author of this thesis asked most of the questions, except for in the first couple of interviews.

As interviews were analyzed, we reviewed our interview questions in line with the emerging theory, and particularly regarding the very prominent theme of history leveraging, the need for negotiators to be able to find and compare earlier versions of clauses, as well to find and search in notes previously taken and other background information.

Figure 10 shows some of the codes created and the hierarchical structure of their sub-codes.

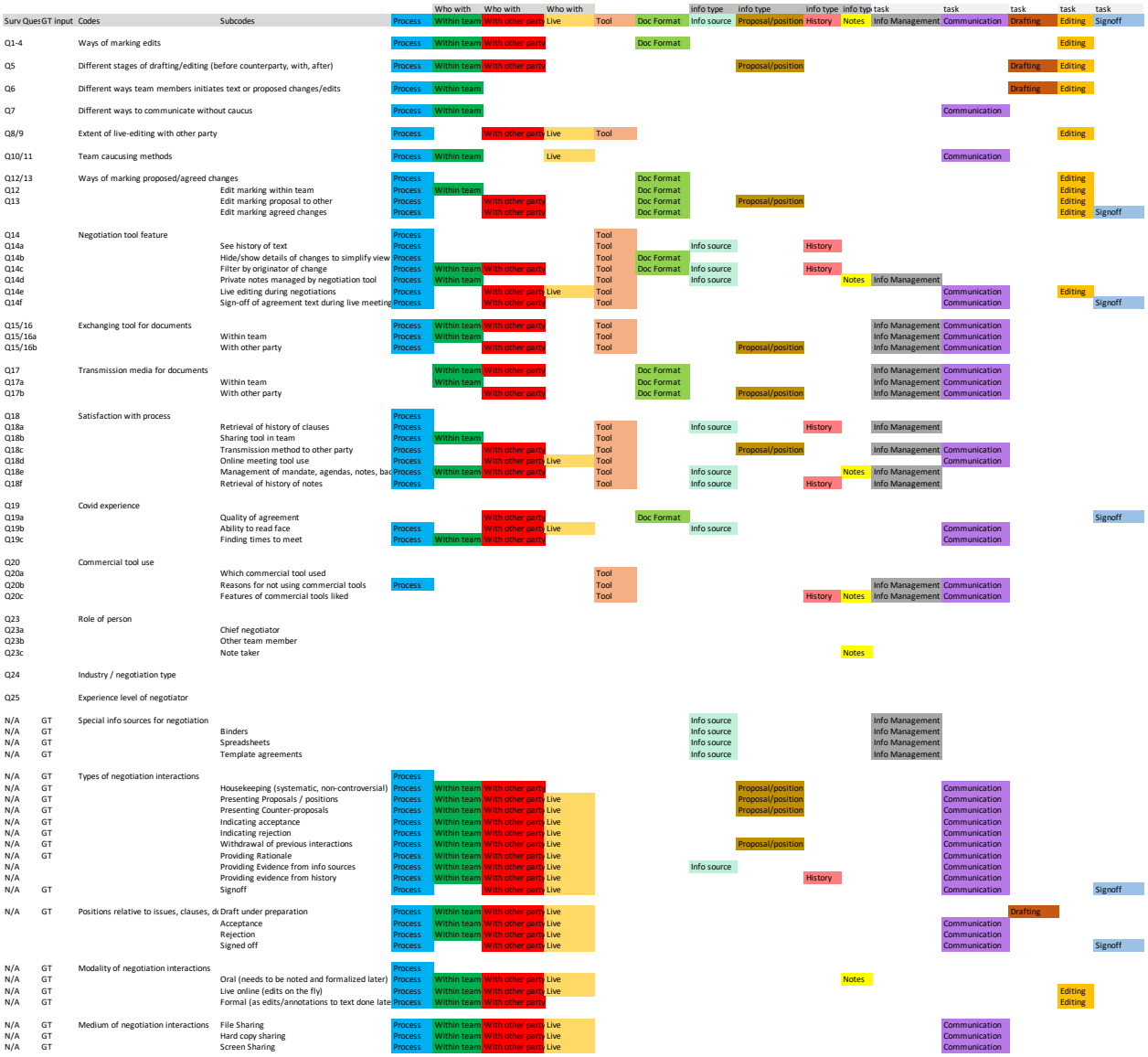

Figure 11: Part of the larger MS-Excel screenshot of codes categorized in two dimensions during axial coding

3.1.7 Axial Coding

We used a spreadsheet with color-coded tags in rows and columns, partly shown in Figure 11, to identify the context, condition, action/interaction strategies and consequences between categories. This deepened our insight into the themes. Prior knowledge of the supervisor and narrative from the interview sessions further guided the arrangement of the themes across rows and columns. As with open coding, the researcher took the lead in this stage of coding. Then, to improve validity, the supervisor iterated refining the results.

3.1.8 Selective Coding

At a brainstorming session involving both researcher and the supervisor, an initial version of the sketch of connections between categorized themes was prepared on a whiteboard as a model that ties together almost all the emergent themes in the data gathered. We then populated each category using the sub-codes from Figure 10 and Figure 11. After iteratively comparing the data, codes, and categories, careful analysis of the themes, and evolution of the sketch version, a model (Figure 13 showcased in Chapter 4) emerged as a model of negotiation.

3.1.9 Threats to Validity

In this section we examine three sources of potential threats to the validity of our grounded theory study. The analysis is guided in part by a study of GT trustworthiness by Sikolia et al (Sikolia et al., 2013).

3.1.9.1 Construct Validity of our Grounded Theory Research

Construct validity considers whether we are measuring the right thing and whether we are using reasonable representations.

In grounded theory, the data gathering instruments include the recording, the transcription of that recording, and the interpretation of what was said. The first two of these are unlikely to give rise to threats. The third (interpretation) might be a threat if we did not understand clearly what the interviewees were saying. This was mitigated by having both authors present for most interviews and double-check the interpretation, particularly the coding. Another mitigation was that we received a lot of the same information from multiple interviews.

3.1.9.2 Internal validity of our Grounded Theory Research

Internal validity is also called *credibility* by Sikolia et al. It asks the question, is it credible that the theory (model and requirements) actually arises from the data (what the interviewees said) or is it possible instead that we have somehow created results that are biased in some way.

Possible such threats include the bias of the authors arising from their prior experience, leading to results that cannot be justified.

We addressed this in the several ways, as suggested by Sikolia (Sikolia et al., 2013).

- **Prolonged engagement**: Our interviews lasted 47 minutes on average and went into depth.
- **Triangulation**: Our data sources included the interviews, the past experience of the author's supervisor (AR), and a survey that corroborates our conclusions.
- **Saturation**: We kept interviewing and adjusting our models until we concluded that we were not obtaining new information.
- **Use of participant words**: We included many quotations from the participants in our model, and phrases from their words in the requirements.

3.1.9.3 External validity of our Grounded Theory Research

External validity is also called *transferability* by Sikolia. It considers generalizability; in other words, can we reasonably argue that these results are likely to apply outside the context of this study such as to all types of negotiations.

The main threat here is that we focused mostly on labor-management negotiations, and only a little on other types. Also, we did not manage to talk to people using commercial tools, or people in a wider set of countries (beyond Canada and Nigeria). The results should therefore be considered more relevant to the context we focused on, and further studies should be done to broaden the coverage. Indeed, in our follow-up survey (discussed in Section 3.2), we did contact people in some additional contexts, such as those who had used commercial tools. The results of the follow-up study are consistent with the grounded theory.

3.1.10 Requirement Analysis Using Grounded Theory

We used the grounded theory results as a basis for much of the subsequent work in this thesis. In particular we used as a key tool for requirements engineering, to define a metamodel, use cases, needed features and our prototype SoarNego tool. The following justifies this.

Chakraborty and Dehlinger (2009) proposed grounded theory in requirement engineering to interpret the requirement for a proposed enterprise software system. Typically, the qualitative data used for requirement elicitation is obtained from interviews, meeting minutes, observations, and other processes. "Such data is then analysed and converted to a semi-formal description of the proposed software systems and provide the basis for its design and construction… There is a lack of systematic procedures within requirements engineering that enable the bridging between qualitative data and the final description of the system" (Chakraborty & Dehlinger, 2009) . This leads to reduced traceability between the source data (elicitation of requirements from qualitative data) and the final proposed model. "The emphasis has focused on the representation of the systems. (e.g. UML) rather than the process by which they may be discovered and elaborated from the available qualitative data" (Chakraborty & Dehlinger, 2009).

The work of Halaweh (2012) further emphasizes that applying grounded theory to requirements analysis helps to solve the following challenges facing the existing requirements analysis methods:

1. Reliance on formal notations and modelling that make it difficult for end users to validate and review the requirements.
2. There can be "poor communication and understanding between the development team members as it is not clear how the outcome produced by the requirement gathering team can be used by the others like designers and programmers" (Halaweh, 2012) .

3.2 Our Quantitative Research: The Survey

In our survey we reached out to a broader set of negotiators than our grounded theory (GT) work; this helped us to further refine and validate our GT results. We formulated the questions based on our GT results. We asked the respondents about the extent to which they would like certain features (which we describe in detail in Chapter 5) to be available, such as the ability to explore and compare previous agreement versions, and notes taken during negotiation. We also

asked about satisfaction with current processes, as well as how negotiators prefer to mark-up text and to collaborate in various ways. All the evidence from the survey will help negotiation tool designers produce better software. Such software will enable negotiators to visualize and share information more rapidly, hopefully giving rise less expensively to better agreements.

3.2.1 Participants Recruitment

We created the survey on SurveyMonkey, asked a peer student to review it to make a improvements, and then piloted it with a few people to further improve question clarity. We created unique collector links for each person or group we reached out to, so we could send follow-up messages to people or groups from which we did not receive a response within a couple of weeks.

We reached out via emails and LinkedIn InMails to roughly 111 and 119 prospective participants respectively (230 in total). We also reached out to the 10 people that had previously participated in our grounded theory research interviews. To search for additional negotiators, we used keywords: "Lead Negotiator", "Chief Negotiator", "Principal Negotiator" or "Negotiator" in websites of law firms, universities, government departments, treaty-producing organizations, and the LinkedIn Sales Navigator. The author also shared an advertising post of the survey link on his LinkedIn wall. In an attempt to give the survey more publicity, one participant assisted us by sharing our post on his LinkedIn account where he has over 11,000 potential connections.

We did not record email addresses of the respondents, and once an individual collector link was used, we deleted it from our records, to enhance privacy. We received 36 responses; this is roughly a 15% response rate considering only our direct contacts. Since many contacts would have been indirect (people viewing posts) the real response rate was clearly considerably lower.

3.2.2 Survey Questions

The survey featured 26 questions. The full set of questions are listed in Appendix 2.

The median time spent to respond was 12 minutes. Two of the survey questions (numbers 16 asking about importance of features and 20 asking about satisfaction with current process) were mandatory.

The following describes the question categories:

a. **Ways of marking contractual text edits** (Questions 1-4, and 12-13): Contractual text is subject to editing while parties are negotiating. The editing does not necessarily take place in

a live negotiation session. Negotiators use word-processor track-changes features as well as annotation features such as bolding, italicizing, underlining, square bracket, and appending asterisks to mark their proposed or counter-proposed edits. Since all negotiators we have been in contact with use Microsoft Word, we wanted to study how negotiators mark edits both when discussing privately within one party's negotiation team, and when presenting proposals to the other party. The objective of these questions is so that any tool we later create can mimic negotiator's preferred approaches. We present results in Section 6.5.

b. **Drafting or editing of contractual text** (Questions 5-6): We wanted to know at what stage of the negotiation is new contractual text drafted, or the existing text edited and how. Such knowledge will help to know another part of the requirements that must be met to effectively support negotiation. For instance, does a party most commonly prepare draft text of proposed wording before the first meeting with the other party, after the initial meeting or during the ongoing negotiation? And how does the party drafting the initial version of contractual text or editing the existing one go about it: do team members collaboratively do the drafting in person or electronically with the aid of technology? Responses to these questions, and the questions in the next three groups are discussed under the heading 'Patterns of Collaboration' in Section 6.6

c. **Different ways to communicate without caucus** (Question 7): There may be need for team members to communicate privately while in a live negotiation session. We wanted to know how such communication is disseminated without need for a separate caucus meeting, such as by text message.

d. **Extent of major or minor live editing with the other party** (Questions 8-9): There may be a desire for parties in a negotiation to jointly edit the contractual text in a live negotiation session. We wanted to know the extent to which this happens for both major changes (e.g. adding entire clauses) or minor changes (e.g. changing a clause wording from "shall" to "should").

e. **Team caucusing methods** (Questions 10-11): Any team member can call for a private caucus meeting at any stage of the negotiation. At caucus, the direction the team will pursue or the positions they will take on various issues are discussed and agreed among members. Pre-pandemic, negotiation was mostly in-person and caucus meeting used to be called by a party outside of the negotiation room. With the prevalence of virtual negotiation during and after the

pandemic, we wanted to know the extent to which features such a breakout rooms, available in software used for virtual negotiation, has changed the mode of caucus meetings.

f. **Negotiation tool features** (Question 14): Our grounded theory study uncovered some features that negotiators desire to assist them in the process of negotiation. We wanted to validate need for these features by a larger negotiator sample. We are also interested in the order of importance of these features to negotiators and whether there are some they might dislike. We discuss the results of this in great detail in Section 6.2, although the features themselves are discussed in Chapter 5, since they were drafted prior to the survey based on the grounded theory data – the survey focused on ascertaining the importance of each of the features.

g. **Sharing and transmitting of documents** (Questions 15-17): We wanted to find out what tools are being used to share material (e.g. Google Drive, Dropbox), transmit material and messages (e.g. email, instant messaging, physically), and what file forms are used (e.g. Word, pdf, paper, etc.). The answers to these questions are briefly summarized in Section 6.5.

h. **Satisfaction with the process** (Question 18): Negotiators make use of a generic text processor, Microsoft Word to negotiate. We wanted to know their satisfaction while carrying out various negotiation-specific tasks following this current process. The objective of the question is to help us focus on the tasks where satisfaction is lower. We discuss results in Section 6.3.

i. **Covid Experience** (Question 19): Covid-19 brough about transition of negotiation from in-person to virtual. We wanted to know how such transition has affected the speed and quality of agreement and general negotiation process. Results appear in Section 6.4. This is not core to the research in this thesis, but we took the opportunity to seek this timely information with little extra effort.

j. **Commercial Tools** (Questions 20-22): There are specialized commercially available negotiation software on the market. Our grounded theory result suggests few are using them. We wanted to validate this result or to see if any of our broader sample of negotiators had used them in the past, is aware of them, or is actively using them. We also wanted to know what makes people avoid such tools, and (if they use such tools) what features they find most attractive. Our findings about this can be found in Section 6.7.

k. **Demographics** (Questions 23-25): We asked about their experience level, and the roles and domains in which they have experience. We discuss the results of this in Section 6.1. We asked the demographics questions at the end since, although they are useful for categorizing

responses, it was more important to ensure respondents answered the earlier questions, since some would be likely to quit before the end of the survey.

l. **Open Ended** (Question 26): We gave the respondents an opportunity to add anything else they would like to contribute.

3.2.3 Survey Data

We extracted the responses from SurveyMonkey into an Excel spreadsheet, and preprocessed the data by separating certain un-needed data such as respondent ID. For certain questions that use Likert scales, we had asked SurveyMonkey to provide us with the responses as numeric values. We then used Excel's 'Filter' capability and 'subtotal' functions to enable us to filter according to different criteria (such as only self-identified expert negotiators) to have better insight into data. We also created a sheet for each question that obtains its data from the master sheet and allows us to plot charts and tables that further deepen our analysis reported in this study. Complete results from the survey are presented in Chapter 6 and an anonymized version on the spreadsheet is accessible through https://www.site.uottawa.ca/~tcl/gradtheses/eayeleso/.

3.3 Our Design Science Research to Create Artifacts

Design Science (DS) is "fundamentally a problem-solving paradigm. It seeks to create innovations that define the ideas, practices, technical capabilities, and products through which the analysis, design, implementation, and use of information systems can be effectively and efficiently accomplished" (Hevner & Chatterjee, 2010). Design science addresses unsolved problems in evolving domains by developing the background knowledge in parallel with creating artifacts. This is quite different from ordinary engineering design that makes used of well-established methods, models, and principles to solve problems. Key differentiators between standard design and design science are firstly that an outcome of design science is to contribute to the knowledge base of theory, technology or methods, and secondly to communicate those contributions to the stakeholders of the domain (Hevner & Chatterjee, 2010).

"Design science (DS) research is motivated by the desire to improve the environment through the introduction of new and innovative artifacts and the process for building the artifacts" (Hevner & Chatterjee, 2010). Hevner and Chatterjee's describe the goal of creating solutions to

the limitations of the tools that are available to negotiators while undertaking qualitative negotiations as purposeful thinking. DS is a pragmatic research paradigm that posits creation of innovative artifacts to solve real-world domain-specifics problems.

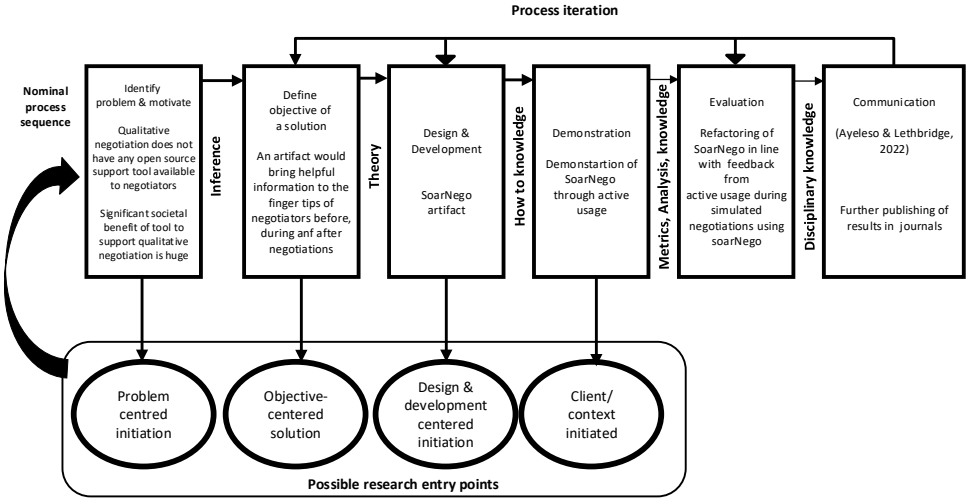

Figure 12: DS process model adapted from (Peffers et al., 2007)

Figure 12 presents the consensus-building DS approach that we have adapted as our DS guidelines (Peffers et al., 2007). The approach synthesizes DS process models from seven representative research activities, as reported by (Peffers et al., 2007). We describe part of these guidelines next.

3.3.1 Identification of Qualitative Negotiation Process Challenges

We triangulate set of requirements collected from multiple data sources. As discussed in Section 2.2.3, we studied the existing negotiation model designs in the literature. We studied the aspects that work and those that will not work in the qualitative negotiation context that is the focus of our study, as opposed to the quantitative negotiations that many of the existing negotiation models focus on. We adopted or adapted designs that are applicable to the context of our research focus. We also noted the grey areas in that the existing models have not considered.

We identified that many qualitative negotiation-specific operations are not fully considered by most of the existing models designs. Similarly, the existing tools that we have reviewed, and some that we have tried our hands on cannot support qualitative negotiation effectively. These tools were never designed for qualitative negotiation in the first place.

3.3.2 Motivation for Qualitative Negotiation Artifacts

Our identification of the problem mentioned in Section 3.3.1 requires an artifact that would effectively support qualitative negotiation. The artifacts, which we will develop in this research, target implementation of requirements that a tool that will support negotiation effectively should possess.

3.3.3 Description of how new artifacts would support solutions to the problems.

The artifacts will implement use cases and requirements for features (Chapter 5), a metamodel (Chapter 7), as well as a prototype (Chapter 8) of a tool to effectively support negotiation as defined by the results of our studies. The final artifacts are results of triangulation from multiple data sources: AR, SLR, ALR, GT and survey. The artifacts are open-source and available for use by the stakeholders.

Chapter 4 Results of the Grounded Theory Research

In this chapter, we present the results of the data collection and analysis done during the GT exercise discussed in Section 3.1.

We have divided the themes (i.e. GT code categories) into two groups. The first group is central to the remainder of the thesis as it discusses issues related to software tool support for qualitative negotiation. This group is the subject of Section 4.1. Later in the chapter (Section 4.3) we cover secondary themes, such as what negotiators experienced while negotiating virtually and in the pandemic. These results are interesting, but we will not be making use of this second group in later chapters.

For both groups, we present quotations from the interview sessions that led to the emergence of the various concepts and themes.

We consolidated the themes related to tool support into a negotiation model, that we present in Figure 13, and discuss in Section 4.2.

The results of the GT analysis reveal that the existing support tools used by the negotiators do not have sufficient capabilities, particularly the capability to manage, view, search and compare history, notes, and background information regarding agreements. Our GT results also define baseline requirements for a negotiation tool that would support qualitative negotiation successfully; this is presented in Section 5.4. The results also contribute to our metamodel, presented in Chapter 7.

4.1 A Detailed Look at Some Tool-Related Code Categories

In this section, we provide examples of themes that arose from the analysis related to the tools that qualitative negotiators use. We corroborate them by giving quotations from the interviews.

4.1.1 Negotiators Know They Use Unsophisticated Software Needing Improvement

Negotiators' activities are supported by computer technology across every stage of the negotiation process. However, negotiators feel a lack of satisfaction regarding the tools accessible to them.

A key result from our research is that without exception, they used Microsoft Word to edit their agreements. The author's supervisor had observed this in almost all negotiations in which he had participated, but we were actually surprised that *none* of the interviewees had used more sophisticated technology.

But our result is even stronger than that: The interviewees meet and interact with large networks of colleagues. The union leaders exchange knowledge and practices with large networks of unions in their areas, and the labour-management teams do the same. The international treaty negotiator similarly work on numerous treaties and are thoroughly familiar with large networks of bi-lateral and multi-lateral international negotiators. So it was of great interest that *all* our interviewees confirmed that, without exception, their colleagues all also use Word for editing their documents during negotiation. In other words, we can state with confidence that *all other international negotiators* known to the trade negotiator we interviewed, and *all other unions and management teams* known to the labour negotiators we interviewed exclusively use Word. Taken together these colleague networks represent many thousands of negotiators, making this a very strong result.

In addition to Word, the negotiators all used some kind of technology to manage and share multiple Word files within their teams. This might be Microsoft OneDrive or SharePoint, or Xerox Docushare, or Google OneDrive. They would normally also print the files and keep them in a binder, with tags pointing to important information; they had no simple way to search or cross-reference the files stored electronically, although they relied on naming conventions and directory hierarchies to organize the files. Sharing of files with the other parties would be done on paper or by email. Sharing within the members of a negotiating team might also be done this way.

Example quotes from the interviews:

> "We are pretty archaic at [organization], we don't have any sophisticated technology …" – P#8
>
> "But I will say, it is not very sophisticated. It could be a bit time consuming; you will be working from multiple versions. But we will always take the latest version." -P#6

The phrase, "we don't have any sophisticated technology" or, "not very sophisticated" and similar narrative in the codes easily formed the *ineffective technology capabilities* category for us.

Since early interviews quickly reaffirmed our prior observation that negotiators use Microsoft Word as a key tool, we restructured some of our questions for subsequent interviews to learn *why* negotiators generally use Word, *how* they use it, and what operations do negotiators feel Word does *not handle* effectively. This led us to define requirements (presented later in Section 5.4) that could be a baseline for effective tool design.

4.1.2 Capability Needed: Better Information and Change Management

This code builds on the previous one. We observed general difficulties in the information management capabilities of the software technology used by the negotiators. As mentioned, the main document format is Word files, often printed and stored in binders, and/or converted to pdfs and stored in some form of document storage system (OneDrive, Google Drive, etc.).

New versions are stored, with markup of changes as negotiations proceed. Updated versions are made available to some or all team members, or simply emailed to team members. Exchange with the other party is most commonly on paper or by email. Some quotes:

> "...It is hard to keep track: I make this change, then they make this change, then we make this change on the change. … we could have track changes and things that are highlighted in grey … You are working on parts that were accepted and others that you are still working on. …"- P#3

> "… can be difficult to locate sometimes the exact thing you are looking for because of the volume of information you have in preparation for the process and as the process unfolds, you gather more … I will say information management is one of the biggest challenges."- P#8

The key trouble is that there are very large numbers of versions of each document that accumulate over time. Each version may have a purpose such as a version being prepared to send to the other party, or the response of the other party. A version may have just a few key changes,

or maybe a lot of relatively unimportant 'housekeeping' changes, such as updating dates or references to other documents that have changed. Versions may have notes associated with them, such as the rationale, or the comments received from the other party. Negotiators forget changes nor notes recorded in each version and have to spend a lot of time searching through electronic or paper records, and comparing clauses line by line. Or they may just not bother to do these searches because it is so hard, or due to lack of time: They may hence just rely on memory, resulting in inevitable errors.

So, it is clear that difficulty in managing information is a core concern; the following subsections describe more specific concerns related to information management.

4.1.3 Capability Needed: Quick Access to Information While Negotiating

Negotiators desire to have all pieces of relevant information needed for negotiation quickly accessible to them at the negotiation table. This is in contrast to their present typical, which is cumbersome, with the use of paper binders, often with colored tags to help them find important information.

Some quotes:

> "Typically for me, for each negotiating round, I have a bunch of these [participant raised up binders] …" -P#2

> "So, I will not have to carry around a binder of 500 pages and all my versions that grow. …"- P#7

> "And then we have binders, like all the versions, …. It can get pretty cumbersome…"- P#3

> "I keep copy of everything from day one …. I have the first response they sent. … We go back forth and back..."- P#4

> "There are … junior officers in the room whose job is to be passing notes to the negotiators, helping to flip through pages." -P#2

Several negotiators assign team member(s) who flip through binders to access information, or search electronically. Where successful, they pass retrieved information usually to the lead negotiator presenting at the negotiation table. It would save time and resources if the lead negotiator had the information available at her or his fingertips.

4.1.4 Capability Needed: Access History of Clauses in Agreements

History is important to negotiators. They want to know how things have evolved. Some quotes:

> "There have been instances in my past where we had to actually dig up the notes and history around the different iterations of clauses. ."- P#6

> "For the changes, we just keep a previous iteration of language to really document the history of how you arrive to a certain point. .."- P#6

> "… we will go back to that first version, … to know when a particular change was made. We will have to open maybe 20 documents to be able to find that particular modification. It is very manual, depending on what you are looking for, if you don't recall the date that exact changes were made..."- P#6

Information gathered from history guides understanding: the intent and evolution of clauses; supports argumentation (rationale), and wording of contractual text. Negotiators forget why certain clauses are present, or what was the rationale for changes. Without access to history, either side may be at a disadvantage due to forgetting or overlooking important information.

4.1.5 Capability Needed: To see Evolution of Each Clause

Beyond just seeing previous versions, negotiators want to know who edited what and when in the contractual text. Some quotes:

> "… proposals … can be presented multiple times ,.. So, there is a lot of back and forth. You could have the same- proposal back and forth maybe 12 times during the session. So, what we do is to record at the top of each document each time a proposal is presented and by what party–proposed on such a such a date by which party, counter proposed on such-a-such a date by the other party etc. The heading may be very long. … a half of a page sometime, which are history of the exchanges of this proposal."- P#4

> "…you don't have to go through the computer system or filling boxes by pulling these versions of languages to see what the evolution is. ...Then you could click on and read them quickly and see what happened. That will be invaluable. It will probably add 10 years to my life."- P#7

Seeing the evolution of a clause is a step beyond just seeing history at a point in time. Such detailed information can be needed to help verify that the decision finally reached reflects the give and take of the discussion.

4.1.6 Capability Needed: To be Able to Record and Review Notes

Notes are normally taken by multiple members of each negotiation team about what was said at the negotiation table, including what either is proposing and why. Notes are independent of the contractual text and are used to elucidate agreements or disagreements and the intent of clauses and proposals. They are normally kept confidential from the other party, but sometimes might be revealed if there are disputes, or may be made accessible to an arbitrator. Rationale is normally recorded in private notes of members of one party and only communicated orally to the other party, who in turn take their own private notes of what they have heard expressed orally. The exchange of rationale electronically tends to be limited so as to avoid the other party obtaining wording that might have greater legal weight if it was in writing.

The notes tend to be very informal, sometimes merely on paper, and often in separate electronic documents from the documents they refer to. They may perhaps be Word or pdf

comments embedded next to text, but due to the need to avoid having these notes become part of the final agreement, the annotated files are *additional duplicative files* that are distinct from and have to be tracked separately from the ones that represent the formal wording.

Some quotes:

> "I think if there were just a way to quickly bring all these [notes] and cross reference all of the various documents, I think that will be helpful..."-P#8

> "Notes are critical because they also give you the bargaining history and the intent of what was discussed." – P#7

> "…but the notes are not easily linkable in terms of what you are looking for and the date of what you are looking for…"- P#4

> "I think obviously building a database of notes going forward will be helpful." -P#6

The independence of notes and the contractual text requires an effective cross-referencing tool, and care to avoid revealing notes to the other party.

4.1.7 Capability Needed: To be Able to Comprehensively Search and Compare

Negotiators want to search for pieces of information such as clauses and key words in the contractual text, in the notes and in other documents. Negotiators might also need to compare retrieved results across these sources (highlighting differences). Some quotes:

> "…can be difficult to locate sometimes the exact thing you are looking for because of the volume of information you have in preparation for the process and as the process unfolds, you gather more …" -P#8

> "we don't really have a great system for [searching]. It is not tied to any specific article and sometimes we have to go back and refer to the note. We know we talk about certain heading, topic certain date, so we have to go back and search the notes. … you can work right on the Word document by search if you are looking for a specific article or a specific heading or subject." -P#8

Searching for a piece of a clause and comparing with similar ones, informs negotiators' understanding and may guide the wordings of contractual clauses, proposals of counterproposals.

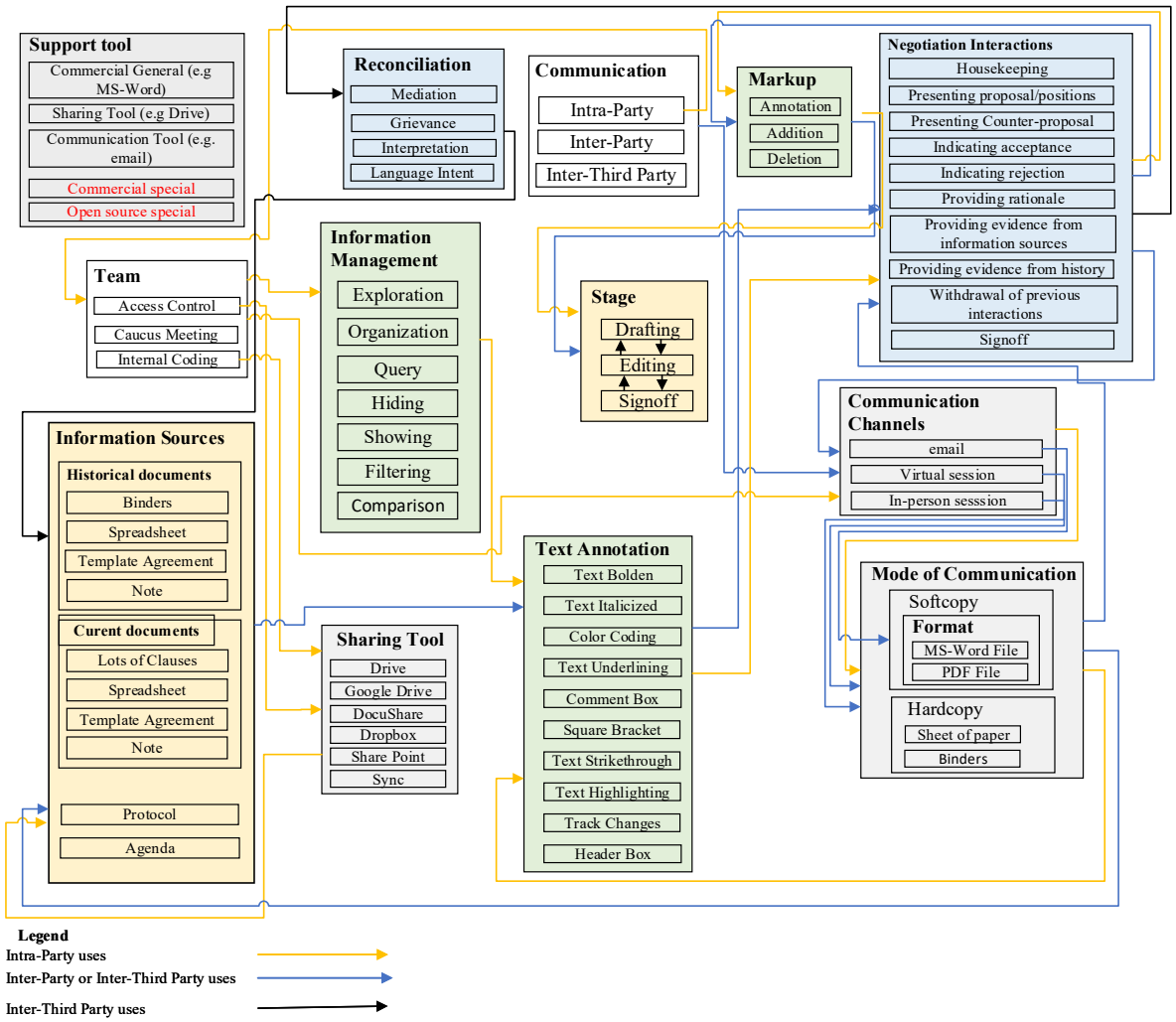

Figure 13: Negotiation model derived from grounded theory data during the selective coding step

4.2 GT: Negotiation Model

Figure 13 is the resulting model from the grounded theory conducted in this research. A model from grounded theory is by nature informal since there are no standard notations or rules for drawing the model unlike the case of class diagrams, entity-relationship diagrams (ERD) and data flow diagrams (DFD) that can be used to represent certain other information in the domain (Halaweh, 2012).

Typically, informal models use simple language and representation that are understandable to end users and other stakeholders. Similar informal model emerged in grounded theory research

of a variety of other authors (Carvalho et al., 2005; Chakraborty & Dehlinger, 2009; Coleman & O'Connor, 2007b; Georgieva & Allan, 2008). "On the other hand, the equivalent models in RA [requirement analysis] such as UML [unified modeling language] models (e.g., class diagram) is easily created from the informal models and used in communication between the analyst and developers"(Halaweh, 2012).

Figure 13 presents background color coding of related concepts. While colored arrows (see the legend) show whether the concepts relate to processes within a party, or between parties. For instance:

- Information Sources: These include historical or current documents that serve as factual sources explored by negotiators when retrieving information needed for document-driven preparation and document-driven presentation at the negotiation table. A key information source is also the current contractual text. Information source themes are colored yellow and are at the bottom left.
- Party and Modes of Communication: These are groups of concepts that are used in isolation or in conjunction to communicate. The categories are colored light gray, and are in several separate boxes in Figure 13.
- Interactions: These are the types of interactions that occur during the negotiation. These are colored blue.
- Information Management: These are the operations and contextual features that negotiators expect in a tool that would support negotiation effectively. They are colored green.
- Support Tools: These are the observed classes of tools that can be used to support negotiation; they are a special type of communication mode colored darkest gray at the top right. Commercial special tools are not easily accessible to or adopted by negotiators, and open-source special tools do not yet exist. These two concepts are highlighted using red text in the support tools box.

4.3 Detailed Look at Other Code Categories

There are some other factors that contribute to the success of negotiation beyond the support tools. Such important factors' codes also emerged from our GT. The codes in this section

were not used in later chapters for development of artifacts such as our metamodel and prototype, but are nevertheless useful research results.

4.3.1 Many Negotiators Prefer In-Person Negotiation

Several of codes buttress a conclusion that negotiators tend to prefer in-person negotiation over virtual negotiation. Some general quotations confirming this preference:

> "I think it's easier to negotiate in person." -P#1

> "[The] reality is trade negotiations are best done in person..." -P#2

> "Much easier to negotiate in person, but, unfortunately, we have almost [negotiated virtually for] 2 years now". -P#8

Next, we discuss some categories of codes supporting the reasons why negotiators prefer in-person negotiation over virtual.

4.3.2 Negotiating Takes so Much Time that Virtual Negotiation Becomes Impractical

Negotiation takes a lot of time that may run into months or years. Typically, the negotiating parties meet in person at the designated centre for the negotiation. Negotiators believe doing such marathon sessions virtually will not be feasible, due to challenges arising from the lack of human interaction. Some relevant quotes from the interviewees:

> "When you do in-person bargaining, you can sometime plan a whole day or weekend and that's something that I will not recommend virtually." -P#1

> "They basically take place over an entire day and usually an entire week. Like a 40 hours week. Let us say 50 hours over the course of a week, negotiating with a partner." -P#2

4.3.3 Monitoring the Other Party's Concerns and Consent is more Difficult in Virtual Negotiation

Reading the reactions and the body language of the other party during negotiation is very important to negotiators. Such reading is important to gain confidence in the actual positions or concerns of the other party as regards the decisions reached during the session. Non-verbal gestures of the negotiating parties are a key and quick means to ascertain the extent to which the other party has hidden concerns, is in agreement, is willing to make a compromise, or is simply willing to generally cooperate or not.

Virtual meetings make it difficult to read non-verbal norms and gestures. Unfortunately, negotiators may not comply with the protocol to keep their camera open during virtual negotiation, making detection of non-verbal information even harder.

> "There may be a team member that [does not] put their camera on and you might miss the opportunity to view that non-verbal [communication]. Without meeting in person, it may be little challenging to know the body language of someone in this virtual context." -P#6

> "It would be very difficult … I know you could see the faces. We had a bargaining protocol … asking people to keep their camara on always. But it's not always [on]. … it's not always the same as being in person. That would be some of the [biggest] challenges encountered." -P#1

4.3.4 Differing Time Zones Hinder Virtual Negotiation

Different time zones across some countries and continents make it difficult to fix a convenient negotiation time for expected participants across continents and countries. For instance, It will be odd for two negotiating teams, one during the day, and the other team late in the night to be engaging in virtual negotiation. Some quotations confirming this difficulty are:

"Because you really need face-to-face to do that [to conduct long session negotiations] and [the] time differences is a challenge" -P#2

"Time differences make it really difficult to do that [to conduct virtual negotiation]. Let say you are negotiating with someone in Asia for example; the time window [IS] 3 hours, maybe they are just waking up [from] sleeping or you are. So, you can do some work using … video conference platform... But there is only so much you can get done, maybe 3 hours a day". -P#2

4.3.5 Socializing is Difficult and May Not be Feasible in Virtual Negotiation

Creating a good relationship with the negotiating parties is a strategy employed by many negotiators to improve the chances that the other party will understand their reasoning or be willing to compromise in order reach agreement on critical subjects, and to avoid risk of deadlock. Informal contact, via in-person socialising is very helpful to successful negotiation. Some quotations confirming this are:

"a lot of the real heavy lifting in negotiations does not get done in the negotiating room, it gets done in the hallway. Where you put your counterpart aside and say, 'I say listen, I can't say this in front of my negotiating team, But I really feel we can look for a common area.' So, it's that kind of conversation is where the actual agreement takes place. Sometimes, what is done in the negotiating room is a lot of theatre." -P#2

"I mean we experience that [human-to-human interactions] day-to-day, that human connection may [be] missing in this virtual context" -P#6

"Something you can definitely say, is that successful negotiation depends on kind of building face-to-face relationships. And that

involves not just seating across a table from someone, but you know having dinner, like delegation dinner, building trust. That is a big part of it." -P#2

Some other quotations confirm that Covid-19 forced many negotiations to switch from in-person to virtual. Such transition slowed down many negotiations. Some negotiators decided to discuss only trivial issues during this period, while the critical issues were placed on hold till after the pandemic. Table 5 summarises the difference between pre-pandemic versus pandemic negotiations according to the interviewees, and Table 6 summarises in-person negotiation versus virtual negotiation. These comparisons were deduced from the interviews' data.

Table 5: Emerged Differences Between Pre-pandemic and Pandemic Negotiation

Pre-Pandemic	Pandemic
Protocol and contractual text would be distributed at the negotiation table	Protocol and contractual text are sent ahead of the negotiation meeting
Used to negotiate both high and low sensitive issues	Data gathered showed a general slowdown in the negotiation of highly sensitive issues
Dominantly featured in-person negotiation	Dominantly featured virtual negotiation

Table 6: Emerged Difference Between In-Person and Virtual Negotiation

In-Person Negotiation	Virtual Negotiation
No time zone hinderance	Time zone difference does hinder long-term and long-time negotiation session
Comparatively may cost more	Saves some logistic cost, such as: Transportation cost of the negotiators
More preferred by the negotiators as its easier to monitor non-verbal gestures and compliance to protocol	Difficult to monitor consent as regards protocol can be easily breached. For instance, a negotiator may not turn on their camera, making it impossible to monitor non-verbal gesture
Binders are carried around by negotiators and hard copy is shared at the negotiation table	Documents are sent electronically, and accessed electronically
Easier to reach agreement as a result of human face-to-face interaction	More difficult to reach agreement
Whispering, tapping, raising of hand is used to call for caucus meetings.	Slack and other apps are used to call for caucus meetings

4.4 GT: Results Discussion

We now summarize the grounded theory part of this research.

Contractual text is the main instrument of negotiation on which we are focusing. It evolves as either party proposes changes. Clauses are discussed by negotiators, with argumentation from notes or other documents being used to explain their rationale. Clauses are subject to multiple rounds of counterproposals, some of which are eventually accepted.

A typical agreement is managed in Microsoft Word files, with large agreements being typically split into separate Word documents, such as one per article. New versions of each of the documents are created by people proposing changes internally in a team, when preparing a version to present to the other team, when proposals are received from the other team, and when annotating all of the above with notes. There is thus a vast and complex array of Word documents (sometimes printed or converted to pdf) to manage.

The multiple documents are often managed using a file storage system such that all the members of a negotiating team can access the same files. Multiple versions of each Word file are normally maintained in the storage system, to allow negotiators to look back into the history of the changes. Other files, such as files with notes about rationale, may be maintained in the same storage system.

As negotiations proceed, Word documents are marked up in some way to indicate the changes that are being proposed or have been accepted. Some negotiators use the 'track changes' feature of Word, while others use custom markup schemes such as using colours or bold or italics. Tracking which party has bargained for which wording change and the relevant rationale, i.e. knowing the history, allows negotiators, as well as the boards or members they report to, to ascertain whether a fair compromise has been reached. Access to the history of changes also allows for negotiators to check that no mistakes have been made.

Proposals for changes may be linked together, such that changes in several articles must be accepted together, or not at all.

This manner editing of the contractual text and the need to maintain editing history, as well as notes and background information, makes managing the evolution of agreements challenging. The existing generalized support tools such as Word, cloud-based storage tools, and email used by the negotiators handle modern negotiations in a very sub-optimal manner.

Negotiators desire a more sophisticated tool that could quickly give them easier access to information, more easily change the information and better understand the evolution of information. In particular, they desire that such a tool could perform wording comparison of the clauses across version of the contractual text; filtering and display of only desired results on their screen. They also want to ensure authorised access only to the notes taken during the negotiation sessions, and possibility to cross referencing the notes to the relevant clauses, or to other types of relevant notes.

Beyond what happens in the negotiation room, negotiators want to build a face-to-face relationship and trust with the other team. The protocols and formalities in the negotiation process may set the tone for negotiation. Some processes may lead to a relaxed and comfortable mood, whereas others may be counterproductive towards reaching satisfactory compromises. Strategies such as having a cup of coffee or dinner together, or chatting in a relaxing spot, are explored by negotiators to set a productive mood that positively affects reaching compromises during negotiation sessions. Like one of our respondents said, "it's that kind of conversation [referring to informal chats] is where the actual agreement takes place. Sometimes, what is done in the negotiating room is a lot of theatre."

However, frustration with the tools in use during negotiation can also impact the tone. If tools cause frustration, errors, delays or misunderstandings, a bad relationship can develop.

Despite the transitioning of many negotiations to virtual mode during the pandemic, negotiators feel virtual negotiation slowed down negotiation as many critical issues were not negotiated during this time. They also find it more difficult to assess the cooperation of the other party to make compromises during negotiation sessions. The difficulty is primarily caused by the inability to read non-verbal communication such as body language of the other party. This is made worse when negotiators refuse to turn on their camara during virtual negotiation.

4.5 Conclusion

The GT approach helped us form the groundwork for socio-technical system design. It helped us to fill gaps in, and strike a balance among, social and technical issues in traditional requirements analysis. The traditional requirements analysis approach mainly focuses on technical functions and constraints of the system.

The GT results described in this chapter will help us in later chapters to define negotiation support tool requirements and a data model that can be a baseline to future researchers and developers that may want to work on textual negotiations. The work in this chapter sets the stage for a more detailed survey to deepen confidence in our findings, as well as our prototype that will implement negotiation support tool requirements that emerged from this study.

Chapter 5 Suggested Use Cases and Features for a Qualitative Negotiation Support System

Informal data gathered from the researcher's supervisor during his experience at the negotiation table, literature reviews (SLR and ALR) and the early phases of our grounded theory work uncovered a set of capabilities that must be supported in an effective qualitative NSS. In this chapter we initially introduce set of general use cases that our research uncovered (Section 5.1). Table 7 compares similarities and differences of these use cases in the context of what is obtainable in MS-Word (the word processor used for most qualitative negotiation currently, as discussed in Section 4.1.1), Github and our proposed technology. We also indicate what we have prototyped in SoarNego, the qualitative negotiation tool developed in this research, that we will present later in Chapter 8.

As our grounded theory work progressed, we validated and tuned the initial set of use cases, but also uncovered an expanded set. These are discussed in Section 5.2. presents the relationship between the two sets of use cases, the research process that produced each use case and GT theme(s) that validated or expanded each use case. In Section 5.4 we describe the most important set of qualitative negotiation requirements defined by the two sets of use cased introduced earlier.

5.1 Initial Set of Use Cases for Qualitative Negotiation Tool

We will present use cases in this chapter in a relatively informal manner.

We uncovered the first set of textual agreement use cases for a tool that will support qualitative negotiation from the supervisor's experience at the negotiation table (discussed in Chapter 3) and the literature reviews (SLR discussed in Section 2.2 and ALR discussed in Section 2.3). We next present this first set of the use cases:

5.1.1 Creating a New Document or a Proposal for a Revised Version

A proposal may be an initial document, or proposed changes to an existing document. By its nature, a proposed change to an existing document means that there will then be multiple versions in the system if the proposal is accepted. There needs to be a way to create these.

UC-Cr-1-Init: Inputting an *initial* document in the system by *typing* it in, or copying and pasting from other sources such as a template, or a previous agreement that will now be managed by this system. Other sources may be in in word processing or or pdf form.

UC-Cr-2-Parse: Inputting an initial document by *parsing* from another source. Previous versions or templates may only be available in pdf, so a tool may be needed to parse the pdf. No matter the source, *dividing the agreement into clauses* automatically will help negotiators later track changes, since clauses are so elemental to agreements.

UC-Cr-3-Prop: Inputting a *proposal for changes* to an agreement that is already in the system. This would consist of a set of textual changes to an agreement: additions, clause edits, deletions, rearrangements, and renumberings. There will likely be many such proposals during a negotiation.

UC-Cr-4-Condit: Making a proposal that has conditions (e.g. either text A and B or text A and C)

Note that creating a proposal for a document or changes is not the same as sending it to the other party or merely repeatedly editing text. A proposal involves initiating an entirely new potential version.

5.1.2 Editing Text

UC-Edit-1-Simple: Simple editing of existing wording that has already been entered in the system, *without establishing a new version*. Being able to edit is elementary, and many edits may be done before the resulting text is proposed as a change to be considered by others (which would be UC-Cr-3). All word processors allow editing, and any new tool would allow it too. Tracking of changes made to pre-existing text by the edits would be an essential part of the process.

5.1.3 Searching for Text

UC-Srch-1-Within: Searching *in a version of the agreement* for wording. It is not normally infeasible to expect negotiators to have to visually scan when they are looking for information. All word processors allow in-document searching, so this is an elemental feature. The search may need to cross the boundaries of what would have been several documents prior to the use of the new tool, such as in the case of where an agreement had separate documents for each article. The result of this operation is that the found text is highlighted; multiple locations of the found text would all be highlighted.

UC-Srch-2-Across: Searching across *multiple documents* when an agreement is large and is split, perhaps into one document per article.

UC-Srch-2-Versns: *Searching the change history of a given document* for wording. This is a more sophisticated operation that enables negotiators to search for text that may have been deleted, but existed in a previous version. Negotiators may want also to know when the wording was introduced or deleted, and by who. The result of this would initially be a list of versions, with a little context about the found words. The user can click on an item in the list to see the full historic version with the wording found highlighted.

5.1.4 Sending a Proposal to Others

UC-Snd-1-Propintra: Sending a proposal to *team members*. A team member may privately prepare some changes that she or he suggests might be suitable to propose to the other party. There needs to be some way for the team member to make a private version appear as a candidate to be considered by the team.

UC-Snd-2-Propinter: Sending a proposal to *other parties*. Once a team agrees that a proposal will be sent to the other party, there needs to be a way to do this.

Uc-Snd-3-Propreply: Replying to a proposal, i.e. providing a *counterproposal* consisting of edited changes or new changes, or some changes being rejected. This combines capabilities of the previous use cases.

5.1.5 Entering Rationale for or Notes About a Proposal

All of the following might be made privately, or shared within the team of a party, or shared with other parties.

UC-Note-1-Rwhole: Providing rationale for a proposal as a whole.

UC-Note-2-Rchange: Providing rationale for each change that is part of a proposal.

UC-Note-3-PrCmt: Recording comments or feedback about a proposal that someone else has created (or parts of it). This might be arguments against the proposal, or questions about it.

UC-Note-1-Revl: Progressively revealing certain rationale or notes to others.

5.1.6 Indicating Agreement or Disagreement With a Proposal

UC-Indic-1-AgIntra: Indicating, within a team, that you agree with a proposal or disagree with it. This is a normal process prior to transmission of the proposal to other parties. Normally members of a negotiating team work towards a consensus, where all team members are happy with a proposal before it is sent to the other parties.

UC-Indic-2-Disag: Indicating to other parties that one party disagrees with (rejects) a proposal without providing a counterproposal. They are proposing the 'status quo' in a simple way, rather than counterproposing with the original words.

UC-Indic-3-Sign: Indicating to the other parties that you agree with certain wording, generally specific clauses in a proposal, or an entire proposal (which may be only part of the entire agreement). This is often formalized as signing off among all parties on a proposal that all parties accept, that will become part of the final agreement, or at least a baseline for further negotiation. Once a party signs in this way, it would be considered bargaining in bad faith to back out of agreement on this particular wording.

5.1.7 Withdrawing a Proposal

UC-Wd-1-Temp: Withdrawing a proposal (taking it off the table, but without deleting). It could be introduced again later in the same or modified form. This use case may be invoked when negative comments or questions are received by the other party, but not a counterproposal.

UC-Wd-2-Del: Deleting a proposal entirely. This might be particularly relevant for an individual or within a team, but it is also relevant after another party responds negatively to a proposal, and the proposing party does not want to 'push' the issue further so as to more easily reach a final overall agreement.

We observe that some of these use cases represent operations (not unique to negotiation) that MS-Word can handle. These include creating a new document, editing it and searching within it.

Many of the operations are also undertaken as software development teams negotiate among themselves about code, i.e. deciding on which changes will be made to implement a feature or fix a bug. Therefore, we realized that comparing the use cases to those available using Git and GitHub would be a useful exercise. The comparison to code negotiation is not an exact one, because with code there is generally only a single team (a single party) with members that work towards a consensus. Also, while Git and GitHub may have some features we are advancing in this research, and as discussed later, use of either technology for qualitative negotiation is not feasible for several reasons, including:

- Git and GitHub can be used to propose changes to plain text (e.g. through 'diffs' or 'pull requests'), but this typically happens in a fully collaborative context, where all team members can see and approve, or further edit, each others changes. The tools do not make provision for the potentially confrontational process, where changes have to be first discussed within a team, and then presented to one or more other parties.
- These tools do not provide a 'what you see is what you get' (WYSIWYG) interface by default, unlike Word and other word processors.
- The tools require a higher degree of technical expertise than would be possessed by typical negotiator.

Nonetheless, the features of Git and GitHub can help inform the design of the technology we are proposing in this thesis.

Table 7 compares our negotiation use cases with what is obtainable in MS-Word and Github. The ones not obtainable in these technologies imply negotiators will have to improvise how to cover such operations. In the table we have provided our assessment of the extent to which the use cases are doable in Word and Github using the following notation: ✓ means the use case

is intrinsically doable. ~ means the use case is doable with extra complexity or mostly doable. ⊂ means that the use case is not doable or requires very awkward workarounds. The second-to-right column indicates our design aspirations for the technology we are proposing in this thesis: We intend the eventual tool to be able to do all these use cases.

In Chapter 8, we will discuss SoarNego, that targets some of the use cases as a prototype. The rightmost column in Table 7 indicates the extent to which SoarNego has so far enabled the use case. We have used the symbol ☞ to indicate that the use case is future work.

Table 7: Comparison of how Word, GitHub and the proposed technology support the initial set of use cases of a qualitative negotiation support system. A checkmark indicates doable; an X indicates not doable. ~ means partially. A forward pointing hand means future work.

Use Case Code	Summary	MS-Word	Github (+Git)	Proposed technology	In Proto-type?
Category: Creating new documents or proposals					
UC-Cr-1-Init	*Typing* text or copying and pasting text (i.e. using ctrl or command C or V)	✓: Basic feature of all text editors	✓: Basic feature of all text editors	✓ : Basic feature of all text editors	✓
UC-Cr-2-Parse	Capturing structured text from another source by *parsing* (e.g. from a pdf file)	✗: Cannot parse any document into required internal structure.	➲: Cannot parse any document. External tools must be used.	➲ : Parse documents into internal structures matching the metamodel as discussed in Chapter 7	☞
UC-Cr-3-Prop	Making a *proposal* or a *new version* of one that can later be referred to	✓: Create a *new version of the file* with change tracking or informal markup of proposed changes	✓: A proposal can be made with a formal *pull request* on a new branch.	➲: Track *proposals as formal data*, so they can be compared, annotated, and searched	☞
UC-Cr-4-Condit	Making a proposal with conditional logic	~: Can be expressed using comments or markup but not captured formally	~: Can be partly expressed using branches or alternative pull requests, with informal commenting	✓: Allow *formal* representation of the logic so it can be analyzed and visualized	☞
Category: Editing					
UC-Edit-1-Simple	Ongoing editing of existing text of clauses or proposed changes	✓: Simple editing of text. *Track changes* can be used to mark the edited portion of the text, or else *markup* such as bold or colours can be used	✓: Github allows editing of plain text. No need to worry about tracking changes as these are calculated when needed using *git diff*	✓: Simple editing of text. Changed text is *instantly highlighted* with colored background	✓
Category: Searching					
UC-Srch-1-Within	Searching for specific text *within* a document	✓: Word allows user to search within a single document; highlights all changes matching the key	✓: Searching within a file using browser search	✓: Allows searching within document, highlighting all changes matching the key	✓
UC-Srch-2-Across	Searching for specific text *across* documents, such as multiple articles of an agreement	✗: Word does not support this operation. Negotiators have to manually search multiple files.	➲: Search operation creates a *list of locations* where the key is found in multiple documents across directories, repos, organization or all of Github. All found words are bold	✓: Search operation creates a *list of locations* where the key is found in multiple documents	✓

Use Case Code	Summary	MS-Word	Github (+Git)	Proposed technology	In Proto-type?
UC-Srch-3-Versns	Searching for specific text in *previous versions* of the same file (or the same clause, etc.)	✖: Word does not support this operation. Negotiators have to manually search.	~: '*Git log*' can be used to identify commit hashes to search through versions, and 'git show' plus a standard text search tool like 'grep' to search for desired text	✓: Has modules that handle this operation as discussed in Section 5.4	✓
Category: Sending					
UC-Snd-1-Propintra	Making a *proposal* only visible to team members	~: An online Word document can be shared just with specific people, but it is the *whole document* shared, not just the proposal.	~: *Private Github repos* can be used, but it is not then possible to selectively share just proposed changes with others. A *private non-Github repo* could be shared with team members prior to publishing on Github, but this requires technical expertise.	➲: Visibility control where a negotiator can set visibility of a proposal to team members only.	☞
UC-Snd-2-Propinter	Sharing a proposal with the other party	~: A Word document can be shared or can be sent through email, dropbox, etc, but it is the whole document shared, not just the proposal, and tracking of such proposal files is challenging.	~: Repository owners (a party) can invite other negotiating party members, but this is only for the repo as a whole, not just a single pull request.	➲: Visibility of a proposal can be extended to other parties	☞
Uc-Snd-3-Propreply	Replying a proposal by accepting, rejecting, or editing it.	~: Same as for UC-Snd-2-Propinter with new markup. But markup on markup and file naming becomes increasingly challenging the more iterative counter-proposing occurs	✓: Acceptance or rejection can be done through the Github pull request review process. A counterproposal would be done using additional commits to the pull request, or a branch based on the original branch.	➲: A new proposal is based on the previous one, with differences visible.	☞
Category: Notes					
UC-Note-1-Rwhole	Making notes to justify a whole proposal (i.e. rationale)	✓: Comment features in MS-Word can achieve this operation informally. It may not always be clear what the comment is referring to though.	➲: Comments can be made on a pull request as a whole.	➲: A note can be assigned to a whole proposal	☞

Use Case Code	Summary	MS-Word	Github (+Git)	Proposed technology	In Prototype?
UC-Note-2-Rchange	Making notes to justify a *part* of proposal (e.g. why a particular clause is being changed as part of the full proposal)	✓: The comment features in MS-Word can achieve this operation. Annotations in the text (e.g. in square brackets) are used by some negotiators.	➲: Comments can be made on a commit. They can also be embedded in code, referring to the pull request number.	➲: A note can be assigned to a specific part of a proposal	☞
UC-Note-3-PrCmt	Providing feedback for or against a proposal	✓: Comment replies are possible in Word.	➲: Comments can be added in the code review process, or as an additional comment on the pull request as a whole.	➲: Notes can be made on other notes.	☞
UC-Note-1-Revl	Controlled viewing of notes/comments	✗:MS-word does not have features that support restricting viewing of metadata such as comments of a document to only some people out of all that have access to the file	✗: Github does not support this UC.	➲: Designed to allow such kind of restricted viewing of a document. Comments restricted to party members only will not be visible to the other parties' members unless changed.	☞
Category: Indicating Agreement or Disagreement					
UC-Indic-1-AgIntra	Indicating opinion of a proposal within a team.	~: Can be done through comments, but awkward.	➲: Could be done with approval process on a PR in a private repo. But see UC-Snd-1-Propintra for issues.	➲: Designed to support this UC	☞
UC-Indic-2-Disag	Rejecting a proposal from another party (counterproposing 'status quo')	~: Would need to be done by creating a new document with the changes crossed out and/or replaced by wording such as 'status quo'.	~: Can be done in a PR review process by simply not merging it, or giving a negative review, But informal.	➲: Designed to support this UC	☞
UC-Indic-3-Sign	Formally agreeing to a proposal made by another party. Once all parties agree, a new baseline is established.	~: Can be done by marking approved changes in a new colour, or creating a new file with a file name that indicates approval, or generating a pdf that is then signed. All these are awkward and informal.	~: Can be done by the PR review process, Once all accept the PR can be merged forming a new baseline.	✓: Designed to support this UC .	☞
Category: Withdrawing a Proposal					
UC-Wd-1-Temp	Temporarily withdrawing a proposal	✗: MS-Word does not support this operation. Although negotiators might send a file with the word 'Withdrawn'.	~: Can be done by commenting on a PR or moving the PR to draft status, but awkward.	➲: Designed to support this UC	☞

Use Case Code	Summary	MS-Word	Github (+Git)	Proposed technology	In Proto-type?
UC-Wd-2-Del	Permanently deleting a proposal	✖: Not doable. Requires communication on another channel to say ignore the proposal.	➲: Deleting the pull request	✓: Designed to support this UC	✓

5.2 Expanded Use Cases for Qualitative Negotiation Tool

We validated and expanded the set of use cases discussed in Section 5.1 with data gathered during early stages of the GT study results presented in Chapter 4. In this section, we present an additional set of use cases that are more specific requirements that negotiators would benefit from in a tool to support textual negotiation; these arose as we reached closer to saturation in the latter stages of our grounded theory work as well as in open-ended comments in the survey discussed in the next chapter.

Table 8 uses the same approach as Table 7, with respect to this second set of use cases.

5.2.1 Collaboratively Editing

UC-Edit-2-Collab: By both parties, performing edits to a *live in a common document* in a live meeting.

5.2.2 Advanced Searching and Filtering

UC-Srch-4-SeePrev: Instantly going back to see previous versions of wording (either from earlier in current negotiations, or in previous contracts). The steps would involve generating a list of all previous versions, and clicking on one to see it.

UC-Srch-5-ComPrev: Comparing previous versions. This would display a list of versions. The user could click on one to generate a comparison with the current version or click on more than one to generate a comparison among all those selected.

UC-Srch-6-SeeHist: Seeing on request, for any clause, its 'history' (date of change, which party asked for it; rationale for it; what alternatives were discussed, what are the differences as compared to the current version). This differs from searching for plain text (UC-Srch-3-Versns): This use case would start by having the user select a clause. But the clause may have had a different number, and/or have had so many

changes that the exact words it contains do not exist in earlier versions, making a plain text search ineffective.

UC-Srch-7-ShowHide: Showing and hiding details such as change history, rationale, etc. to manage the complexity of the document being viewed. Change history could be requested to be shown colours or bolding of text or other highlighting format for each author or change date, and any notes could be shown (private notes, or those entered by others). These could be then hidden as needed. This provides fine-tuned capabilities as compared to UC-Srch-6-SeeHist.

UC-Srch-8-Filt: Filtering document wording to show or hide only the changes made or proposed by *certain individuals or parties*, or changed in *certain date ranges*. Upon initiating this, the list of authors, parties or dates would be shown, and upon selecting one or more of them the relevant wording would be shown in context. This expands on UC-Srch-7-ShowHide by allowing finer control over what changes are shown, and allowing hiding of parts of a document that are not relevant.

5.2.3 Indicating Agreement with a Proposal Collaboratively

UC-Indic-4-SignLive: Signing off partial or complete agreements by both parties in a common document in a *live meeting*.

Table 8: Comparison of how Word, GitHub and the proposed technology support expanded use cases of a qualitative negotiation support system

UC-Codes	Use Case	MS-Word	Github	Proposed technology	In Proto-type?
Category: Editing					
UC-Edit-2-Collab		✓: Possible when Word documents are shared using OneDrive or SharePoint, and collaborative editing is enabled,.	✗: Currently Requires external tools.	➲: To be a key feature, enabled optionally if both parties desire.	☞
Category: Searching					
UC-Srch-4-SeePrev	Seeing clause wording in previous versions, even if it has changed completely.	➲: Not available	~: Doable using git commands but requires technical expertise.	➲: To be a key capability, due to its internal data metamodel presented in Figure 27	✓

UC-Codes	Use Case	MS-Word	Github	Proposed technology	In Proto-type?
UC-Srch-5-ComPrev	Comparing clause wordings across versions	~: Word also has a compare feature that takes two files as input, but it is awkward and not instant and generates a third file. This will not support the context of this UC	✓: Two branches can be selected and compared, although the full sequence of changes cannot be easily generated.	➲: To be a key capability	✓
UC-Srch-6-SeeHist	Showing all details clause history, such as who proposed each change	~: With track changes, the author would be visible, but awkwardly as changes are made to previous changes.	~: Doable using git commands but requires technical expertise. But granularity at the level of clauses is not natively sported.	➲: To be a key capability	☞
UC-Srch-7-ShowHide	Fine tuning what is shown in history of a clause	~: Word also can hide and show, to some extent, the authors if change tracking is on. But this UC request more specific context that Word cannot handle	✗: It does not offer native feature to support this UC	✓: Designed to fully support this type of negotiation specific UC.	☞
UC-Srch-8-Filt	Showing filtered part of the agreement only based on various criteria.	☹: Word does not offer native feature to support this UC	✗: Github does not offer a full feature to support this UC. It does allow showing of only changes (hiding unchanged text), but does not allow the level of fine tuning of filtering this use case requires.	➲: Designed to fully support this type of negotiation-specific UC.	☞
Category: Indicating Agreement or Disagreement					
UC-Indic-4-SignLive	Live signing off agreements	✗: Signing signature with a draw feature is possible in MS-Word, but the context lack integrity for negotiators to sign off document by parties. For instance, a signature can be copied and pasted from one document to the other by anybody.	✗: Signing git commits enables one to distinguish between verified and unverified changes which is not in the context of this feature	✓: Designed to fully support this type of negotiation-specific UC.	☞

We observe that the results of the expanded use cases introduced in this section represent specialised operations that are particular to qualitative negotiation. As a result, these use cases are either only partially or not obtainable in either MS-Word or Github

Table 10 and Table 11 present the source and the GT theme(s) identified with any of the two set of use cases.

5.3 Use Cases Supported in Commercial Tools

As discussed in Section 2.3.4, we had investigated various commercial contract lifecycle management (CLM) tools that may be able to support qualitative negotiation. Table 9 shows our assessment of which used cases appear to be supported by the commercial tools.

Table 9: Features of some selected commercial contract lifecycle management (CLM) software that tallied with some of our initial use case sense of a negotiation tool

Use case	ContractRoom	AirSlate	Framesoft	ParleyPro	Concord	Contractsafe
Initial set of use cases						
UC-Cr-1-Init UC-Cr-2-Parse	√	√	√		√	√
UC-Edit-1-Simple	√	√	√	√	√	√
UC-Cr-3-Prop	√	√	√	√	√	√
UC-Cr-4-Condit						
UC-Note-1-Rwhole	√			√	√	
UC-Note-2-Rchange	√			√	√	
UC-Note-3-PrCmt	√				√	
UC-Note-1-Revl	√			√	√	
UC-Wd-1-Temp						
UC-Wd-2-Del	√		√	√	√	√
Uc-Snd-3-Propreply			√	√		
UC-Indic-3-Sign	√	√	√	√		
Expanded Use Cases						
UC-Srch-5-ComPrev		√		√	√	√
UC-Srch-6-SeeHist			√			√

We had difficulty in confirming and testing the presence of all the use cases (features) in Table 9 due to lack of access or limited information about some of the CLM systems. CLM software companies appear concerned that explicit description of features to non-paying customers might divulge secret of their business; access to almost all their features is based on subscription and financial commitment. Where blanks appear in this table, it therefore means we could not confirm presence or absence of the capabilities.

5.4 Core Features for a Qualitative Negotiation Tool

Based on the use cases discussed in the previous sections we now move on to present a list of the most important (i.e. core) features that we believe should be essential for a qualitative negotiation tool.

These features form requirements for a negotiation tool that current open-source technology as well as the Word-based approach to most negotiation, do not provide effectively. We have derived these from the grounded theory study presented in Chapter 4 and the use cases presented in the previous sections. Table 10 presents GT themes and the feature(s) associated with them.

We call these the core features, since they are the ones that stood out most among the GT results. We also made this list as a basis for surveying negotiators, as discussed in the next chapter. It is clear that there could be many additional features that we are not going to focus on in the rest of this thesis. Some of them, like plain editing, are needed but do not need research.

Table 11 connects the features described here with the use cases described earlier.

5.4.1 The SeePrevious Feature: Instantly seeing earlier wording

Our summary description of this feature is: Ability to instantly go back to see previous versions of wording (either from earlier in current negotiations, or in previous contracts). This summary description is how we will quickly explain the feature to end users. We use this summary (and that of the features described in the following sections in the survey in the next chapter).

This was derived from the following Use Case UC-Srch-4-SeePrev (Section 4.1.3, 4.1.5, and 4.1.7)

The feature would work as follows:

- After selecting an entire agreement, or just a set of clauses or subclauses, the user would see a button or menu item labelled “Show Previous versions”.
- Selecting this would show, alongside the current text, a non-modal panel with a scrollable list with earlier dates, each with a few words highlighting what differed (e.g. number of words or clauses added, deleted, changed), and which party or author initiated the change) in that version. This list might be lengthy.

- Selecting one or more of the dated versions, would show those versions alongside the current version (see also ComparePrevious, below).
- On any version displayed, there would also be a 'Previous' and 'Next' button pair, allowing the user to go back and forth one version at a time, with the date (and time if necessary) of each version always shown.

5.4.2 The ComparePrevious Feature: See what has changed between versions

Our summary description of this feature is: Ability to compare wording as it existed in selected different versions of the document (so additions and deletions, and who proposed them are highlighted).

This feature builds on SeePrevious by showing all the differences between two selected versions, whether they be the current version and an earlier version, or two different earlier versions. A single specific version of an entire document would be visible, not just changes to a selected clause, although the clause originally on display before the operations should be in the same location in the display.

Microsoft Word has a capability to enable comparison of versions; the output of this feature would have similarities to that, but would be more immediate (without forcing the user to search for files to compare), and more tailored to negotiation, as it would take account of the notion of different parties, rather than merely different authors.

This was derived from the Use Case UC-Srch-5-ComPrev (Section 4.1.3, 4.1.4, 4.1.5, and 4.1.7)

The feature would work as follows:

- The parties who introduced changes would have colours associated with their changes, and the changes would be marked as to whether they were deletions, additions or replacements, and whether the changes have been approved yet by other parties or not. In international negotiations there may be many colours. To ensure this would still work for the colourblind, the party names would appear if the user hovered the cursor over some text.
- Highlighting of changes could appear on both the earlier and the later version (e.g. an addition in a later version could show as a deletion in an earlier version, to make clear what going back to the earlier version would entail).

- The user could switch off the highlighting when no longer needed. In other words, the capability would not create an entirely new document (as is the case in the Compare Documents feature of Word).

5.4.3 The HideShowDetails Feature: Instantly Switch Visualization of Meta-Information On or Off

Our summary description of this feature is: "Ability to hide/show details such as change history, rationale, etc. to manage the complexity of the document being viewed."

When changes are being displayed, current technology (i.e., normally Microsoft Word with change tracking and/or special highlighting) can often make documents hard to read due to so many superimposed details, especially where parties have gone back and forth with many changes to the same text, and where each party has multiple authors. Also, note-taking about rationale and the other parties' perspectives is central to negotiation (see more about this in the PrivateNotes feature below). In the grounded theory study, it became clear that recording notes associated with pieces of text or changes would be very important to facilitate negotiation and subsequent interpretation. However, the ability to hide and show different types of notes would appear important, lest the text be too cluttered.

This was derived from the Use Cases UC-Srch-7-ShowHide (Section 4.1.3, and 4.1.7).

The feature would work as follows:

- A menu or button labelled something like "Hide/show details" would be displayed whenever any agreement text is visible, including versions presented by the ComparePrevious feature.
- If the user selects the above menu item or button, then a checkable panel of potential types of details to show or hide would appear with checkmarks by those currently shown. The detail types would include the names of parties whose proposed and/or accepted changes should be highlighted (in specific colours), private notes by the current user, notes visible to all users this party but not others, rationale for highlighted changes, response to the other party about highlighted changes, and so on.
- As the user selects or de-selects any checkbox, the text would instantly change.

5.4.4 The SeeHistory Feature: Generate Full Information About Certain Wording

Our summary description of this feature is: "Ability to see on request, for any piece of wording, its 'history' (date of change, which party asked for it; rationale for it; what alternatives were discussed)". This is intended for viewing just one clause (or small group of clauses) since the idea is that it will generate a lot of data about that clause, and the result would be unwieldy if it were applied to an entire agreement.

This was derived from the following Use Cases UC-Srch-2-Versns and UC-Srch-6-SeeHist (Section 4.1.3, 4.1.4, 4.1.5, and 4.1.7)

This would complement the above features and could simplify the interface. Rather than cluttering the text with lots of details by using HideShowDetails or ComparePrevious to show the details of all the wording and its many changes, the user could just select any particular piece of wording to bring up a non-modal dialog containing an entry for each change that has been made in the past, starting with the first versions that contained the text. The data would include the party and date of the change (or of deletion if appropriate), any rationale or other notes visible to that user about the text, and what the wording was at that time highlighted with colours to show the additions and deletions..

An alternative implementation for a touch screen would be a menu item that would display the above history information of the selected text in a side panel.

Selecting a version from the pop-up or side panel would allow the user to see the document version as it existed, in the same manner as the SeePrevious feature.

5.4.5 The FilterChanges Feature: See What an Author or Party Proposed

Our summary description of this feature is: "Ability to filter document wording so as to show or hide only the changes made or proposed by certain individuals or parties".

This was derived from the following Use Cases UC-Srch-8-Filt (Section 4.1.2, 4.1.5, and 4.1.7)

This feature would be implemented a special case of HideShowDetails: It is particularly important so as to be able to rapidly show what the user's party has managed or wants to achieve (i.e. accepted wording) or that of any of the other parties. Other wording not related to the filtering would be hidden to make the filter results stand out. The ability to instantly flip between views could help the parties understand each other or work towards agreement.

5.4.6 *The PrivateNotes Feature: Record Comments the Other Party Cannot See*

Our summary description of this feature is: "Ability to create and later examine private notes on the document that the other party cannot see."

In our grounded theory study, it became apparent that the recording of notes by all people involved in negotiation was central. Often these notes are currently recorded in private files or on paper, stored in in binders. The notes helped participants remember and develop rationale, ensure they understand the other party's position, and so on. An important use of notes can be long after an agreement is signed, when later negotiators or an arbitrator are trying to understand why certain changes were introduced. The ad-hoc nature and the difficulty of later retrieving notes are key problems that can cause delay and lead to sub-optimal negotiations, or even disputes.

This was derived from the following Use Cases UC-Note-1-Rwhole, UC-Note-2-Rchange, UC-Note-3-PrCmt, and UC-Note-1-Revl (Sections 4.1.3, 4.1.4, 4.1.6, and 4.1.7)

This feature would work as follows:

- The current user could annotate any text they see; by default, such notes would by default be private to them.
- Any notes the user has previously entered could be made visible via the HideShowDetails feature. A small unobtrusive icon might appear in the user's view of text if there is a hidden note.
- Optionally, an author could make any note visible to other users in the same party, or even to the other parties. Business rules and security constraints (such as approval being needed from the chief negotiator) would have to be developed to ensure that private notes are not accidentally revealed to the other party.

5.4.7 *The LiveSignoff and LiveEdit features: Multi-Party Interaction With an Online Document*

Our summary description of these features: "Ability to sign off partial or complete agreements by both parties in a common document in a live meeting." And "Ability by both parties to perform edits to a common document in a live meeting".

This was derived from the Use Case UC-Indic-2-SignLive

Microsoft, Google, and others now provide the ability for multiple authors to collaborate by simultaneously editing the same text. This feature would bring that capability to live negotiations:

- The parties would try to formulate wording simultaneously.
- As representatives (generally the chief negotiator) of the parties type, the wording they type would be coloured, so the other party can clearly see what the other party is suggesting.
- Representatives with appropriate authority could select any block of text with changes proposed by the other party (including an entire set of clauses), and mark it as agreed, hence 'signing off' on particular text. Alternatively, it could be marked in some other way, such as 'declined', or a further edit could be applied by the other party.

5.5 Cross Referencing negotiation tool features and their sources in our research

In this section, we connected the core qualitative negotiation features discussed in Section 5.4 to the portion(s) of the data analysis or results that gave birth to each feature, the associated use case(s) and applicable affordance.

Table 10 presents themes and sub-themes that emerged during the grounded theory results discussed in Section 4.1, and the associated features that came out of the results.

Table 10: Cross referencing of the GT themes and the derived features of a tool that will support qualitative negotiation

GT Theme	GT Sub-themes	Associated Features
Need for Tool improvement 4.1.1	Better Information and Change Management 4.1.2	FilterChanges 5.4.5
	Quick Access to Information While Negotiating 4.1.3	SeePrevious 5.4.1 ComparePrevious 5.4.2 FilterChanges 5.4.5 LiveSignoff and LiveEdit 5.4.7
	Comprehensively Search and Compare 4.1.7	SeePrevious 5.4.1 ComparePrevious 5.4.2
	Access History of Clauses in Agreements 4.1.4	SeeHistory 5.4.4 ComparePrevious 5.4.2 HideShowDetails 5.4.3
	To see Evolution of Each Clause 4.1.5	SeeHistory 5.4.4 ComparePrevious 5.4.2 HideShowDetails 5.4.3
	To be Able to Record and Review Notes 4.1.6	PrivateNotes 5.4.6

Table 11 presents the source(s) of the core features for a negotiation tool discussed in Section 5.4, and associated use case(s) to each feature. Also, building on Section 2.3.6 discussion, we classify negotiation software features and operations with the three significant types of affordances discussed earlier, as identified by Adler-Nissen and Drieschova in their discussion on diplomatic negotiation technology: shareability (shr), visualization (vis) and immediacy (imm) (Adler-Nissen & Drieschova, 2019).

Table 11: Source(s) of negotiation features, associated use case(s) and their affordance classification

Feature	Source of the feature	Ucs associated with the feature	Affordance		
			Shr	Vis	Imm
SeePrevious	4.1.3, 4.1.5, 4.1.7	UC-Srch-3-SeePrev		√	√
ComparePrevious	4.1.3, 4.1.4, 4.1.5, 4.1.7,	UC-Srch-4-ComPrev		√	√
HideShowDetails	4.1.3,4.1.7,	UC-Srch-6-ShowHide		√	√
SeeHistory	4.1.3, 4.1.4, 4.1.5, 4.1.7	UC-Srch-2-Versns, UC-Srch-5-SeeHist		√	√
FilterChanges	4.1.2, 4.1.5, 4.1.7	UC-Srch-7-Filt		√	√
PrivateNote	4.1.3, 4.1.4, 4.1.6, 4.1.7	UC-Note-1-Rwhole UC-Note-2-Rchange UC-Note-3-PrCmt UC-Note-1-Revl		√	
LiveSignoff and LiveEdit	4.1	UC-Indic-SignLive	√	√	√

Table 12 presents our assessment of the availability of the core qualitative negotiation features in the common text editors used in society. Though Microsoft Word is the only editor used by all the negotiators from whom we collected data during our research, we have extended the analysis to assess the degree to which other common editors can support qualitative negotiation as well. For brevity, on Table 12 we use the following marks to indicate the degree of support: "✓" fully supported, and "~" partially supported, "!", possible but with difficulty, and "×" not supported. In the Proposed Technology column we have marked the features fully available in the SoarNego prototype (described in Chapter 8) with an asterisk.

We have indicated that SeePrevious is only "possible with difficulty" in MS-Word; it is not 'instant' as negotiators desire, because the negotiator must find and open the required files, and may not find the files, may make mistakes (opening the wrong files), and will still have to search in each file for the relevant text. The same is true of the other word processors. GitHub, on the other hand makes the task slightly easier using the *diff* capability, but it is still not instant, and the user must have knowledge of branch names.

ComparePrevious (generating a visualization of what has been added, deleted, and changed between two versions), is a core feature of tools like Github. It can be done in desktop-based MS-Word using the "Compare Documents" feature, which generates a new document with the differences marked. Google Docs has a similar feature. The fact that the word processors generate an entirely new document is, however a drawback. Ideally ComparePrevious would temporarily annotate the document the user is editing, to inform the editing process. The user should be able to simply turn off the annotations and then continue work.

HideShowDetails is possible in word processors such as Word. In Word one selects various options in the "Highlight Changes" dialog, and one can switch off highlighting entirely. However, options are limited. For example, one cannot ask to highlight only the changes proposed by a certain author (or party), one can only ask for each author's changes to have different colours. This can be confusing when there are multiple authors in each party, or when you want to just focus on the changes one party has proposed. In our grounded theory and our analysis of the supervisor's experience at the negotiation table, we noted a certain amount of frustration with the track changes capabilities, some of them relating to this. Another weakness of Word is that it is not possible to hide changes before or after a certain date (or starting/ending with a certain version), which can be extremely useful when negotiators want to assess progress.

The SeeHistory feature, which should allow the detailed history of any wording to be displayed is not available at all in word processors. With expert use of git, it can be accomplished, but this requires a technical expert.

FilterChanges builds on HideShowDetails but is in fact even less supported than HideShowDetails in other tools.

All the word processors allow note taking as comments, that are visible by all who can see the document. However, none of them allow secure selective visibility of PrivateNotes.

The online word processing tools allow the LifeEdit feature. LiveSignOff, i.e. clearly tagging in a live meeting which text is now approved by multiple parties, is not available.

Table 12: Analysis of suggested features across popular text editing technologies

Feature	MS-Word Offline	MS-Word Online	Git / GitHub	Google Docs	Dropbox paper	LibreOffice	WPS Office	Proposed technology
SeePrevious	!	!	~	!	!	!	!	✓*
ComparePrevious	~	~	✓	~	✗	✗	✗	✓*
HideShowDetails	~	~	~	~				✓
SeeHistory	✗	✗	!	✗	✗	✗	✗	✓
FilterChanges	✗	✗	!	✗	✗	✗	✗	✓
PrivateNotes	✗	✗	✗	✗	✗	✗	✗	✓
LiveEdit	✗	✗	✗	✗	✗	✗	✗	✓*
LiveSignoff	✗	✗	✗	✗	✗	✗	✗	✓

5.6 Conclusion Regarding Use Cases and Features

In this chapter, we presented set of initial and expanded use cases that were uncovered during our research. In Table 7 and Table 8, we described application of these use cases in Microsoft Word, the only tool used by all the negotiators that we collected data from during our research. We also compared the use cases to those possible in Github, a version control tool for managing code (which is also subject to negotiation among software developers), although this would not be appropriate for the non-technical negotiators we are targeting, but support some of the use cases; our proposed technology, and availability of these use cases in SoarNego, a

prototype we have developed that we will present in Chapter 8. In Table 9, we presented our assessment of availability of the use cases in commercial CLM tools.

Later in the chapter, we discussed features that were derived from the use cases, and their source being grounded in our research as presented Table 10 and Table 11. In Table 12, we analyzed availability of the discussed features in the popular text editors.

In the next chapter, we will present our survey results that corroborate desire of negotiators of the use cases and features described in this chapter, in a tool that will effectively support qualitative negotiation. The survey results also quantify negotiators' difficulties and priorities, which suggest some use cases and features are more important than others.

Chapter 6 Results of the Survey

In this chapter, we present the results of the data collection and analysis done during the survey exercise discussed in Section 3.2. The result validates and expands on the GT results presented in Chapter 4, and also validates aspects of our list of features presented in the last chapter.

Section 6.1 presents the demography of the negotiators that participated in the survey. While Section 6.2 to Section 6.7 presents the degree to which negotiators desire the features we have identified. In the survey, negotiators also expressed desires for certain other features beyond those that had been encountered in the results discussed in previous chapters.

6.1 Participant Demography (Survey Questions 23-25)

Figure 14 shows that more than 88% of the participants rated themselves to have at least medium experience in conducting negotiations. Figure 15 shows that two thirds of participants have fulfilled the role of chief negotiator while conducting negotiations.

Figure 16 presents the participants' domains of negotiation experience. Almost half of them have experience in labor negotiation union or management side (one of these had experience on both sides); 37% had experience in commercial negotiation as vendor or buyer; and 17% had experience in intergovernmental negotiations, such as in trade or climate change. Several had experience in multiple domains.

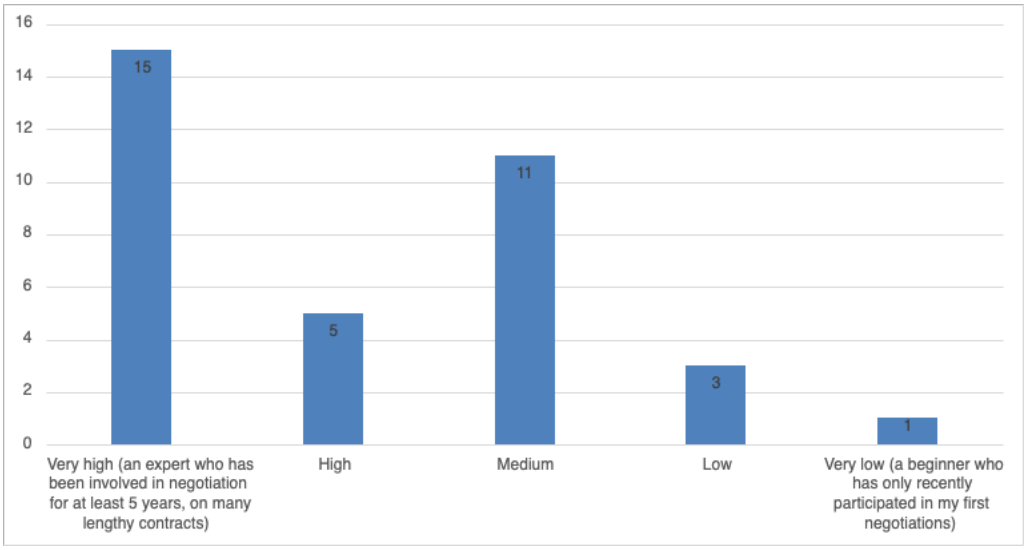

Figure 14: Participants' level of experience in conducting negotiations (Q25; n=35)

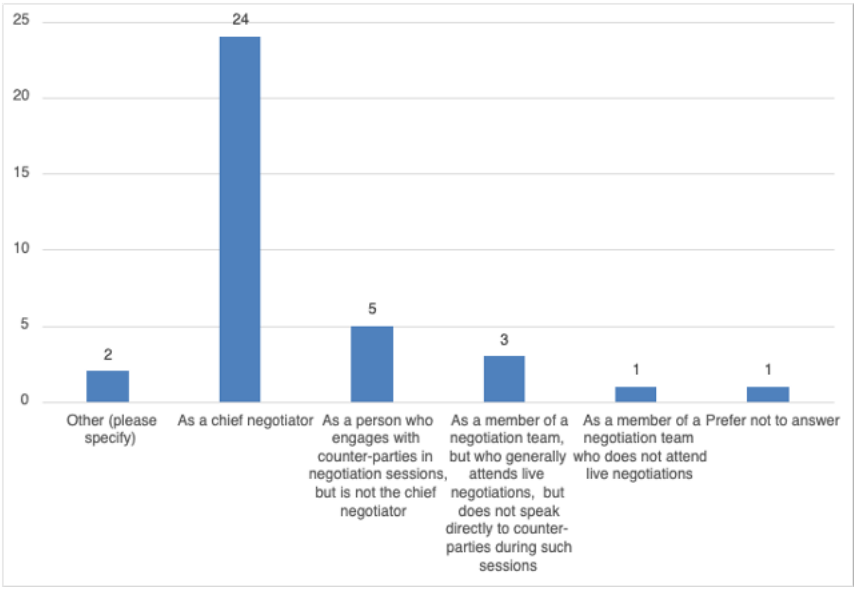

Figure 15: Participants' primary role while conducting negotiations (Q23; n=36)

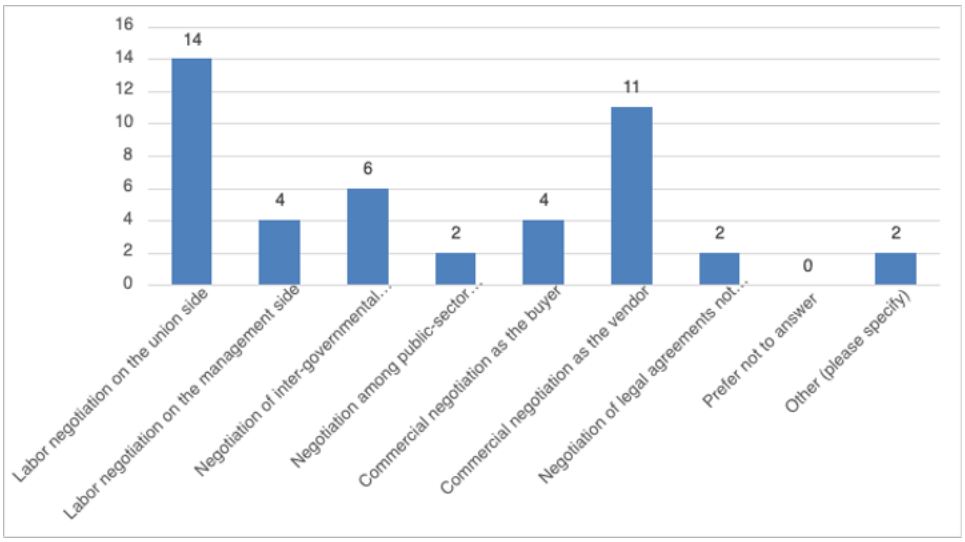

Figure 16: Participants' domain of negotiation experience (Q24; n=35; multiple responses allowed)

6.2 Software Features to Potentially Benefit Negotiators (Survey Question 14)

In Question 14, we explored potential features that our grounded theory interviewees suggested would be of benefit to them in a negotiation system. Details of the features were presented in Chapter 5.

We asked our participants to rate potential features with the question: "As a negotiator, to what extent do you think you would benefit from each of the following potential features in negotiation software?" Potential responses were.

- "Not likely (and would prefer not to have it available)", which we rated -1;
- "Not likely (but does not bother me)" rated 0;
- "Possibly" rated 1;
- "Probably" rated 2; and
- "Definitely" rated 3.

The numeric rating was not visible to the participants, but we applied it to the data to enable basic descriptive statistics. We provided several levels by which the participants could rate their positivity towards the features. We chose a rating of zero for the case where the software does not interest them, and three higher levels of positivity. However, since there is a risk in any software that it can get cluttered or confusing as too many features are added, we also enabled respondents to indicate that they actually had a *negative* view of the feature. The negative rating might also be selected if a respondent was perhaps wary of the feature due to a risk such as confidentiality breach.

We received 36 answers to this question, and the result is presented in Table 13. The grey shaded cells in Table 13, signify the most interesting results, with the highest and lowest means and standard deviations.

In the right columns of Table 13 we classify negotiation software features and operations with the three significant types of affordances discussed in Section 2.3.6, as identified by Adler-Nissen and Drieschova (Adler-Nissen & Drieschova, 2019) in their discussion on diplomatic negotiation technology. The categories are shareability (shr), visualization (vis) and immediacy (imm).

We presented in the survey our summary of each proposed feature (as described in the last chapter), so the respondents could understand the intent, without having to read too much detail.

Table 13: Extent to which respondents felt each proposed feature would be useful, in question 14, n= 36. Scale ranges from -1 (would negatively impact usefulness); 0 (not likely to be useful) to 3 (definitely useful)

Feature code and summary	Mean	St. Dev.	95% Confid-ence Interval of mean		Top two (Defin-itely or prob-ably)	Bottom two (Not likely, but would prefer not to have it available or does not bother me)	Affordance		
							shr	vis	Imm
SeePrevious: Ability to instantly go back to see previous versions of wording	2.58	0.73	2.34	2.83	91%	2.7%		√	√
ComparePrevious: Ability to compare wording as it existed in selected different versions of the document	2.47	0.81	2.20	2.75	85.8%	2.8%		√	√
HideShowDetails: Ability to hide/show details such as change history, rationale, etc. to manage the complexity of the document being viewed	2.36	0.90	2.06	2.67	77.8%	2.8%		√	√
SearchPrevious: Ability to search through previous versions	2.36	0.87	2.07	2.65	80%	2.7%		√	√
SeeHistory: Ability to see on request, for any piece of wording, its 'history'	2.19	0.98	1.86	2.53	72.2%	5.6%		√	√
FilterChanges: Ability to filter wording by certain individuals or parties	2.14	0.99	1.80	2.47	75%	8.3%		√	√
PrivateNotes: Ability to create and later examine private notes	1.92	1.32	1.47	2.36	66.7%	19.5%		√	
LiveSignoff: Ability to sign off changes in a live meeting.	1.50	1.48	1.00	2.00	55.5%	25%	√	√	√
LiveBothEdit: Ability by both parties to perform edits to a common document in a live meeting.	1.14	1.61	0.60	1.68	44.4%	41.6%	√	√	√

Table 13 is sorted from top to bottom in the descending order of means. Any mean value close to 2 or above in Table 13 signifies such a feature is perceived to be probably or definitely of benefit by the respondents; this is the case for all features except the bottom two, where the

respondents may be concerned about the need to consult with other stakeholders about wording, rather than instantly editing back and forth, and instantly agreeing.

A low standard deviation (less than 1) in Table 13 shows negotiators are in agreement; this is the case for all features except the bottom three, which are clearly more controversial.

Some other interesting results are:

- SeePrevious: Dominantly, negotiators believe they would benefit from this feature being the highest (definitely) ranked feature. Furthermore, no negotiator indicated they prefer not to have it available in a negotiation system. This feature will aid the process of negotiation by bringing past information to the fingertips of the negotiators.
- ComparePrevious: Similarly, negotiators also want to be able to actively compare the wordings as they existed in selected different versions of the document. Access to such information may guide the eventual wording of the contractual text as it would appear following completion of negotiations.
- LiveSignoff: Also has a high standard deviation (1.48) which implies disagreement in the opinions of the negotiators. We can also see that 55.5% would like the feature and 25% do not like it. There may be various reasons for this, discussed below.
- LiveBothEdit: Negotiators also indicated in disagreement (highest standard deviation, 1.61) whether they would want the benefit of this feature.

Results in Table 13 suggest negotiators want features that would enhance their knowledge of the negotiation subject, give them a timely access to pieces of information needed for negotiations and allow them to manage such information to their best advantage. On the contrary, they have divided opinions of features that would allow them to make commitments such as: signing off complete or partial agreement (hence risking creating a legal commitment prematurely); or editing a common document in a live meeting. Nevertheless, majority of negotiators are in support of these features. Non-popularity of these features may be subject to the following factors:

1. Degree of delegated power and autonomy of a chief negotiator to act in a capacity to commit or act on behalf of the represented organization in a live negotiation setting.

2. Quest for more time to reflect and reconsider details of the decision taken at the negotiation table before making a sign-off commitment.
3. The need to consult the senior management or the members of the organization.
4. Concern about accidentally committing something incorrect, or even having the other party make a change that is not noticed.

6.3 Negotiators' Satisfaction with the Negotiation Process (Survey Question 18)

In Question 18, we investigated the satisfaction or dissatisfaction experience of the negotiators in carrying some of the activities that are involved in the process of negotiation. We asked for how satisfied or dissatisfied these negotiators are with aspects of their current negotiation process. We rated these on a scale (-2 to 2) with the values not visible to the respondents:

- very dissatisfied, rated -2;
- dissatisfied, rated -1;
- neither satisfied nor dissatisfied, rated 0;
- satisfied, rated 1, and
- very satisfied, rated 2.

The activities listed are some reframed grounded theory interview questions and the results from the grounded theory.

Table 14: Extent to which respondents are satisfied with various processes, survey question 18, n = 36. Scale is from -2 (very dissatisfied) to 2 (very satisfied)

Process	Mean	St. Dev.	95% Confidence Interval of mean		Top two (Very satisfied or satisfied)	Bottom two (Very dissatisfied or dissatisfied)	Affordance		
							shr	vis	Imm
ProposalTransmission: The process you use to transmit proposals to the other party	0.97	0.65	0.75	1.19	83.3%	2.8%	√	√	√
DocSharingTool: The tool(s) you use for sharing documents among team members that you selected in Q15 [e.g. Google Drive, Sharepoint]	0.89	0.82	0.61	1.17	72.2%	5.6%	√	√	√
VirtualNegotiation: Your ability to conduct virtual (online) negotiations with the other party using tools such as Zoom or Microsoft Teams.	0.77	0.81	0.49	1.05	69.4%	8.3%		√	√
InformationManagement: Your ability to manage information other than the contract document itself, such as the negotiation mandate; meeting agendas, personal notes, and resource material to reference or to back up proposals during a live negotiation session.	0.36	0.87	0.07	0.65	50.0%	19.4%		√	√
ClauseEvolutionRetrieval: Your ability to retrieve information about clauses and agreements made between parties in the past, including the evolution history of clauses	0.11	1.04	-0.24	0.46	44.4%	36.1%		√	√
PastNoteAccess: Your ability to access notes taken during past negotiations as a reference and in preparation for the future negotiations	0.03	1.00	-0.31	0.37	30.6%	33.3%		√	√

We received 36 answers to this question, and the result is presented in Table 14. The bold keywords in column 1 are our own codes for the capability, which were not visible to the respondents.

Any mean close to 1 or above in Table 14, highlighted in grey, suggests negotiators are satisfied with that aspect of their current negotiation process. There were no means below zero, which would have signified overall dissatisfaction, but some were close to zero, also highlighted in grey, with a third of people being dissatisfied.

If the standard deviation is lower than one, this suggests negotiators were in agreement.

The grey shaded cells in Table 14, signify the most interesting results, as follows:

- **ProposalTransmission**: Negotiators are clearly satisfied with the process they use to transmit proposals to the other party. Current processes such as transmitting by email seem to work well-enough, so focusing on features for improving transmission seems not to be a priority at this time.
- **DocSharingTool**: Negotiators are also satisfied with the document sharing tools they use internally among team members, such as shared repositories. This suggests that if multiple files have to be managed, the existing tools work well. However other results suggest that it may be better to avoid having to manage multiple files.
- **VirtualNegotiation**: Negotiators seem reasonably satisfied with their current choice of tool, if they perform virtual negotiation. Features to support this should therefore will not be the focus of our design work in this thesis.
- **InformationManagement**: Only 50% of negotiators are satisfied with tools for this suite of functions, due to the management of multiple files of various types. This finding is consistent with our grounded theory study, where we heard complaints about having to keep track of large numbers of files electronically and on paper. It reinforces our discussion presented in the last chapter about the importance of use cases that allow comprehensive management of all aspects of negotiation data including versions of the the text and proposals, as well as annotations.
- **ClauseEvolutionRetrieval**: This capability had a low mean (0.11, close to 0), with over a third being dissatisfied. This confirms our other data showing that negotiators want better ways to retrieve information about the evolution of the agreement. It hence reinforces our

decision that a negotiation too should provide features to help with this. The slightly-higher-than-1 standard deviation (1.04) indicates there is disagreement about this.

- **PastNoteAccess**: Many negotiators indicated dissatisfaction with their ability to access past notes. This has the lowest mean (0.03, close to 0) compared to other aspects of the negotiation. Negotiators do not have a tool to link notes taken during negotiation and the agreed content of the contractual text. Many times, notes can only be accessed by manually flipping through pages in a binder. This reinforces our conclusions from the previous chapter that formal note management should be central features in a negotiation tool.

6.4 Changes that the Pandemic Introduced into Negotiation (Survey Question 19)

In Question 19, we investigated change of experience as regards transition of negotiations from in-person to online because of Covid-19. We asked how negotiators' experience changed with respect to various aspects of negotiation. We later rated these on the scale (1 to 5):

- much better online, which we rated as 1;
- somewhat better online, rated 2;
- not much difference, rated 3;
- somewhat better in person, rated 4; and
- much better in person, rated 5.

Table 15: Negotiators' experience as regards change from in-person to online negotiation during the pandemic, survey question 19; n 32. Scale is from 1 (much better online) to 5 (much better in person)

How did your experience change regarding the following due to the move from mostly in-person negotiations to mostly online-negotiations during the Covid-19 pandemic?	Mean	St. Dev.	95% Confidence Interval of mean		Somewhat or much better in person	Much or somewhat better online
NonVerbal: Ability to read facial expressions, body language or other non-verbal information from the other party about whether they like or dislike something	4.23	1.12	3.82	4.64	77.4%	9.7%
Effort: Effort to reach agreement	3.16	1.24	2.71	3.62	41.9%	32.3%
Quality: Quality of agreement following negotiation	3.10	0.88	2.77	3.43	22.6%	12.9%
Speed: Speed of reaching agreement	3.10	1.16	2.67	3.52	32.3%	32.3%
Cost: Overall cost of the process of reaching agreement	2.29	1.07	1.90	2.68	12.9%	61.3%
Time: Finding times to meet	2.13	1.09	1.73	2.53	6.5%	58.1%

We received 31 answers to most of these questions, and the results are presented in Table 15. Any mean close to 1 in Table 15 means negotiators prefer online negotiation processes; and a mean close to 5 means they prefer in-person negotiations for these aspects. We chose not to use negative numbers for one end of the scale, with zero as neutral, since neither mode seems inherently negative.

Standard deviation near or lower than 1 implies negotiators were in general agreement in their opinion of that aspect of the negotiation process. The grey shaded cells in Table 15, signify the most interesting results:

- **NonVerbal**: Dominantly, the inability to read nonverbal gestures is the greatest challenge faced during virtual negotiation. Negotiators want to be exposed more clearly to the subtle emotional cues the other party members might intentionally or unintentionally present during discussions. Such cues might suggest various levels of consent, dissent, uncertainty, confusion or questioning. The only way to read such cues and hence reduce unnecessary back-and-forth in a negotiation session is by matching non-verbal gestures such as: nodding of head, facial expression, snapping of fingers, and other aspects of body language

with whatever is stated verbally. The challenge with virtual negotiation is you can only read limited non-verbal gestures even when all negotiators are compelled to open their video during negotiation.

- **Cost**: The overall cost of the process of reaching agreement via virtual negotiation is cheaper, in the opinion of the respondents. This is likely mostly because people do not need to travel from one location to the other.
- **Time**: Negotiators find is easy easier to find mutually agreeable times to hold negotiation sessions virtually, although many negotiations were nonetheless suspended in part because of people not wanting to give up access to the non-verbal cues as discussed above.

6.5 Preferred Markup Formats for Proposed Changes (Survey Questions 1-4)

Questions 1 through 4 recorded participants' preferred way of marking up documents when indicating proposed additions and deletions. The answers to these questions might not be as impactful as the above questions in terms of design of negotiation software, but can nonetheless help inform design.

Table 16 shows preferences for marking up text to be added, both within the team, and later when presented to the other party. Most negotiators use multiple techniques, either simultaneously, or in different contexts (e.g. depending what the counterparty wants to do), hence the percentages add to far more than 100.

It is clear that tracking changes is by far the most widely used approach, especially within the team; the percentage drops from 69% to 53% when the proposals are presented to the other party.

Table 16: Usage of markup techniques for indicating added text from Q2 (n=35) and Q4 (n=36)

Technique	% For discussion within the team	% When presenting to the other party
Track changes	69%	53%
Word processor commenting feature	51%	36%
Boldface	26%	28%
Colored text	26%	22%
[Brackets]	11%	17%
Underlining	11%	8%
Italics	6%	3%
* Asterisks before changes	9%	0%

Preferences for marking up proposed textual deletions were broadly similar, as can be seen in Table 17. The key difference is that striking through text is a one of the three top markup methods, as might be expected.

Table 17: Usage of markup techniques for indicating text to be deleted from Q1 (n=34) and Q3 (n=35)

Technique	% For discussion within the team	% When presenting to the other party
Track changes	62%	54%
Word processor commenting feature	56%	40%
~~Striking through~~	62%	60%
Colored text	9%	14%
* Asterisks before changes	6%	3%
Boldface	6%	3%
Italics	3%	0%

The results above will help inform the interaction design of tools. It might be sensible to allow changes to be highlighted in a variety of ways, set though a preference pane, so as to satisfy more negotiators.

To further clarify how track changes is used, we asked further questions about that specific feature. In Question 12, we asked the extent to which track changes is used when gathering team members' suggestions. Participants could respond on a scale from 'We don't use track changes for this', which we rated as 1, to 'we use track changes to mark all or almost all proposed changes'.

A total of 69% gave the top two answers (i.e. that they use it), and 31% gave the bottom two answers (they do not use it). This is generally consistent with results from Questions 1-4 regarding how markup is done, as presented in Table 16 and Table 17.

In Question 13 we used the same scale to ask how much track changes is used when presenting proposals to the other party. In this case 56% gave the top two answers and 42% gave the bottom two. This is again generally consistent with Table 16 and Table 17.

6.6 Patterns of Collaboration (Survey Questions 5-11 and 15-17)

We asked several questions to ascertain various details about how collaboration typically happens.

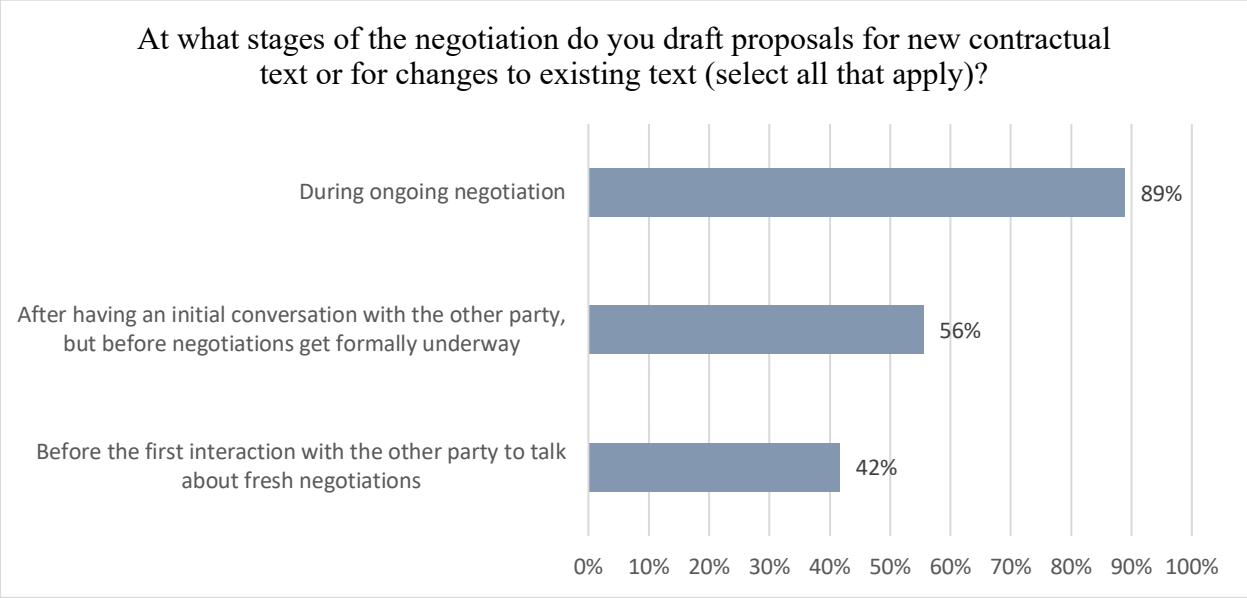

Figure 17: Stages at which negotiators draft proposal

In Question 5, we asked at what stages during negotiation are changes *drafted*. 42% reported doing this before meeting the other party; this is the most confrontational mode, i.e.

preparing demands. 56% said they draft proposals after having initial conversations with the other party. And 89% report doing this during ongoing negotiations, which reflects either collaboration or give-and-take in response to what the other party is proposing. Clearly these numbers add to more than 100% because negotiations use all three methods in different contexts.

The results of this can help inform aspects of the user interaction of a negotiation tool.

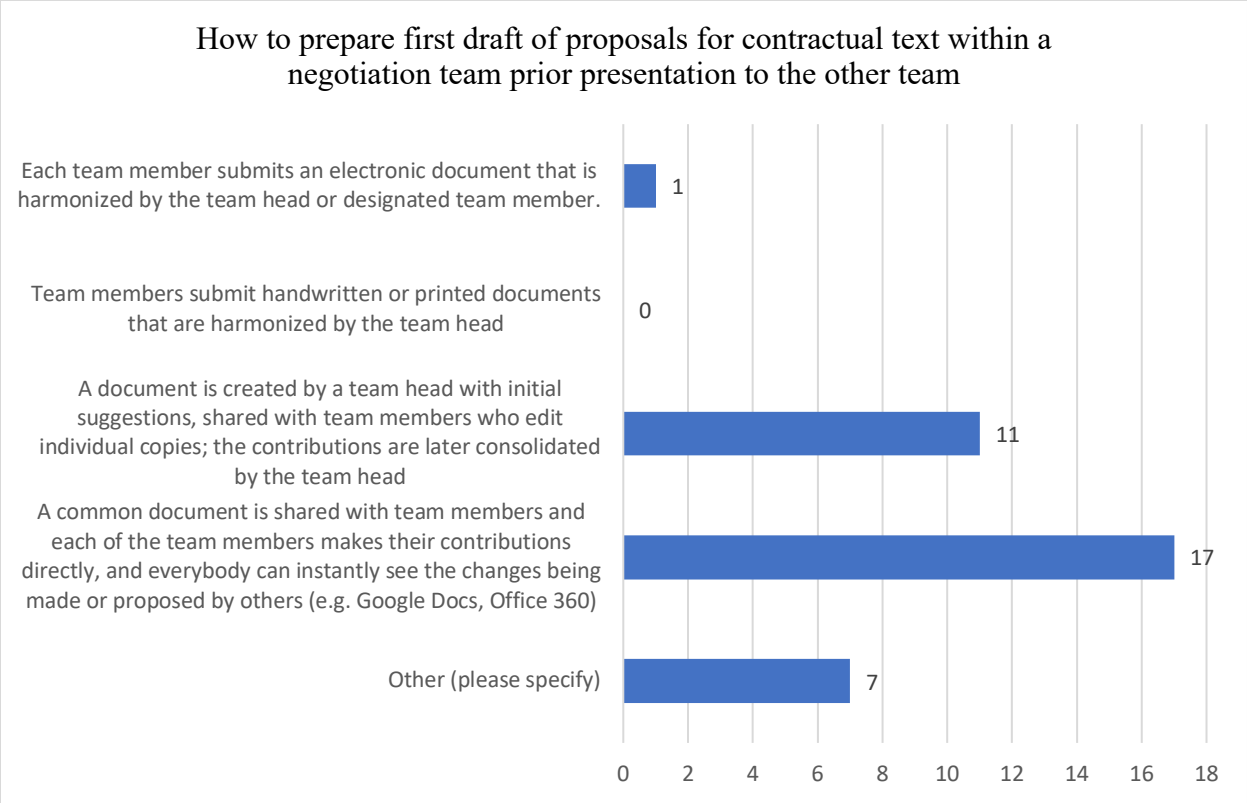

Figure 18: How first draft of the contractual text is produces within team

Table 18: Open-ended 'Other' responses to survey question 6, regarding how proposals are gathered within the negotiation team of a party.

Respondent response	Analysis
A common document is shared with team members and each of the team members makes their contributions directly. Drafts are sent back and forth between each party and changes are discussed/accepted/rejected.	A document is directly edited by the team members
1 person makes a first draft; then we discuss and edit collectively, using a shared doc (Sync)	
We are working with an adjusted standard agreement. Change proposals are collected in an excel sheet. One responsible then drafts the agreement based on the final redlining requests.	A document editing reviews are suggested by team member, edited by only the team head or designated member and final reviews are validated by all the team members
a document is created by the team head with input from team members. Once the document is complete the team members will review and approve.	
A common document is shared with team members by the head negotiator. A meeting is organized to discussed (verbally) the proposal and if the document need to be amended it's done by the head negotiators. Most time we will validate the changes with the team by circulating an amended document.	
This occurs in the Contract Committee, a group related but different than the Negotiating Team.	How it is done not mentioned
first and second options are used.	

In Question 6 we asked about how proposals are gathered within the team. Approximately half the participants indicate they share a common document among team members, and that they make their contributions directly in this document, which can be seen by all members instantly. About a third of participants said a common document is shared with them also, but they edit individual copies that are consolidated later by the team head. Various other strategies are used by the other participants, such as listing changes in an Excel spreadsheet.

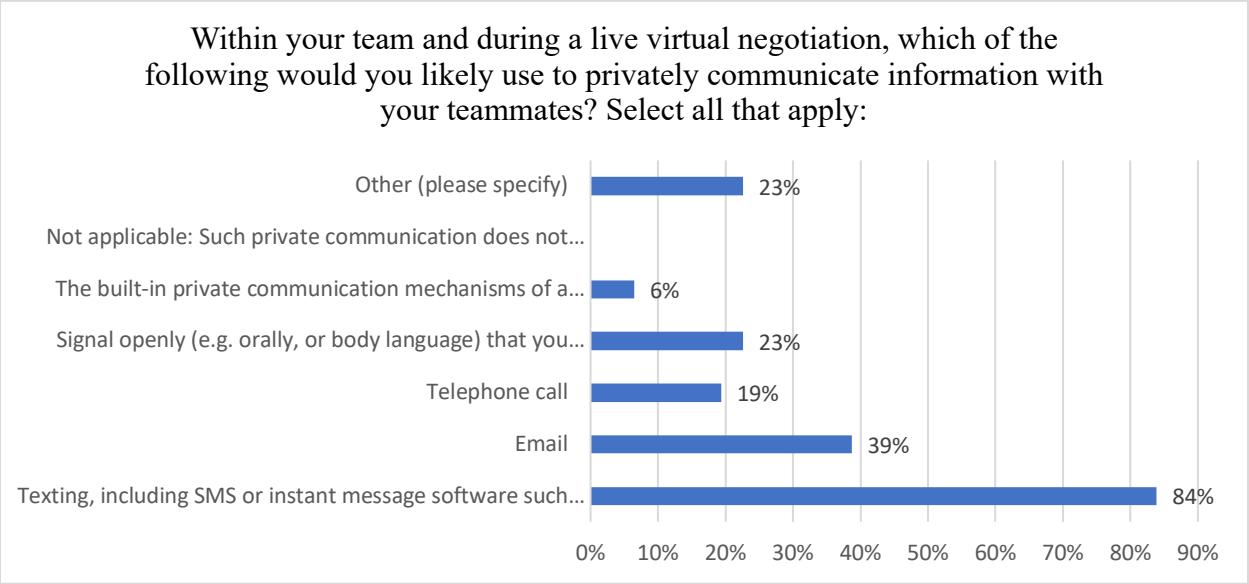

Figure 19: Means of private communication within a team while in negotiation with the other party or parties

In Question 7, we asked about how communication among team members occurs during live virtual negotiation, so as to not disturb interaction with the other party, but to be able to communicate privately.

A total of 84% of participants use texting, SMS or instant message software, 39% use email, 19% use a telephone call, 23% signal openly (e.g. orally, or with body language) that they want to switch to a private caucus, and only communicate private information while in such a private caucus; and 6% use the built-in chat box of a commercial negotiation tool. Clearly these numbers add to more than 100% because negotiators use all these methods in different contexts.

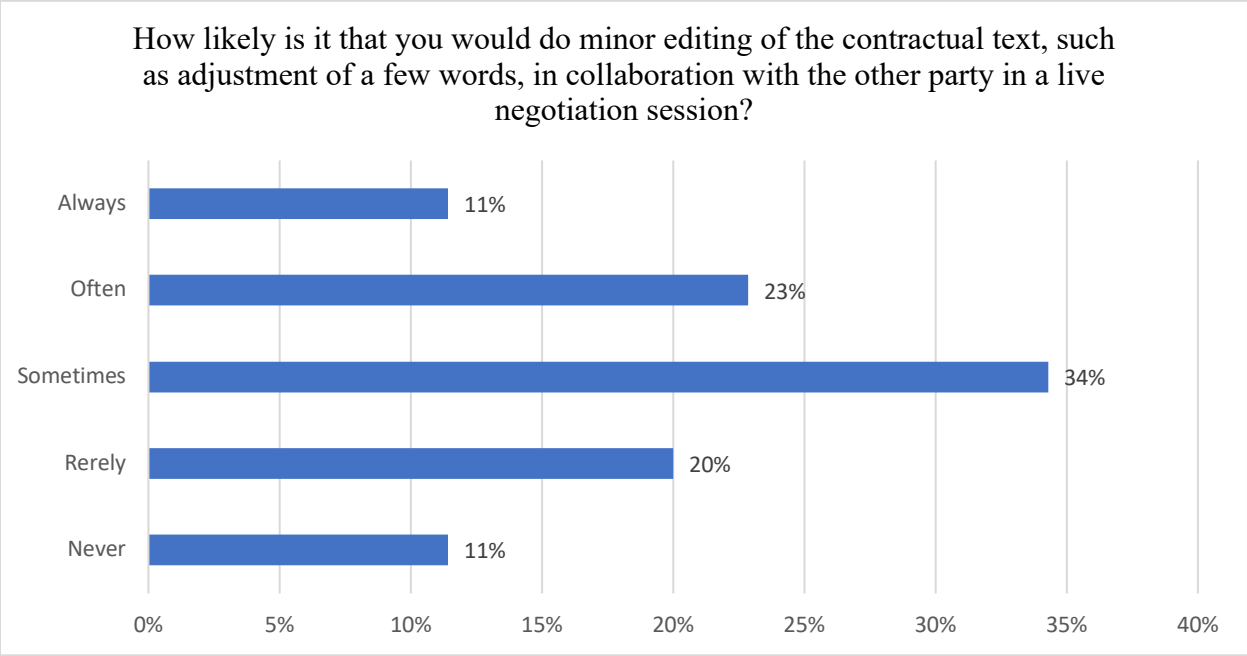

Figure 20: Possibility of a minor editing of the contractual text during live negotiation

In Question 9, we asked how likely is it that negotiators would do minor editing of the contractual text, such as adjustment of a few words, in collaboration with the other party in a live negotiation session. Participants said the following: 11% never, 20% rarely, 34% sometimes, 23% often and 11% always. This result suggests minor editing of contractual text is an activity that can take place in a negotiation session.

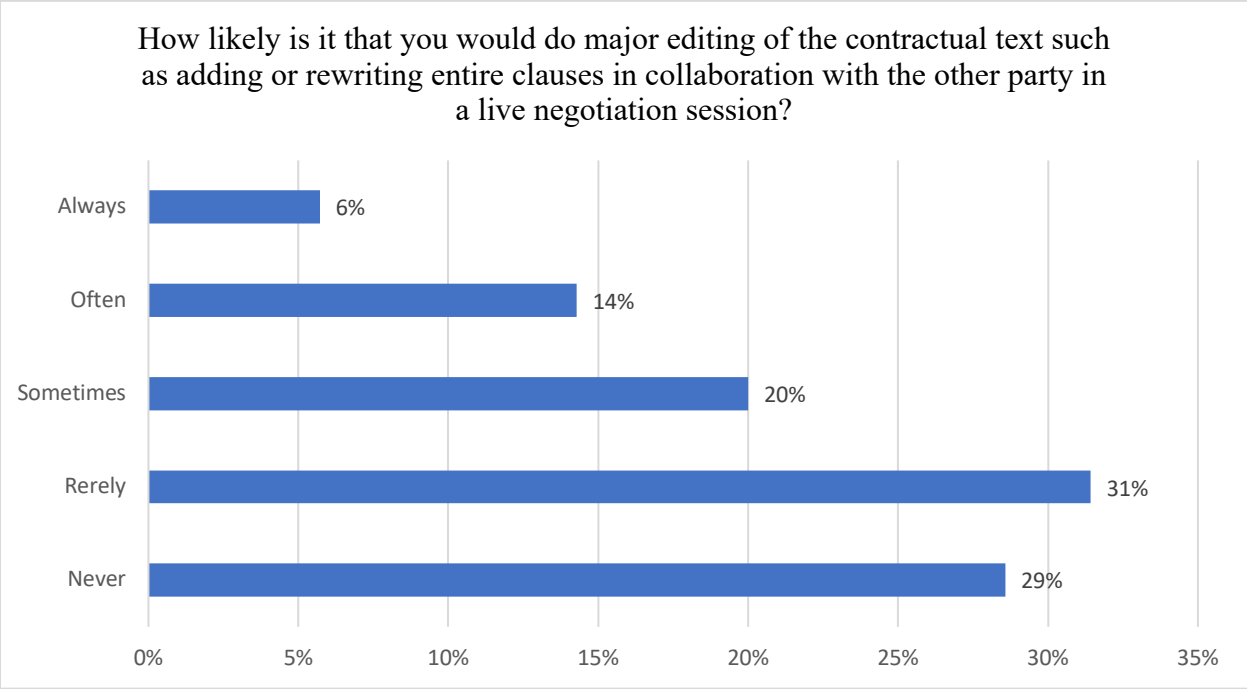

Figure 21: Possibility of a major editing of the contractual text during live negotiation

In Question 9, we asked a similar question about how likely they would do major editing of the contractual text such as adding or rewriting entire clauses in *collaboration* with the other party in a live negotiation session. Participants said the following: 29% never, 31% rarely, 20% sometimes, 14% often and 6% always. This shows that negotiators feel much less confident about enabling major live editing in comparison to minor editing. This could be because the team may not have delegated power to take binding decision from the organization they represent, or because major edits are more likely to need careful review when the other party is not present.

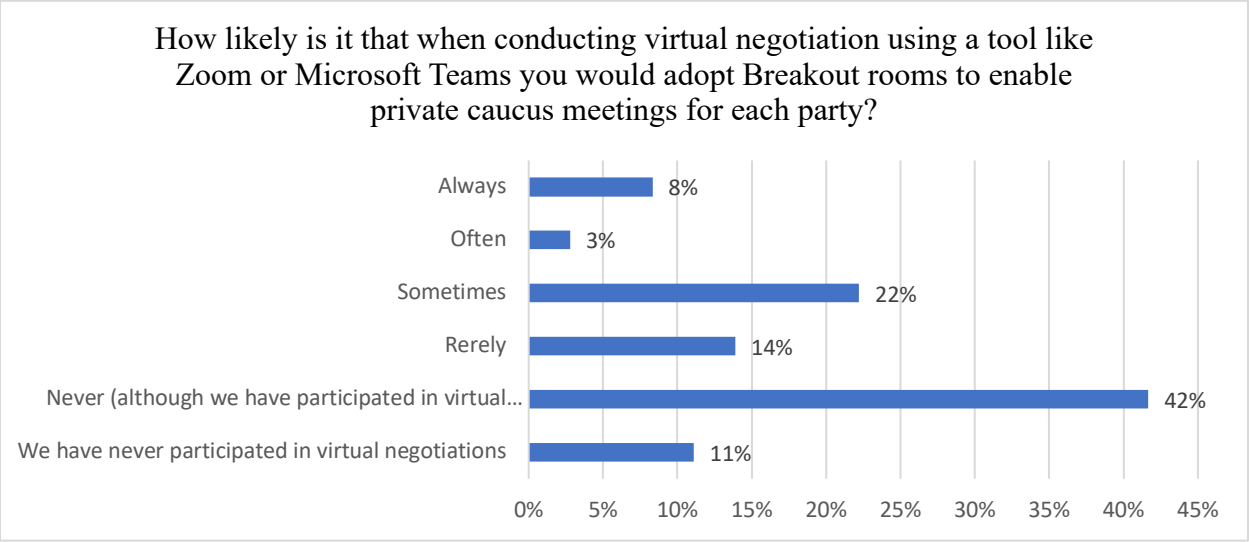

Figure 22: Possibility of calling caucus meeting during virtual negotiation

In Question 10, we asked how likely is it that when conducting virtual negotiation using a tool like Zoom or Microsoft Teams the negotiators would adopt breakout rooms in those tools to enable private caucus meetings for each party (as opposed to disconnecting and using a totally different mode or tool session.

11% said they had never participated in virtual negotiation, 42% said they had never used breakout rooms but had participated in virtual negotiation. Other participants said the following: 14% rarely, 22% sometimes, 3% often and 8% always.

The takeaway from this is that participants may be wary of breakout rooms because they prefer a channel that they might consider more secure.

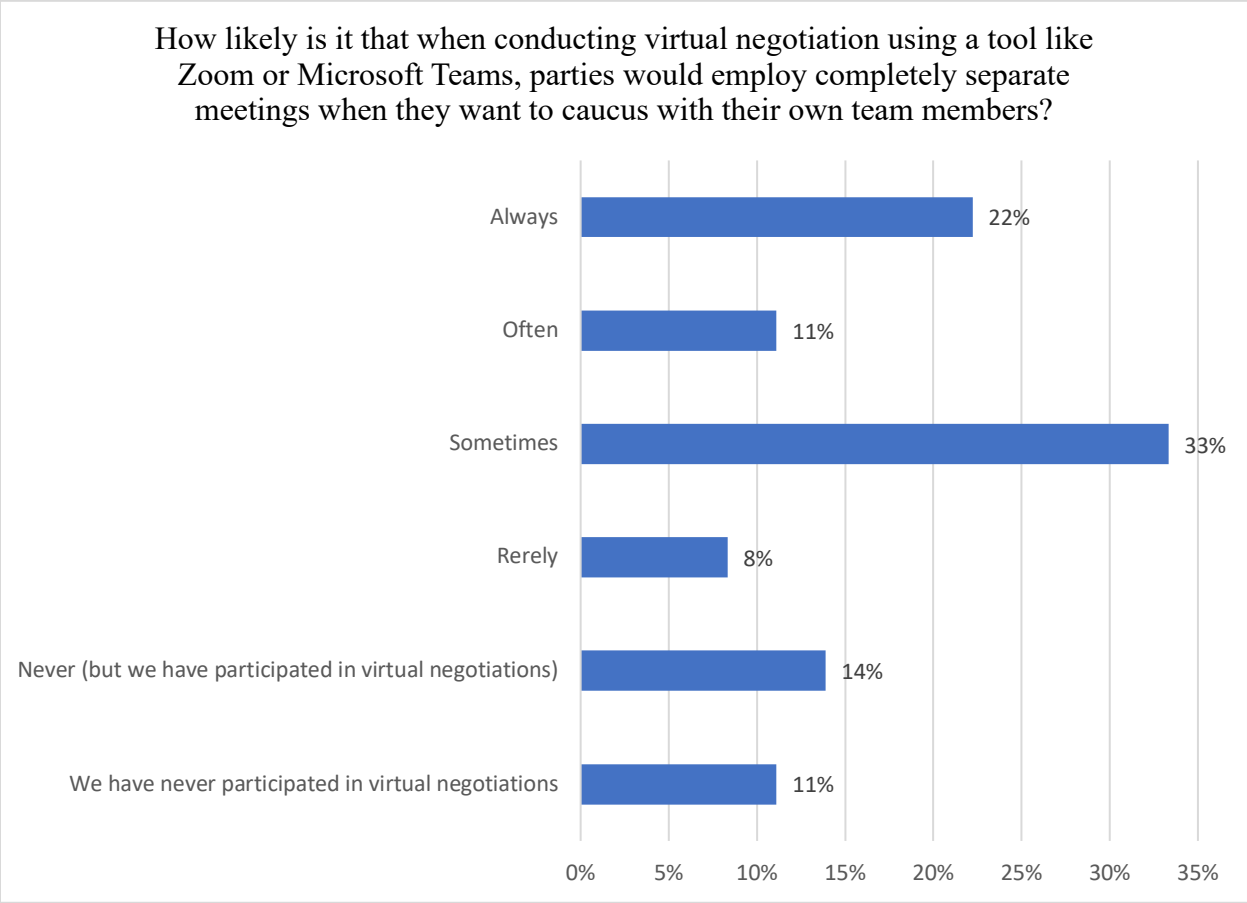

Figure 23: Possibility of calling completely separate caucus meeting during virtual negotiation

In Question 11, we asked participants, how likely is it that when conducting virtual negotiation using a tool like Zoom or Microsoft Teams, parties would employ completely separate meetings when they want to caucus with their own team members; this is the close to the inverse of the last question. 11% said they had never participated in virtual negotiation, 14% said they had never used separate virtual meetings to caucus but had participated in virtual negotiation. Other participants said the following: 8% rarely, 33% sometimes, 11% often and 22% always.

Discussion in a caucus meeting needs confidential and the results of this and the last question suggest a desire to hold such meetings in a completely separate environment.

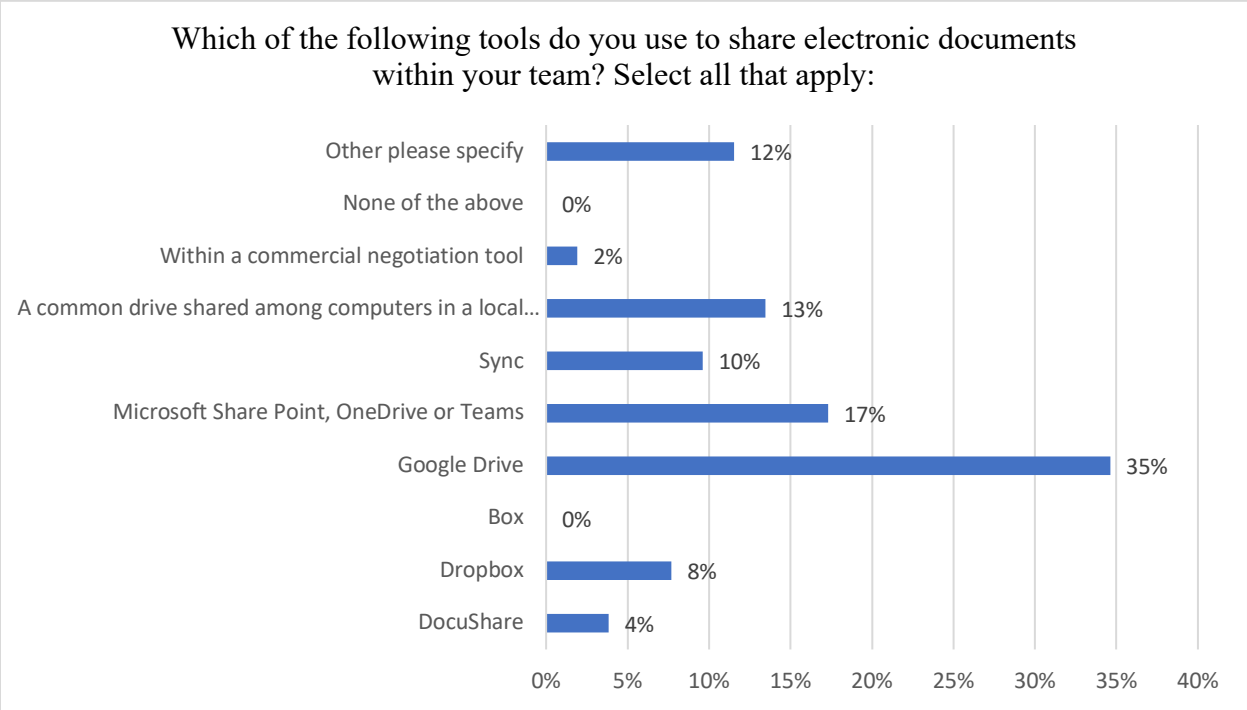

Figure 24: Tools use by negotiators to share electronics document with their team members

In Question 15, we asked participants for the tools they use to share electronic documents within their team. The following are their responses:

- DocuShare 4%
- Dropbox 8%
- Google Drive 35%
- Microsoft Share Point, OneDrive or Teams 17%
- Sync 10%
- common drive shared among computers in a local network 13%
- Within a commercial negotiation tool 2%.

Other open responses were: “email” (2 responses), “email (outlook)” , “I email word copies to them”, “File Management Software – Worldox”, and “Electronic doc management system of Swiss Fed. Govt”.

The use of document sharing tools is standard, and any negotiation software would need to consider working with such tools, or embedding the capability within the tool directly.

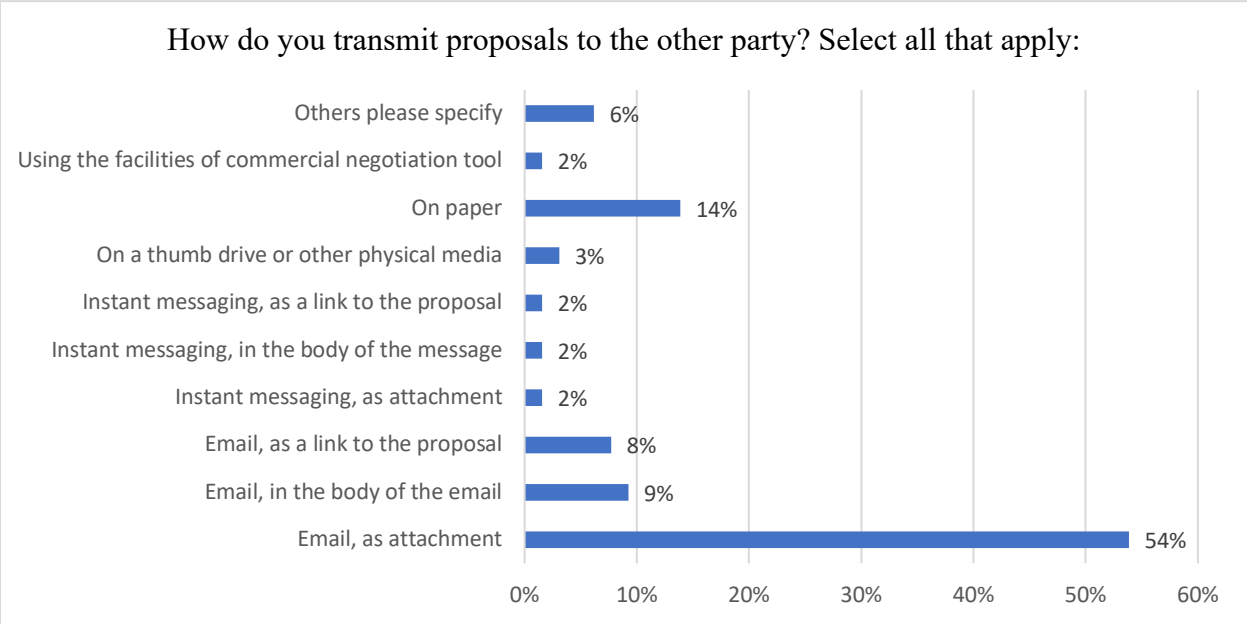

Figure 25: How proposals are transmitted to the other party

In Question 16, we asked how participants transmit proposals to the other party. They said the following, with multiple possibilities allowed:

- 97% Email, as attachment
- 17% Email, in the body of the email
- 14% Email, as a link to the proposal
- 20% on paper

They also indicated a few other methods in the the 'Other' responses, which were : "Via diplomatic courier, depending on content", "Google drive", "Secure ftp for final CA" and "online document sharing platform".

It is clear that the pandemic has shifted transmission from paper to more use of electronic modes, with email being dominant.

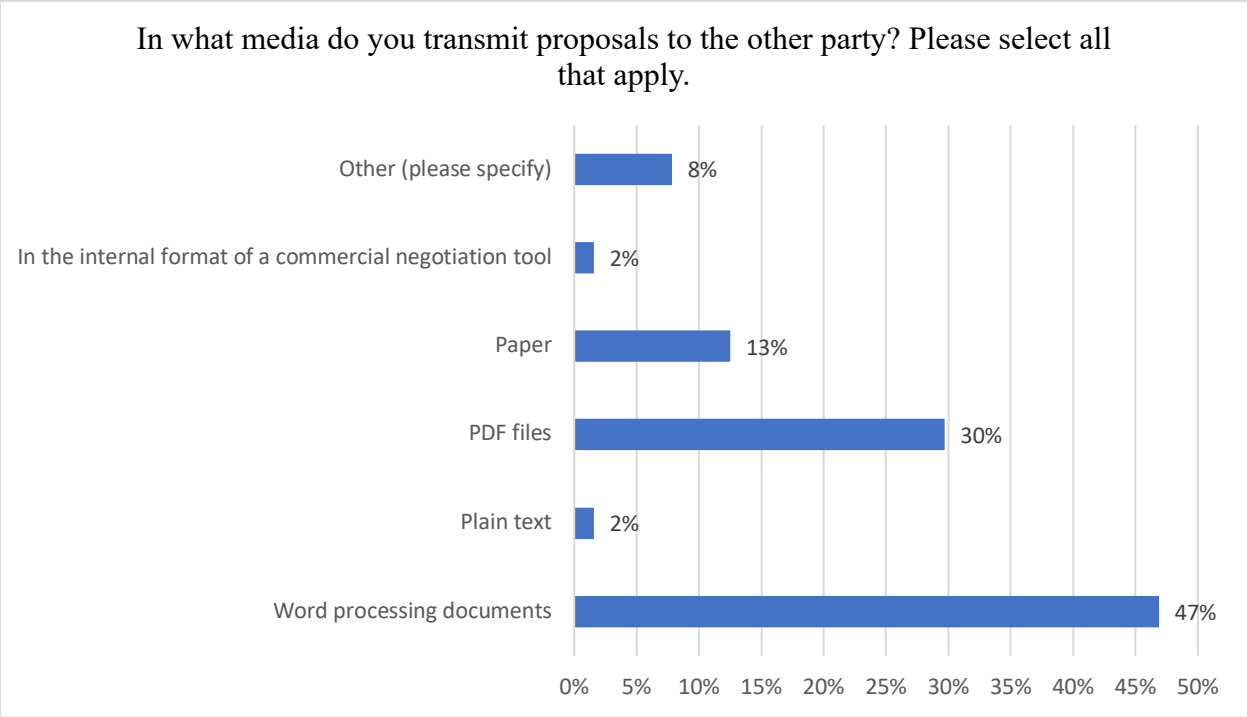

Figure 26: Media use to transmit proposals to the other party

In Question 17, we asked participants in what media they transmit proposals to the other party. 88% said they use Word processing documents, 56% use PDF files and 24% use paper (similarly to the previous question).

It is clear that a combination of word processing and PDF, or either of the two is more used by negotiators in the context of their objectives. For instance, Word processing documents are more used where proposals or counterproposals from the other party are expected, while PDFs are more used where limited input is required such as signing of the proposals, or no input is expected from the other party. Combination of the two can also be used in a situation that the sender wants to keep the version sent, and compare it with the edited version from the other party.

6.7 Experience with Commercial Tools (Survey Questions 20-22)

We looked deeper into the use of commercial tools. This provides information that builds on our discussion in Section 2.3.4 and analyses presented in Table 9. It is clear that contract lifecycle management (CLM) software tools are used by certain *quantitative* negotiators. They are targeted to lawyers, product vendors, and other organizations to manage contracts, negotiations

and approval workflow (Connaughton et al. 2021). Some of these CLM tools support qualitative negotiation. The extent to which these tools can effectively support qualitative negotiation has not been published in the literature because researchers have only been able to interact with a few qualitative negotiators that are aware of CLM tools and have used them; subscription cost may be another hinderance.

Question 20 of our survey therefore asked participants to what extent they have considered or used commercial negotiation tools. We listed ten leading CLM tools as examples. Five of the listed tools were shortlisted from a 2021 Gartner magic quadrant for CLM (Connaughton et al. 2021) and the other five from Google search and LinkedIn. The shortlisted CLM are Concord, Mitratech ContractRoom, DocuSign CLM (Formerly SpringCM), Sirionlabs, Icertis, Juro, Apttus, Avvoka, Conga and ContractPod. At least 77% of participants had never heard or considered using *any* of the ten listed commercial negotiation tools. Only 3% and 9% respectively of participants said they had *tested* using either DocuSign CLM (Formerly SpringCM) or Apttus; but the negotiators did not adopt these tools. Just 3% are currently using Mitratech ContractRoom or DocuSign CLM (Formerly SpringCM) or Icertisand, while 6% are currently using Conga for qualitative negotiations.

For negotiators that have neither considered nor used commercial tools to negotiate, Question 21, an open question, asked for their *reasons* of such avoidance. Thirty-two participants responded. The participants' reasons included no awareness; cost of subscription and training; legal requirements for data protection and classification of information; institutional adoption policy; and a sense that what we are using seems manageable, even though this might prevent them from improving their process.

For the negotiators that *have* used commercial tools, Question 22, asked them an open-ended question, to describe features they find most attractive in the commercial tool they use or have used. Twenty-two participants responded to this question. Sixteen said the question was not applicable to them. The remainder said:

- "almost nothing";
- "tracking agreements, negotiation process and version control, status quo of open items";
- "track previous language";

- "ability to trace back history of changes, maintain a common negotiating text";
- "recall of changes/history of provisions"; and
- "comparing versions of the contract, tagging of variable fields, search within contract, tracking, etc."

Participants' descriptions therefore corroborate the feature set we are proposing as described in Section 6.2.

We posit that more research should be done to know if features in commercial negotiation tools can effectively support qualitative negotiation. Result of such research should be communicated to stakeholders: software developers for extension or improvement of these commercial negotiation tools as may be required; negotiating organization for information and future consideration of such tools. Also, open-source development should be encouraged to reduce limitations posed by the high access cost, versus the perceived benefit, of commercial negotiation tools.

6.8 Conclusions Regarding the Survey

As discussed before, we lean on the supervisor's experience at the negotiation table (discussed in Section 1.3) and a grounded theory study (discussed in Section 3.1) that uncovered several features that negotiators believe should perhaps be available in a tool that would effectively support qualitative negotiation. In our previous research, we also determined that negotiators tend to use only generic word processors which do have useful capabilities such as "track changes", but lack many other needed features.

In this chapter we present result of a survey to characterize how negotiation is performed in various sectors, and to learn how a wider sample of negotiators feel about various proposed new negotiation software features.

The survey showed that negotiators agree with need for many of the proposed features, such as those which would enable negotiators to view, compare and filter information about historical versions of agreement clauses, and also associated notes. The respondents had reservations about features for live editing and signing of agreement changes. The survey also provided a lot of data about how negotiators prefer to collaborate and to mark-up text, which will

also help inform tool design. The survey results generally corroborate the GT results presented in Chapter 5. This corroboration further strengthens that a negotiation tool that implements the features discussed in Chapter 5 will effectively support qualitative negotiation.

Following the positive survey results, developed an open-source prototype that implements some of the features, and in the future will implement all of them. The vision is to develop a complete open-source negotiation system that implements the desired features and is free for the use of non-profit and governmental organizations. In this thesis we present only the design of the metamodel of the tool (in the next chapter), and an early prototype of that tool (in the subsequent chapter).

Chapter 7 Negotiation Metamodel

In this chapter, we present the design of a metamodel for negotiation systems. This model is intended to be used as a baseline for developers who want to create software that implements the negotiation capabilities we have identified in the previous chapters.

We call the artifact a *metamodel*, since it is a higher-level model that can be used to produce specific tool models.

The metamodel design was informed by the research presented in the earlier chapters. The need for various data (classes, attributes, associations) and behaviour (states, transitions) in the metamodel will enable a negotiation tool to properly support all the use cases we discussed in Chapter 5, and particularly the key features also discussed there.

7.1 Evolution of the Negotiation Metamodel

In this section be briefly discuss how the metamodel was developed.

7.1.1 Sources of Input to the Negotiation Metamodel

We developed the metamodel using the following sources of information:

- Informal information from the supervisor's experience at the negotiation table :
- The models used by the existing negotiation support systems:
- The requirements, expressed as use cases and features developed from our grounded theory and our survey:

7.1.2 Preliminary Version

During the early stage of our research, we design a version of negotiation metamodel out of analysed informal information provided by the supervisor during his experience at the negotiation table. This version, capture many of the details of the final metamodel, but needed a lot of fine-tuning.

7.2 Final Negotiation Metamodel

In this section we present the metamodel and describe each of the classes in detail.

The negotiation metamodel created via the process described above can be used for the development of negotiation tools or negotiation domain-specific languages. It is our hope that multiple vendors and open-source developers might adopt it, to allow negotiation data to be exchanged. This is particularly important since is likely that the parties might prefer to use different negotiation tools, so data would need to be transmitted between them using a common vocabulary and common data structures.

In the following, words that are initially capitalized and in ***bold italics*** are classes in the metamodel. We have pluralized these names sometimes in the text to aid readability, but in the metamodel they are singular, as is considered best practice in class diagrams. Words that are *in italics but are not initially capitalized* refer to attributes or associations.

The metamodel is written in Umple, and the code is presented in Appendix 3. The Umple code is also online at the following URL:

https://github.com/umple/agreementNegotiation/tree/main/negotiationMetamodel/src.

The diagrams in this chapter have all been generated by Umple; this means that if any changes are required, the diagrams can easily be regenerated.

The full metamodel showing associations, but with attributes hidden, is shown in Figure 27. To aid understanding, we explain the metamodel using sub-diagrams in Figure 28 through Figure 34, in the following subsections.

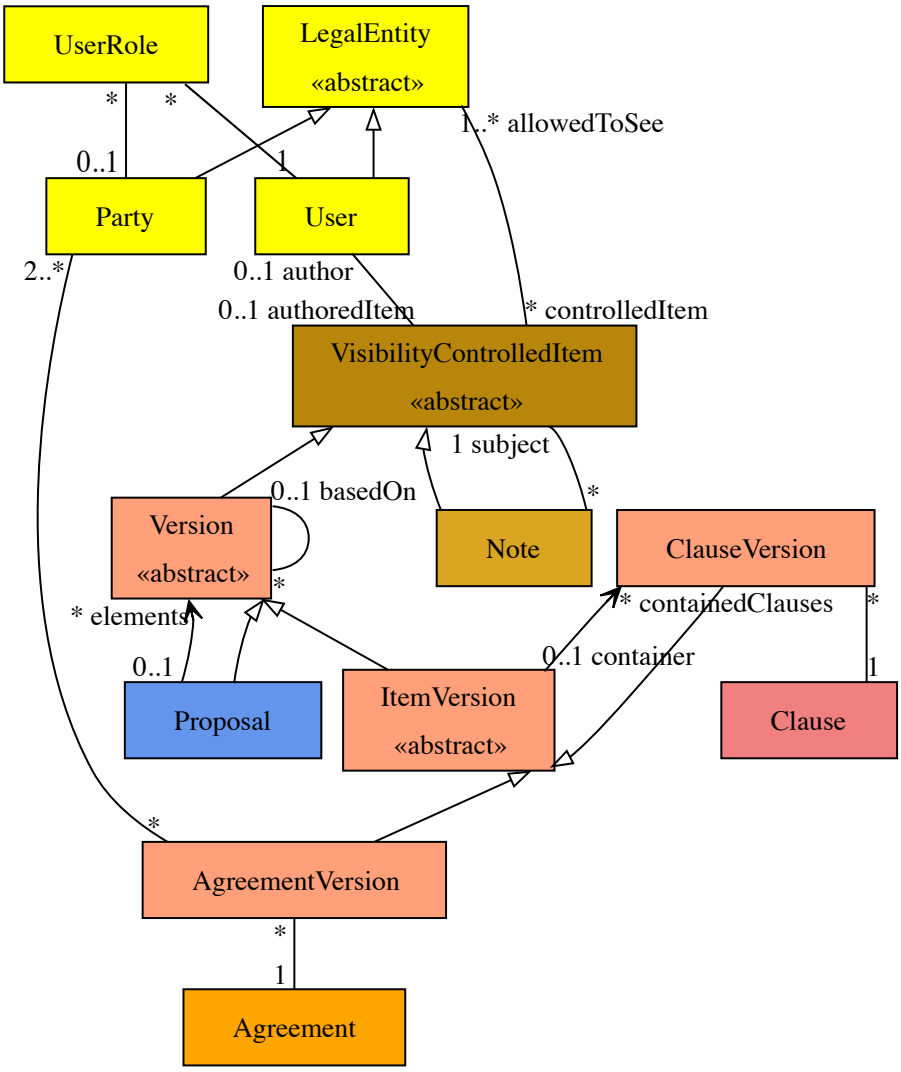

Figure 27: Class diagram showing an overview of the qualitative negotiation metamodel, with attributes suppressed

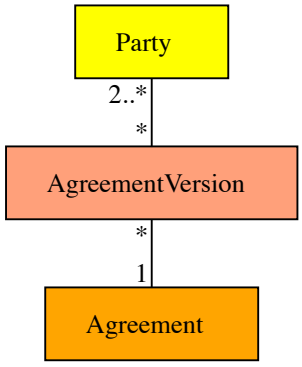

Figure 28: Class diagram showing Agreement and its associated classes, with attributes suppressed

7.2.1 The Agreement and AgreementVersion Classes

All negotiations have as their objective, the creation or update of an ***Agreement***, shown (in orange in Figure 28 and subsequent figures. Figure 28 also shows the two classes most closely associated with it, ***AgreementVersion*** and ***Party***.

There are many relevant types of agreement, including international treaties, commercial purchase agreements, labor-management collective agreements, and family law agreements. All of these can be represented by the ***Agreement*** class and are deeply important for the functioning of society. Yet, as we have shown earlier in this thesis, technology for negotiation of such agreements is currently severely underdeveloped.

Figure 28 shows that an ***Agreement*** involves a minimum of two instances of class ***Party***, shown in yellow (here and in later figures), but could involve more parties, such as in the case of a multilateral trade agreement. We will discuss the ***Party*** class later.

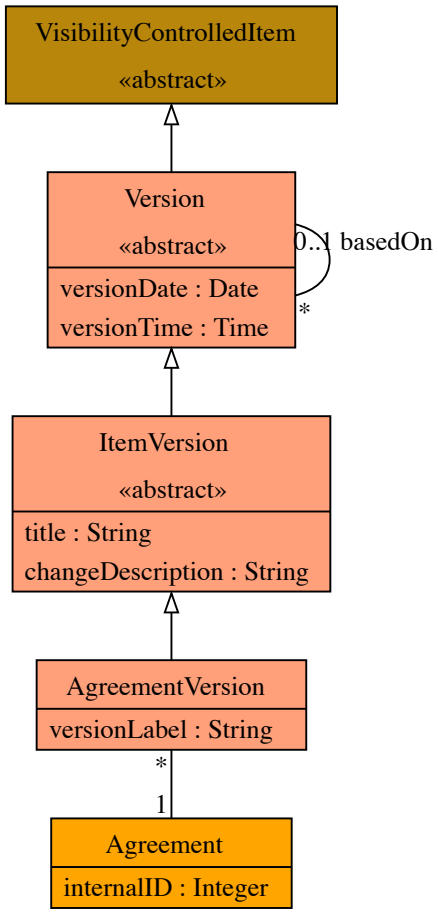

Figure 29: Class diagram Showing attributes of *Agreement* and *AgreementVersion*, including those inherited from superclasses.

An ***Agreement*** will have a series of **AgreementVersions**, shown here and later in Pink. Figure 29 shows more context regarding class **AgreementVersion**, including its superclasses and their attributes.

It can be seen from Figure 28 and Figure 29 that most of the core information identifying *Agreements* is actually found in instances of ***AgreementVersion***. Why? Because as an ***Agreement*** evolves, essentially everything in it can change.

Firstly, the ***Parties*** may change from version to version. Although for some kinds of agreements this might not be normal, for others it is entirely standard. For example, new countries might join trade agreements, thus the list of ***Parties*** may grow in later ***AgreementVersions***. Similarly, ***Parties*** may withdraw, such as the USA withdrawing from the Paris Agreement in 2020. This is why the association in Figure 28 associates ***Party*** to ***AgreementVersion***.

The other attributes essential to each ***AgreementVersion***, as shown in Figure 29, include the following:

- From class ***Version***, *versionDate* and *versionTime*: These are hopefully self-explanatory. Note that other entities in the metamodel can also be versioned; we will discuss these other uses of the *Version* class later.
- From class ***ItemVersion***: *title*: This is what the *Agreement* is known as. It can change from version to version. Note that ***Clauses*** within an agreement are also versioned using the ***ItemVersion*** class, so we will have more to say about this class later.
- From class ***ItemVersion***: *changeDescription*: This is textual information regarding what is new or different in this version.
- Attribute *versionLabel* in **AgreementVersion** can be used to expand on the *title*, perhaps to help form a filename for the ***AgreementVersion***.

Several associations are also inherited by ***AgreementVersion***. Such as to ***ClauseVersions***, and ***Users*** who may have access to the version. We will review these in later subsections.

Class ***Agreement*** itself has only one attribute: *agreementID*. This would be system-generated so as to be unique. It is needed simply to allow a large system to handle many ***Agreements*** independently. Users would only ever need to see the *agreementID* when doing tasks

such as listing agreements (where the titles might be similar) or duplicating an agreement, such as to construct a similar agreement between different parties using a template ***Agreement,*** or performing ComparePrevious discussed in Section 7.4.2.

When a new negotiation is started, some ***Parties*** must first be created. An ***Agreement*** can be created that initially has no ***AgreementVersions***. However, it will not be of much use until the first ***AgreementVersion*** is created, which will require specifying at least two parties, as per Figure 28.

As can be seen in Figure 29, the ***Version*** class has a reflexive *basedOn* association indicating the version of each instance is a modification of (i.e. its parent version). This association forms a hierarchy, since there can be many alternative proposed updated ***Versions*** (of ***AgreementVersions***, but also ***ClauseVersions***, ***Proposals***, etc. as we will see later). These alternatives may be private ***Versions*** being 'played around with' by ***Users***, or ***Parties***, or being prepared as alternatives in case one ***Version*** is rejected. Or they may be ***Versions*** prepared by a mediator.

One of the leaves of this hierarchy will hopefully be the final approved ***AgreementVersion*** (tagged as such using the *changeDescription* attribute). But even that may have descendants as the agreement is opened up, perhaps a few years later, for another round of bargaining after the initial agreement expires.

7.2.2 Legal Entities and What is Visible to Them

Figure 30 shows the ***LegalEntity*** class and its two subclasses, along with the ***UserRole*** class.

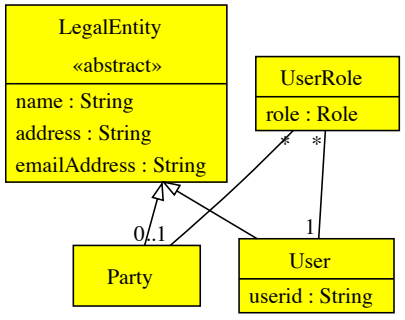

Figure 30: Class diagram of the *LegalEntity* subhierarchy and the *UserRole* class

The ***LegalEntities*** involved in negotiation are the ***Parties*** plus the ***Users***. These classes are shown in yellow in the class diagrams in this chapter.

The abstract ***LegalEntity*** class captures basic identification information, such as the *name*, *address* and *emailAddress*, that is common to the ***Party*** and ***User*** classes. Other similar information could be added to expand the metamodel.

A ***Party*** is the entity whose interests are being negotiated; there must be at least two in any negotiation (as indicated by the association to Agreement in Figure 28); they could be sovereign states, corporations, labor unions or individuals.

A ***Party*** needs ***UserRoles***, with *roles* defined in an enumeration:

- *ChiefNegotiator*: Authorized to 'sign off' with a *chiefNegotiator* of another ***Party*** that a certain ***ProposalVersion*** has been agreed to.
- *MemberAtTable*: Someone working in the team of a ***Party*** who can see ***VisibilityControlledItems*** shared within that Party, but who cannot sign off on wording.
- *Mediator*: Someone who can be given access to certain ***VisibilityControlledItems***, such as an ***AgreementVersion***, but is not specific to a certain ***Party***. Their role is to help multiple ***Parties*** come to an agreement.
- *Arbitrator*: Someone who has been given authorization by the *ChiefNegotiators* of all ***Parties*** to make a decision, typically between two ***Proposals***. When the Arbitrator makes a decision, one of the ***Proposals*** becomes accepted.

Each ***UserRole*** associates zero (in the case of an Arbitrator or Mediator) or one ***Party*** to one ***User***. Large multi-national agreements such as for climate change may have hundreds of ***Parties*** and thousands of ***UserRoles*** involved.

The reason for tracking ***UserRoles***, rather than just having ***Parties*** in the metamodel, is so that we know who is making proposals, or signing off (formally approving final wording) on parts of the agreements, and so that there can be different roles for the people involved with each party.

At the time a negotiation system based on this metamodel is initiated, it will be necessary to initialize the system with at least two ***Users***, who can then be associated with ***Parties*** via the creation of ***UserRoles***. A Party is of no use without at least one ***UserRole***. However, a User can have different ***UserRoles*** associated to different ***Parties*** associated with different

AgreementVersions. For example, a manager might negotiate with multiple unions in their company.

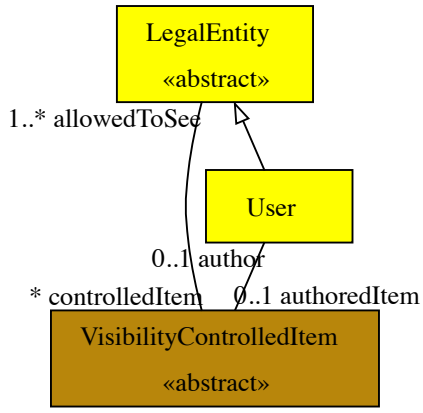

Figure 31: Class diagram showing control of visibility to specific *LegalEntities* and acknowledgement of authorship

As the leftmost association in Figure 31 illustrates, a ***VisibilityControlledItem***. This is something that only one or more ***LegalEntities*** are allowed to see (i.e. a specific set of ***Users*** or all ***Users*** associated with a ***Party***).

VisibilityControlledItems include ***Versions*** (as per Figure 29) of ***Clauses*** or ***Proposals*** that may be incomplete or being prepared by one ***Party***. It may be that only one user, such as the user creating a draft, can currently see a ***VisibilityControlledItem***; or maybe everybody in a ***Party*** can see it, or maybe all ***Parties***, or maybe just users with certain roles such as the chief negotiator of a party and an assistant.

A ***VisibilityControlledItem*** may also have an identified ***author***, as shown in the rightmost association in Figure 31. We have allowed for the possibility of an ***VisibilityControlledItem*** having no author, perhaps to enable initialization with template text. We have not allowed a ***VisibilityControlledItem*** to have more than one author, since logically some system user would be the creator of each item. However, as we will see ***VisibilityControlledItems*** such as ***Clauses*** are composite, and the various parts may internally have separate authors. Authorship of each change would be tracked through the fact that every ***Version*** is a separate ***VisibilityControlledItems***.

7.2.3 Notes

We saw in Figure 29 that ***Versions*** (hence ***AgreementVersions***) are a kind of ***VisibilityControlledItem***. And in Figure 31, we saw that ***VisibilityControlledItems*** have authors and users who can see them.

In addition to ***Versions***, a second type of ***VisibilityControlledItem***, shown in Figure 32, are ***Notes***. Notes can be recorded about any ***VisibilityControlledItems***, including about ***Notes*** themselves (e.g. when someone replies to a note or wants to provide more detail). This is enabled because ***Note*** is a subclass of ***VisibilityControlledItem*** and also associated with it via the *subject* association (this is called the Composite or General Hierarchy pattern).

During our grounded theory study, the tremendous importance of careful note recording, as well as electronic access to those notes became apparent. In our preliminary version of the metamodel, we had a class called ExplanatoryText, which had two subclasses Rationale and Feedback. However, in our refined model we decided that a simpler design would be better: Our new design has class ***Note*** and uses an attribute to optionally distinguish various kinds of notes.

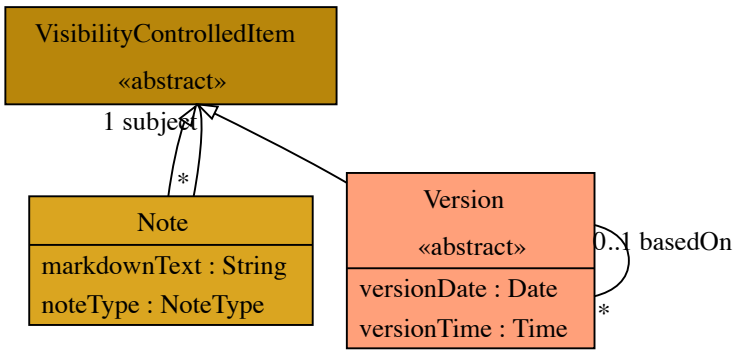

Figure 32: Class diagram of class *Notes* as applied to *VisibilityControlledItems* including other *Notes*

A ***Note*** can be about anything. Some possibilities include:

- Comments about a particular ***Clause*** (a subclass of ***Version***, see next section); perhaps noting a problem with the ***Clause*** before a ***Proposal*** is created.
- Rationale for a ***Proposal*** (another subclass of ***Version***) being prepared by a ***Party***. This might be visible initially only to the ***User*** who is the author of the ***Proposal***, and later to all ***Users*** in the ***Party***.

- Rationale to be revealed to the other ***Party*** for a ***Proposal***, which would be different from the rationale shared internally within the ***Party*** (last bullet). It might be that this rationale-to-reveal would actually be shared in the system, but perhaps it might only be shared orally.
- Rationale that members of ***Party*** B write when receiving a ***Proposal*** from ***Party*** A, recording why it thinks ***Party*** A is making that Proposal (i.e. ***Party*** B's attempt to express what ***Party*** A has told them orally).
- Comments that ***Users*** in ***Party*** B make about a ***Proposal*** from ***Party*** A, but which they don't want to reveal to ***Party*** A (i.e. what Party B really thinks of the proposal, such as concerns with it).
- Private observation commenting on some other ***Notes*** (hence visible to only that ***User***).

In most modern negotiations we studied, notes are currently recorded in rather random ways (often on paper printouts of agreement versions stored in binders), highlighting the need for computer support for notes in the negotiation process. Each member of a negotiating team tends to have their own notes, which may well be incomplete. The chief negotiator or his assistant may try to keep comprehensive notes but, lacking any way of searching for these or sharing them with others, the value of the notes is very much diminished. Negotiators can therefore forget about important points and waste time re-discussing topics.

Notes have types in our metamodel using an enumeration. Types of ***Note*** include the following:

- *Unknown*: The default when a ***Note*** is created. Negotiators may choose not to change this tag to any of the following, but the following can be used to help keep track of the status of notes, if the negotiators want to do so.
- *Rationale_Private*: Reason for the noted ***Clause*** wording or ***Proposal***
- *Rationale_ToReveal*: Reasoning that we plan to reveal to the other ***Party*** but have not done so yet.
- *Rationale_Revealed*: Explanations we told the other ***Party***. It can be particularly useful to use this type of ***Note*** to keep track of what was explained. Whether or not

these ***Notes*** are visible to the other party or whether the revealing was just done orally is a up to the negotiators.

- *Rationale_FromOtherParty*: Explanations revealed by the other ***Party*** about their proposals or about why they like or dislike certain clauses they have authored, as recorded by ***Users*** of our ***Party***.
- *Analysis_AboutOtherParty*-Private: Things we think of the other party's proposals or clauses they have authored, but which we are not going to reveal to them.
- *Feedback_ToReveal*: Things we are planning to explain to the other party about why we like or do not like their proposals etc.
- *Feedback_Revealed*: Things we have revealed to the other party about what we like or dislike about their proposals.
- *Feedback_FromOtherParty*: Explanations revealed by the other ***Party*** about why they do or do not like our proposals, as recorded.
- *ProcessObservations*: Thoughts about the process, not tightly tied to a ***Version***; maybe just associated with the most recent ***AgreementVersion***.
- *Data:* Information that can help inform discussions, but which may or may not be tied to a ***Proposal***. For example, when preparing to negotiate salaries, data about salaries of other employee classifications might be recorded.
- *Other*: Something else not covered above.

Note recording can be used to help ensure all relevant ***Users*** understand the negotiation context during live negotiations, but they can also be used later to help remember or interpret earlier decisions.

Sometimes old ***Notes*** can be used to help resolve disputes about interpretation of intent and what led to wording of a particular ***Clause***. In our grounded theory study, we learned that ***Notes*** can sometimes be examined years later by arbitrators, so arbitration might be a lot simpler if there was online access to them.

Clearly it is of utmost importance to control the visibility of ***Notes***; some have to be totally private to one ***User***, and many would be private to a ***Party***.

7.2.4 Clauses and ClauseVersions

Clauses are the core elements of an agreement. Each time a ***Clause*** is created or changed (even in a ***Proposal***) a new ***ClauseVersion*** (see Figure 33) is created. The purpose of the ***Clause*** class is merely to tie together the multiple versions and allow the history of a clause to be explored, and comparisons made, even if a clause is radically changed over time. This is similar to the modeling construct we employed with ***Agreement*** and ***AgreementVersion*** discussed earlier. A Clause has an *internalID* that would normally never need to be seen by Users.

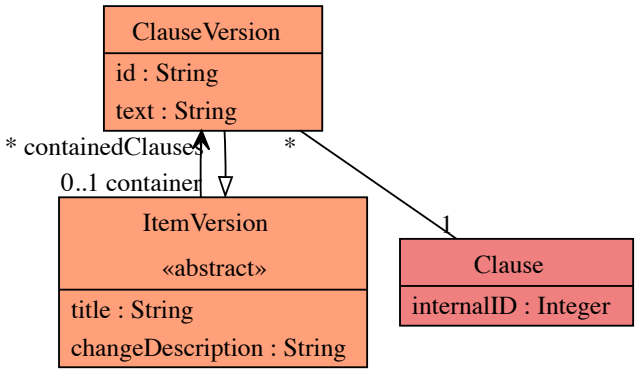

Figure 33: Class Diagram of *Clause* and *ClauseVersion*

Clauses usually also have a human-meaningful *id* that is visible to users, such as 'Article 3' or '3.2.4'; this is maintained using the id attribute in ***ClauseVersion***. This would normally be a number, letter, perhaps preceded by a word such as 'Section. The *id* may be changed from version to version: it is common to re-number clauses when inserting a new clause earlier in the numbering sequence. The human-readable id is, however, optional since we have observed agreements where several low-level sub-clauses are not numbered or lettered.

Clauses are normally nested (see the *container* to *containedClauses* association). One would normally refer to a deeply nested subclause using a scheme such as 3.4.5, composed of the ids of the successively contained ***Clauses*** (i.e. the top-level clause might be Article 3, and within that there might be 3.1, 3.2, 3.3 and 3.4, and within 3.4 there might be several lower-level sub-clauses including 3.4.5. Since ***AgreementVersion*** is also a kind of ***ItemVersion***, the ***AgreementVersion*** contains the top-level ***Clauses*** in an agreement (which may have IDs such as 'Article 1', 'Article 2' etc.).

The text of a clause Is also specific to each ***ClauseVersion***. The text is the essential legal text of the clause.

We suggest that in a tool, the *text* be stored in Markdown format to allow for italics, bullets and a few other simple formatting details that are part of the agreement itself, and not visual markup that would be separately used by the tool to show additions, deletions etc. An alternative would be a heavily constrained subset of ECMA-376, the XML format used in word processors such as Microsoft Word. That would allow easier construction of Word documents and their reverse-engineering. The supervisor's negotiation experience, grounded theory and survey all show that almost all negotiations in international trade, and labour-management domains involve editing Word documents, but that only a small subset of the formatting features of Word are used.

Actual internal storage of ***Clause*** *text* might also use 'diffs' or deltas as is done in git. Whether to do this could be left for detailed design (an implementation of this metamodel could perhaps even be layered on top of git). Our earlier version of the metamodel did have the class Delta, to represent the storage of diffs, but we have decided this should instead be an internal design detail.

ClauseVersion is a subclass of ***ItemVersion***, which is in turn a subclass of ***Version***. As we discussed earlier, ***AgreementVersion*** is also a subclass of ***ItemVersion***; and as we will see in the next section, ***Proposal*** is a subclass of ***Version*** too. The hierarchy ensures ***ClauseVersions*** inherit the attribute *author* (of the change) and *visibility* from ***VisibilityControlledItem***, as well as *title* and other attributes.

A change to any ***ClauseVersion*** contained by an ***ItemVersion*** (higher level ***ClauseVersion*** or ***AgreementVersion***) results in a new instance being created, even if it is a temporary edit; however, any *containedClauses* that are not modified do not have to be duplicated; this is enabled by making the *containedClauses* a directional association.

7.2.5 Proposals

We have discussed most of the metamodel. The only class that remains to be discussed in detail is ***Proposal*** (blue in Figure 34).

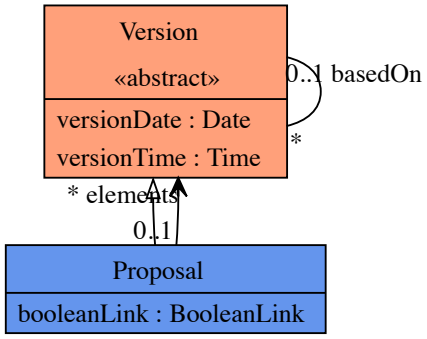

Figure 34: Class diagram focusing on *Proposal*

A ***Proposal*** is a kind of ***Version*** since it can be updated (and has controlled visibility, authorship etc.). It also contains set of ***Versions*** (its *elements*). The *elements* can be any kind of ***Version*** such as ***ClauseVersions*** (i.e. proposals for textual change), or other ***Proposals*** (i.e. subproposals). Because ***Proposals*** can be nested in this way, it allows ***Proposals*** to have Boolean logic connecting the elements, captured in the *booleanLink* attribute. This supports use case UC-Cr-4-Condit, as discussed in Section 5.1.1.

Most sets of ***Proposal*** elements would be linked by 'and' meaning a conjunct. A conjunct means that all changes are requested to be made together ('we need changes to both ***ClauseVersion*** A and B to be made in order for the new ***AgreementVersion*** to remain consistent'). But some elements could be alternatives with 'xor' such that 'either this or that version of the wording would be acceptable'. Others could be 'or' disjuncts ('we are proposing some changes, but they don't need to be all made').

A final approved ***proposal*** has to be a conjunct, or just have a single element.

This need to have logic in proposals was informed by the supervisor's experience at the negotiation table. We noticed that negotiators often get bogged down in the complexities of alternative proposals. A tool that supports Boolean logic with a good user experience (e.g. where versions of an agreement section (i.e. a high level clause) could be rendered with different possible solutions to the Boolean expression) might help resolve impasses.

Proposals could suggest deletions of subclauses by simply proposing the parent clause with the to-be-deleted subclause removed from the *containedClauses*. Additions of clauses could be proposed in a similar manner.

A special type of ***Proposal*** is a counterproposal. This is a ***Proposal*** that is based on another ***Proposal*** but has a changed set of elements (e.g. one ***ClauseVersion*** replaced by a different one. The *basedOn* association enables this.

Since a ***Proposal*** is a ***VisibilityControlledItem***, it can have ***Notes*** containing rationale and feedback, just like ***ClauseVersions***.

We created a state machine model for a ***Proposal,*** presented in Figure 35. Note that the transitions in Umple state machines become methods in generated code.

A ***Proposal*** is *initiated* by its author, who may then be *internallyPresent* it to a ***Party***'s other ***Users***. They may then *returnToAuthorForRefinement* or *present* the ***Proposal*** to the other ***Parties*** at a live or virtual negotiating table.

During negotiation, any of the other ***Parties*** may reject the ***Proposal***, sending it back to be *underInternalConsideration* by the originating ***Party***. Alternatively, any of the recipient ***Parties*** may *takeToCaucus*, meaning the ***Proposal*** is no longer to be discussed openly until it is returned back by either a *reject* or *presentCounterProposal* operation. Either of these may not mean the end of the ***Proposal's*** life as it can sit in abeyance *underInternalConsideration*, perhaps to be presented again later.

A *reject* transition (by another ***Party***) may occur either during live negotiation (*underNegotiation*), or while privately *underOtherPartyConsideration*, in which case the originating ***Party*** has it back *underInternalConsideration*. It could have further *edits* in that state, or even go back to its author for detailed private *edit* work.

A ***Proposal*** could have a *withdraw* operation occur when *underInternalConsideration* or even when merely *initiated*.

If a proposal is agreed by the other ***Parties*** (*agreedAsDraft*) then that is the end of its life as a new ***ItemVersion*** now is active, that has *incorporated* the contents of the ***Proposal***.

On the other hand, the originators may formally *withdraw* the proposal, making it obsolete. Or the originators may decide to consider a counter-**Proposal** (*considerCounterProposal*), i.e. a new ***Proposal*** presented by another ***Party***, agreeing that their original ***Proposal*** is no longer live, and has been *superseded*.

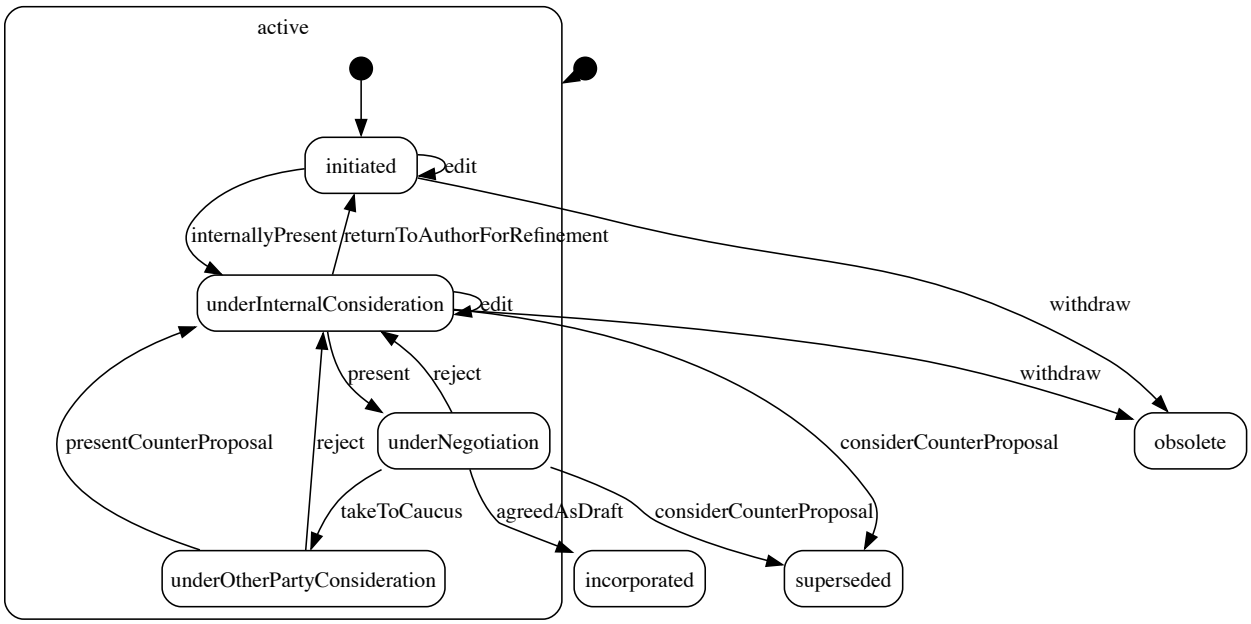

Figure 35: State machine of class *Proposal*

7.2.6 Recap: ItemVersion as parent of AgreementVersion and ClauseVersion

As discussed earlier, the result of negotiations is an ***AgreementVersion***, shown in Figure 28. An ***Agreement*** gets its current title from the latest ***AgreementVersion***, which inherits this from class ***ItemVersion*** class (the title can hence change) and the top-level ***Clauses*** from the *containedClauses* of that ***AgreementVersion***. The current wording of any ***Clause*** can also be viewed by looking at the latest ***ClauseVersion*** (since ***ClauseVersion*** inherits from ***ItemVersion*** as well, as shown in Figure 33).

Agreements are updated over time, so the latest approved ***AgreementVersion*** is normally the most important, although as our research shows, the ability to look back at earlier versions (including the ***Notes*** on them and the ***ClauseVersions*** in them) is also important.

We created a state machine for the ***ItemVersion*** class (and hence ***AgreementVersion*** and ***ClauseVersion***) shown in Figure 36: An ***ItemVersion*** can be *initiated*, *underNegotiation* (there are outstanding ***Proposals***), *underVerification* (the ***Parties*** tentatively agree) but also agree that they may be able to propose additional error correction ***Proposals***), *tentativelyAgreed* (the Parties are obtaining consent from others in their organization such as a president, members or a board of directors), *finalized* (indicating it is in legal force), or *noLongerConsidered* (perhaps one of a potentially large number of versions that was considered but not accepted in the search for an

acceptable compromise). An ***ItemVersion*** can go from *tentativelyAgreed* back to *underNegotiation* if the stakeholders that are required to formally approve in a ***Party*** vote against it.

As discussed earlier, Figure 27 shows an overview of the final version of our negotiation metamodel. This metamodel can be opened directly in UmpleOnline by going to the following URL.

http://cruise.umple.org/umpleonline/?example=Negotiations&diagramtype=GvClass

The Umple code for the metamodel is open source in Github, at https://github.com/umple/agreementNegotiation/tree/main/negotiationMetamodel/src, so pull requests (with evidence from further human studies hopefully) are welcome.

Versions in XMI and several other languages (Java, Python, PHP etc.) can be generated from Umple allowing broad use of the metamodel in other modeling tools as well as generation of actual software.

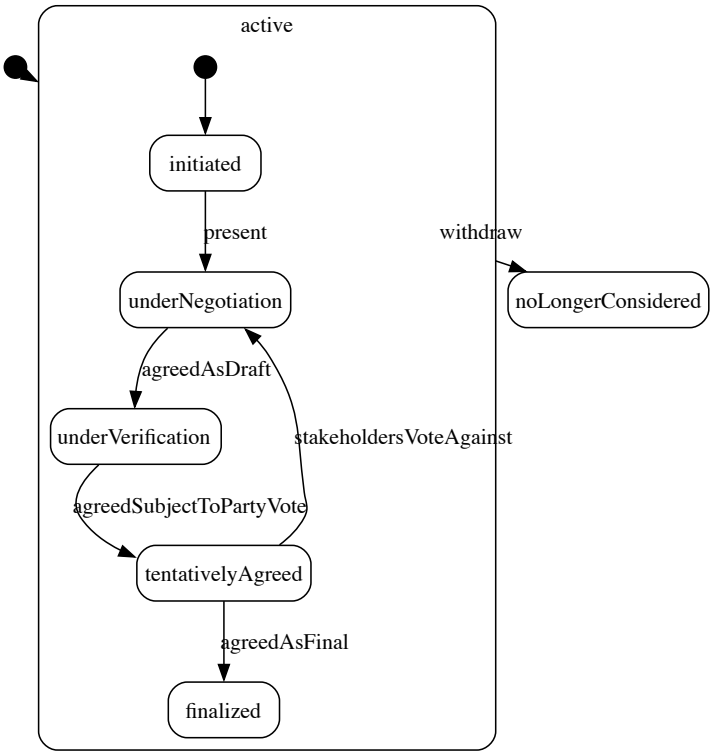

Figure 36: State machine of class *ItemVersion*

7.3 Discussion of the Process of Producing this Metamodel

A key aim of our work is to promote the idea of development of a wider selection of publicly available systems for the important human-centric domain of negotiation.

As we discussed in Section 7.1, we used multiple methods to arrive at the model: The supervisor's experience at the negotiation table gave us information about the structure of clauses and subclauses, and tracking changes over time. The grounded theory and survey provided more detail about many of the same issues, particularly finding ways to enable advanced capabilities, while not taking users too far away from their Word files. In the GT we also learned more about the challenges faced taking notes.

A traditional approach to create a standard metamodel would be for a committee to work on the model; the committee would then seek public input. However, in such committees there is often a lead author whose experience dominates. Even after multiple rounds of public review, that model tends to reflect the view of a few. In our work we combined a variety of methods, obtaining input from people who are both skilled in the domain and applying modeling expertise to ensure their needs are met. Input from developing the use cases, the SoarNego tool (next chapter) also helped fine tune it. We iteratively evolved the metamodel, validating it by generating code and diagrams and working out how it could be used to implement key features (discussed in the next section) We suggest that this multi-method, multi-input approach should be considered a good practice for others to follow.

Naturally the metamodel we have presented is only a work in progress, because (as pointed out above), we will need to involve more experts, and iterate more. We will also need to build prototypes to find further improvements. Nonetheless, we hope that the model we have presented here, and made public on Github, will be used by others as the basis for negotiation systems that can perhaps help countries work better together, or reduce labour strife.

Models built in this manner will serve as one-stop source of information and foster handshake of technologies. We posit that such consensus models must factor in socio-technical issues in their design, be publicly accessible, and open to extension and modification by both the research community and by industry.

7.4 Example of How the Metamodel can Be Used to Implement Some of the Core Features

Here we explain how a software tool can navigate the metamodel to implement some of the key features we discussed in Section 5.4. We are assuming code generation from Umple, but a different code generator could also be employed.

7.4.1 SeePrevious Implementation with the Metamodel

As a reminder, this feature enables the user to instantly see wording of selected earlier versions of a particular clause, including of higher-level clauses and their descendants in the hierarchy of an agreement.

Input: An instance of ***Version*** would be initially on display and selected in the user interface. This could be a ***Proposal***, a ***ClauseVersion*** or an entire ***AgreementVersion***. All of these are versioned and have an *internalItemID* (inherited from ***VisibillityControlledItem***) that can be passed to a function to perform the query for the SeePrevious feature.

An extended starting point would be if more than one ***ClauseVersion*** was already selected. This could be useful if the user wants to examine several very similar clauses.

Processing: A call would be made to *getBasedOn()*, an API function generated from the *basedOn* association in the ***Version*** class. This would return the previous ***Version*** if any. Calls to *getBasedOn()* would be made iteratively to obtain even earlier ***Versions***, generating a list. The list would then be presented in the user interface. If more than one *ClauseVersion* was initially selected, the processing would be repeated for all of them, generating a longer list, that might be sorted by time of creation.

Output: The list on the screen would non-modal and would show the content of the selected ***Version***(s), as well as the dates when they were changed, and a little other context. For a single clause this could be highlighting of what is added and what is deleted, perhaps with an indication of which party. For a larger block of material (perhaps more than 30 words), it would summarize the number of changes and perhaps highlight just the first one. Note that unlike ComparePrevious, this feature focuses on showing the full list of earlier ***Versions***, but not the full content of the versions or a detailed "diff" operation.

Next Steps: The user would be able to select a ***Version*** to view it in full, and optionally to use ComparePrevious to compare a ***Version*** in the list with the current one.

7.4.2 ComparePrevious implementation with the Metamodel

This feature enables comparing two ***Versions*** *in full*.

Input: Two instances of ***Version*** would be selected in the user interface (perhaps using SeePrevious as above), one of which might simply be the latest ***Version*** (or whatever was on display before SeePrevious was invoked). The class ***Version*** has an *internalItemID* (inherited from ***VisibillityControlledItem***) that can be passed to a function to perform the query to get the full text for the ComparePrevious feature.

Processing: A "diff" operation would be performed between the two selected ***Versions*** with the results displayed. This would work in a way similar to tools like Github.

Output: The two versions would be displayed side by side showing the differences. Highlighting would indicate addition and subtraction of text by either party. Hovering over highlighted text would display a tooltip indicating the party that added or deleted a block of text.

7.4.3 PrivateNotes Implementation with the Metamodel

Input: A ***VisibilityControlledItem*** (a ***Version*** such as a ***ClauseVersion*** or an existing ***Note***) would be displayed in the user interface and be selected by the user. A note creation button would appear in a menu or ribbon bar in the user interface. The user would click on this command.

Processing: An instance of ***Note*** would be created and a call would be made to method *setAllowedToSee()* of the class ***VisibilityControlledItem*** to specify by default that the ***LegalEntity*** representing the current ***Party*** is the only one able to see the note. Options would be available to change the ***LegalEntity*** to be just this ***User*** or multiple specific members within the same ***Party***.

Output: The ***Note*** would appear and would be editable. Check buttons would be displayed from which user can check ***VisibilityControlItems*** for the note created. For instance, intra or inter ***Party*** member could be checked so they would be *allowedToSee*.

Next Steps: Note access visibility would require the *ChiefNegotiator's* (*role* within ***UserRole***) approval before being made *allowedToSee* to the other ***Party***.

7.5 Comparison of the Metamodel with Other Models

In this section we discuss in detail how our metamodel differs from other models presented in the literature.

The commonality between our metamodel, Negoisst model discussed in Section 2.2.3.1, SPICA model discussed in Section 2.2.3.2 and the work of (Chiu et al., 2005) discussed in 2.3.3 is that a textual document is the base of negotiation (which is part of the answers to our research question RQ1 discussed in Section 9.2a). Such document is either generated from a template, initiated from an already-prepared document, or directly typed on the tool interface. All the models attempt to break the documents into units called *message* by Negoisst discussed in Section 2.2.3.1; as well as *clauses* by SPICA discussed in Section 2.2.3.2, (Chiu et al., 2005) and our metamodel. The common goal of these models is to enable negotiating parties to input or edit clauses to represent their compromise or agreement. Another similarity is *party* in our model and the work of (Chiu et al., 2005), the same is represented as *partner* in Negoisst and SPICA.

The major difference between our model and the existing ones is the ability to introduce a new clause where need be or edit already existing clauses. Flexibility to edit existing clauses is what other existing models lack, rather they introduce a new clause to supersede the existing ones. As a result, negotiators need to explicitly tag every clause or message to convey what it represents (for instance as, an offer, a request, a reject, etc.) We posit that such tagging is not feasible nor desirable in a densely textual qualitative negotiation that requires a large amount of back-and-forth editing of contractual text.

As discussed in Section 2.2.3.1, We have adapted some of the points used to analyse Negoisst to further discuss how our metamodel differs from the existing models.

1. **Define message type**: Our model supports all message types in the existing models, as well as "withdraw" that many of the existing models do not support. However, message type is implied by the activities performed by the negotiators (editing text, creating a proposal etc.) in our model and not explicitly tagged as further explained in point "4" below.
2. **Ability to deal with complex tex**t: As discussed in Section 7.2.1, 7.2.4, 7.2.5, and 7.2.6, our model has a mechanism to break an agreement into internal structures that keeps track

of clauses and sub-clauses, at several levels of hierarchy, from one agreement version to the other. This is a key mechanism that all the existing models lack.

3. **Version View Control**: This is one of the main advantages of our model over the existing ones. Figure 29 and Figure 32 as discussed in Sections 7.2.1 and 7.2.3 respectively detailed how ***VisibilityControlledItem*** controls both ***Note*** and ***AgreementVersion***.
4. **Message Tagging**: There is no constraints of message tagging in our model.
5. **Party Multiplicity**: There is no restriction to the number of parties that can negotiate. Figure 28, a subset of Figure 27, has 2-to-many relationships between ***Party*** and ***AgreementVersion***.
6. **Notes**: Taking notes and cross-referencing notes is another important technique supported by our model, which was not considered by existing models.

Also, in our metamodel we did not consider a particular issue in the work of (Chiu et al., 2005): Template clauses with variables. This could be added to extend our metamodel, but the need for it did not arise in any of our empirical research.

7.6 Conclusions

In this chapter, we have discussed how we combined data from multiple sources to design a metamodel suitable for managing the data of qualitative negotiation systems. We had designed an initial version of the metamodel based on participation and experience gathered during negotiation by the supervisor. Then we evolved the initial version of the metamodel with data gathered from studying the existing negotiation models in the literature, our grounded theory research, and our survey.

The final version of the metamodel presented in Figure 27, with details in subsequent figures, has colors to enhance studying the model. Related classes have common color and we used sub-diagrams to buttress our explanation of the core classes in sub-sections of this chapter. The metamodel is represented in Umple, so it can be used to generate code in a variety of languages; it can also be exported to other modeling technologies.

Our vision is that multiple negotiating tools could use this metamodel to manage data. Each party would use their own negotiating tool for security reasons, i.e., so there is no risk of seeing

the data of the other parties. Data would be transmitted to keep the data of all parties in sync, except for instances of ***VisibilityControlledItem*** that should remain private to their creating party. A proposal would therefore be transmitted to the other party when the ***LegalEntities*** *allowedToSee* it change.

We compared our metamodel with the existing models and explained how our model differs from them. We also discussed how our metamodel can be used to implement some of the core negotiation features we discussed in Chapter 5.

While we posit that our metamodel can be used as a one-stop source of information for researchers, developers, and other stakeholders in the domain of qualitative negotiation, the metamodel should *not* be considered final, since further research and consensus-building will be needed. A possible future path is that various vendors arrange to create a standard, informed by what we are proposing here.

In the next chapter, we will discuss a prototype called SoarNego. SoarNego that implements some of the negotiation use cases and core features expected in a tool that will support qualitative negotiation effectively as discussed in Chapter 5

Chapter 8 SoarNego

In this chapter, we describe a prototype called SoarNego that implements some of the features discussed in Chapter 5, derived from our research as discussed in Chapter 4 and also taking consideration of the metamodel from Chapter 6. The features are the baseline requirements that a technology for qualitative negotiation should support. SoarNego is an open-source artifact available to negotiators at https://github.com/umple/SoarNego.

8.1 SoarNego Development

SoarNego is a full stack application, and the modules were developed incrementally. As presented in Figure 37, it has front end and back end. The front-end exchanges data with the back end through the restful APIs. Such separable back and front end gives flexible architecture that could be extended later in the future and useable by different front ends. For instance, it would need little or no adaptation to make SoarNego useable on mobile devices.

Also, we were intentional in our choice of the libraries that were used to develop SoarNego. We considered only libraries that are open source, stable, popularly used in industry and have robust-community driven support. Our goal is to enable a more-flexible software development process and to end up with a tool that will be adaptable and customizable.

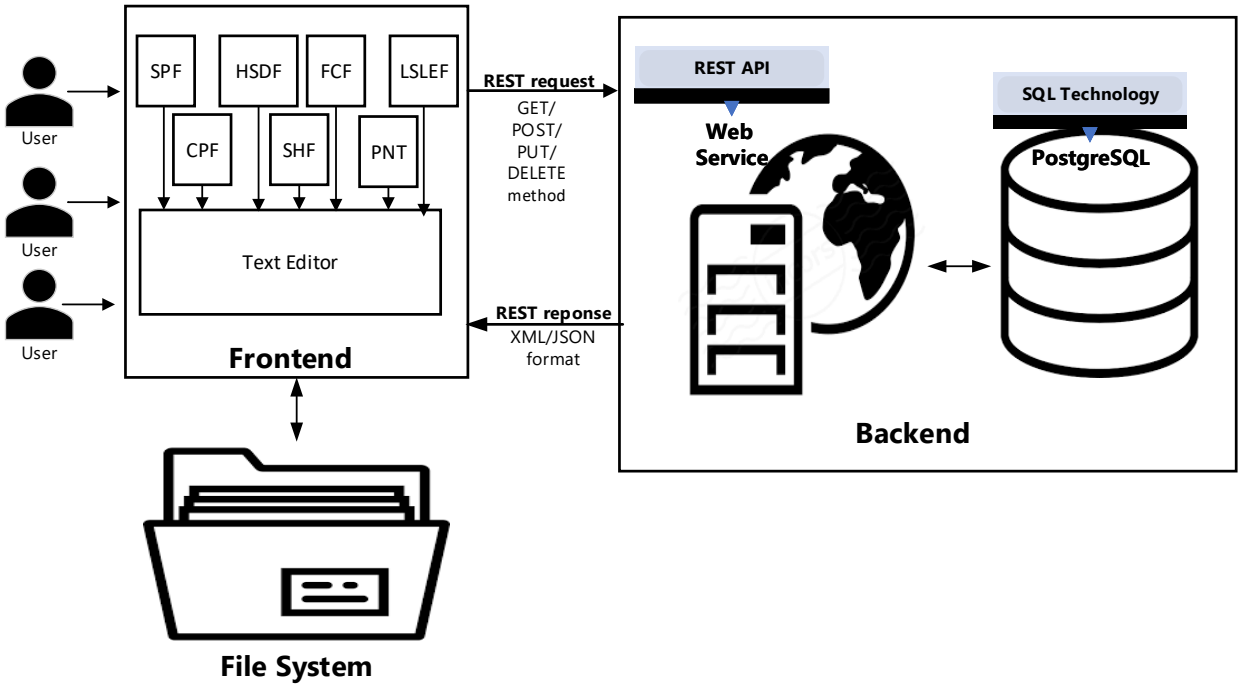

Figure 37: SoarNego system architecture

8.2 SoarNego Design

In this section, we describe SoarNego's front and backend designs.

8.2.1 Frontend

The frontend part (SoarNego interface) was designed with React. React (https://react.dev/) is an open-source front-end JavaScript library created by Facebook's Jordan Walke for building user interfaces for both web and mobile systems. React has ability to show parts of the UI that are changing without re-rendering the whole page, making it suitable for applications that change data quickly and dynamically. This is what SoarNego is all about, that is, the flexibility of using the selected/highlighted text on the editor to filter information that the user wants to see. The filtered data fetched from the backend or already loaded metadata in the session would swiftly be displayed without user experiencing any full-page re-rendering.

As shown in Figure 37, the text editor is the cardinal element of SoarNego. We explored widely-used editor libraries CKEditor (https://ckeditor.com/) and ProseMirror (https://prosemirror.net/), but we eventually settled for Remirror (https://remirror.io/). CKEditor

is a JavaScript rich text editor with MVC architecture, custom data model, and virtual DOM. It had almost all we needed in SoarNego, but some of its features are not free. ProseMirror is a Toolkit library for building a rich text editor; it is open source and legally free to use commercially. However, we opted for Remirror because it is a wrapper library for ProseMirror. It provides React and Prosemirror integration, which makes it easier to work with as compared to ProseMirror, so it works better in the context of our project that uses React for the frontend.

8.2.2 Backend

The backend was developed with Spring Boot (https://spring.io/projects/spring-boot). This is an open-source Java-based framework that is used to develop stand-alone production-ready applications. Spring Boot has been widely used for over two decades. It is a combination of Spring framework and embedded servers; therefore, we do not need to configure any server explicitly, thereby saving us development, unit test and integration test time. Figure 37 presents the architecture of SoarNego.

We adopted RESTful APIs to exchange data between the frontend and backend of SoarNego. REST (representational state transfer) is an architectural style providing standards between computer systems on the web. The high-level abstraction sees any data that would be saved to or exposed from SoarNego backend as a resource. Each resource has an assigned URL such as /Party/Negotiator Proposal/1.

A resource also may have the following representation: XML, HTML or JSON. PostgreSQL, an open-source relational database management systems is used SoarNego data.

8.2.3 The SoarNego Interface

Full architecture and instructions to run SoarNego is on Github: https://github.com/umple/SoarNego/wiki/Architecture. Next, we discuss the prototype interface, illustrated by screenshots.

8.2.3.1 Default Page of the SoarNego

Figure 38 presents default page that is seen by a user that has successfully logged in to SoarNego. It is divided into the following panes:

A. **Text Editor**: Pane A is text editor part of the SoarNego. Content of a selected file from Pane B is displayed in the text editor and the file name is displayed at the top of the page. The editor is the core of the SoarNego application. This is where changes to files occur.

B. **File Explore**r: Pane B is a file directory where all files accessible to SoarNego in a session is listed. Every listed file on this pane, must have been loaded as a file with .docx from directory on a system where SoarNegor is explored. SoarNego sees files listed in Pane B as working files that could explored in the process of any operation invoked in SoarNego.

C. **Menu Ba**r: Pane C lists features of the SoarNego. Some of these features may be applicable to selected text on the editor or whole file content.

D. **Result Consol**e: Pane D displays the results of the SoarNego features' operations.

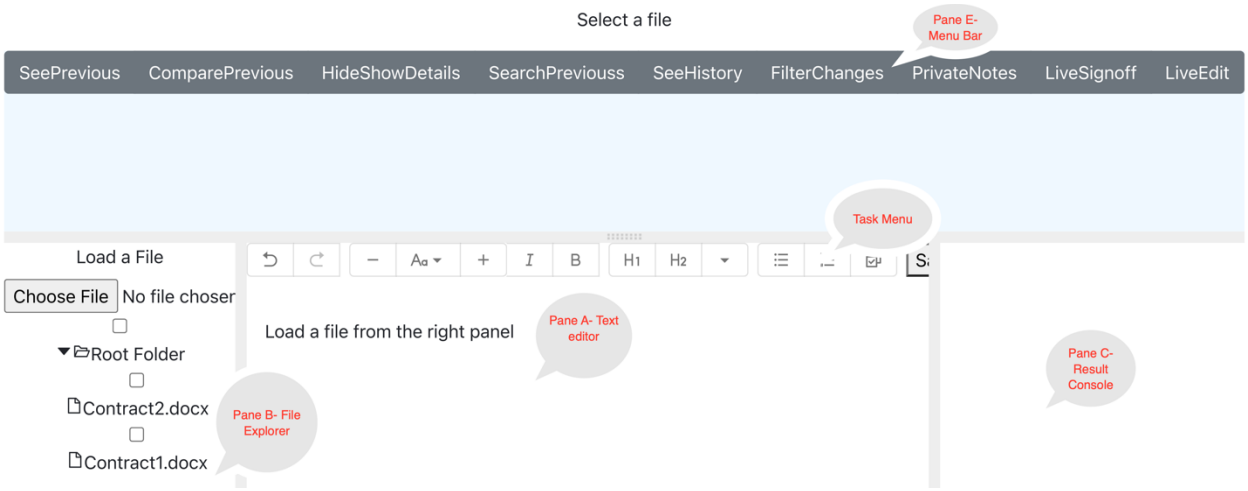

Figure 38: SoarNego Default Screen

8.2.3.2 The ComparePrevious Page

This is one of the main features of the SoarNego application. It will allow a user to select a file, compare it with another file (possibly another version of the same file), and see the difference between both files in Pane C. Deleted text is highlighted in green, added text highlighted in red, and unchanged text is left unhighlighted. Figure 39 presents the ComparePrevious screenshot.

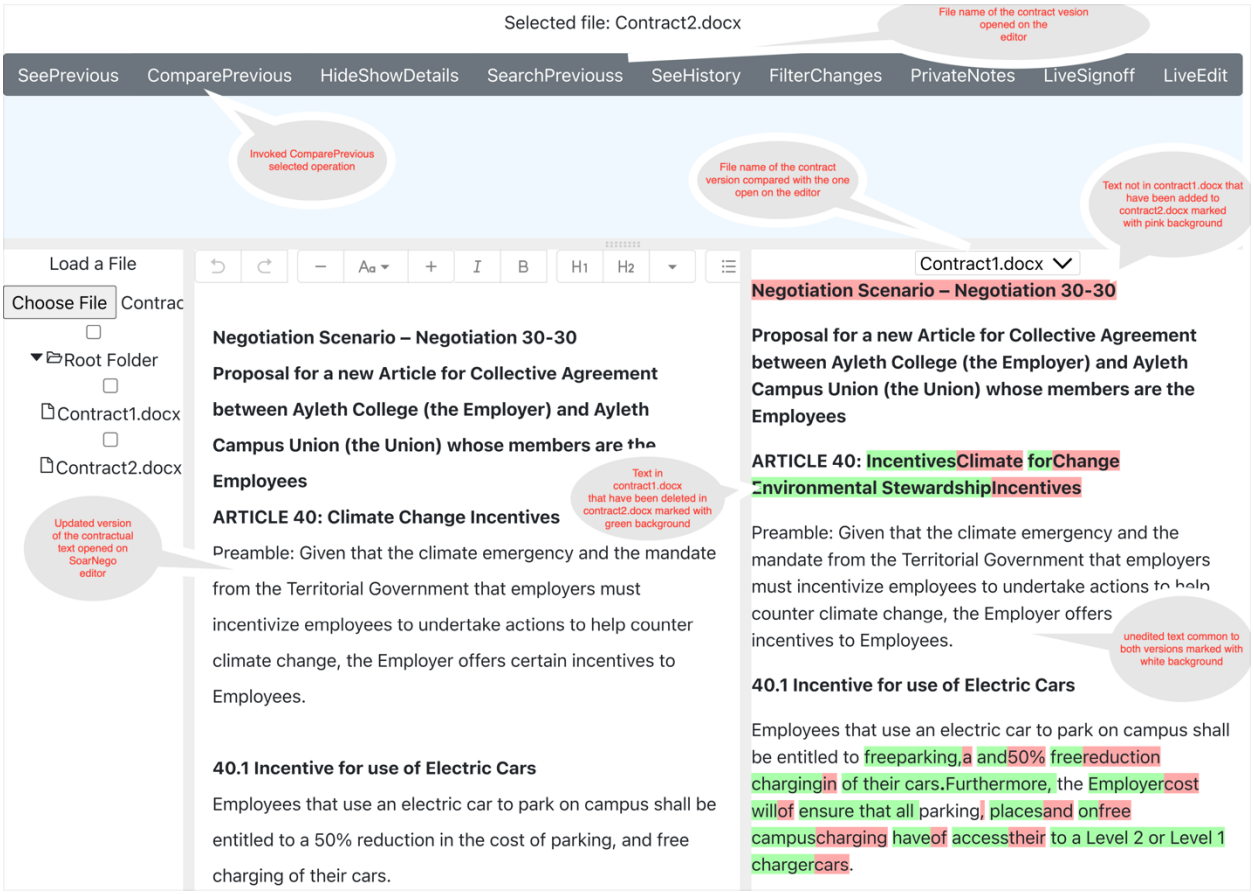

Figure 39: ComparePrevious Screenshot

8.3 Comparison of SoarNego with Negoisst

In this section, we discuss in detail how our SoarNego differs from Negoisst, the only tool we have studied in the literature that is still in use as of the time of writing of this thesis.

As discussed in Section 2.2.3.1, Negoisst constrains negotiators to explicitly tag every message exchange on the tools interface. Such tagging is not feasible in qualitative negotiation due to the large volume of arbitrary text input and editing required during the negotiation process. Since the process is all about back and forth editing of text until agreement is reached, it will be excessively laborious to expect negotiators to explicitly tag every text input.

Message tagging is *implicit* in SoarNego. That is, operation performed by the negotiators imply the type of a text input. Next, we discuss the implicit message types:

1. **Accept**: The receiving negotiator can click an indicator, a small icon attached to block of text as input on the contractual document, in order accept proposed text as it is presented. The text color of the accepted changes to black (indicating the text is now not being proposed by a party, and is accepted) and bold (to indicate that the text is different from what existed when negotiations started).
2. **Reject**: The receiving negotiator can reject a text block. Represented with strike through. Deletion is subject to approval by the initiator of the text or open to editing as discussed in (3) below. Implicitly, reject is proposal for deletion. Which is represented with bold strike through text.
3. **Edit**: A negotiator can edit any text block, with the edits represented with appropriate color coding corresponding to the Party doing the editing. Editing a block of text implies a proposal or counterproposal that is subject to '1' or '2' discussed above.

8.4 SoarNego Forward Thinking

As at the time of this reporting, we have not fully carried out a complete cycle of design science (DS) as presented in the adapted model used to design SoarNego in Figure 12. Our major constraint was time, and we opt to make SoarNego a continuous project. However, we have documented the following key activities, which we intent to carry out in the future to complete the DS cycle and publish our results.

8.4.1 Demonstration of SoarNego

As presented in tables Table 7, Table 8, and Table 12, we intent to enhance SoarNego with more use cases or features that have not been implemented yet. Then, we will demonstrate its usage to negotiators and possibly take feedback through interviews or questionnaires.

8.4.2 Refactoring of SoarNego

We will analyse data to be collected in Section 8.4.1, and use the results to evolve applicable use cases and features. We will redesign SoarNego accordingly.

8.4.3 Beta Testing of SoarNego

We will organise use of SoarNego by real negotiators to conduct negotiation in simulated contexts where no real legal issues will arise. We shall make direct observations, and conduct interviews after the simulated negotiations. We might also administer questionnaires. Analysis of the data gathered through these activities would be used to further evolve use cases, features and redesign SoarNego.

8.5 Possible Value of DS Cycle Iteration

In this section, we discussed possible processes to obtain feedback that may enhance SoarNego. Enhancements may include:

1. Adding features
2. User experience improvement
3. Security hardening
4. Integration with Microsoft Word

8.6 Conclusions

In this section, we described SoarNego development and its architecture. We explained the reasons why we settled for the libraries we have considered or used in the development of SoarNego. SoarNego is a prototype that implements some of the core features that negotiators desired in a tool that will effectively support qualitative negotiation as discussed in Chapter 5. SoarNego is a work in progress and would be enhanced with other features that have not been implemented as at the time of writing this thesis.

The purpose of this chapter is merely to show how a tool that implements features such as SeePrevious might look.

We would like to acknowledge the support of an undergraduate team in course CSI4900 (Winter 2023) who helped with the coding of SoarNego under the technical mentorship and project management of the author of this thesis. The members of that team were: Hamid Abolhassani, Justin Spudic and Amy Wang.

In the next chapter, we present the summary of all we have done during our research and our contributions to the body of knowledge, answers to the research questions we asked in the

beginning of our research (listed in Section 1.2), threats to the validity of our results, our recommendations and feature thinking in the domain of our research.

Chapter 9 Conclusion and future Work

In this research, we have used a multiple-method approaches to study software for qualitative negotiation. Results of our research has produced improved artifacts as models, metamodel and technology that should yield substantial societal benefits spanning from local communities to the global stage in qualitative negotiation domain. These artifacts are open to being reproduced or extended by researchers, accessible and useable by negotiators and will support negotiations to reach decisions more easily and in lesser time that will be of great impact on the society.

The following sections present a summary of the answers to the research questions, key references to the discussions of each chapter, a summary of the thesis contributions to the body of knowledge, a recap of the threats to validity of the results presented in the thesis, as well as recommendations and forward thinking regarding the domain of textual negotiation.

9.1 Comparison with Prior Qualitative Negotiation Technology

Agora (discussed in this Section 2.3.1.3) was one of the early tools that allowed some of the same features such as tracking comments and dividing the document into paragraphs (i.e. clauses). However, our model goes much further as discussed is Chapter 7 and Chapter 8.

In Section 7.5, we compare how our metamodel differs from the existing ones that we have studied in the literature.

Table 9 compares availability of both initial use cases found in the commercial tools to the expanded set that we uncovered during our research.

Table 12 analysed the degree to which common text editors, our proposed technology and the prototype developed during this research, support features required in a tool that will effectively support qualitative negotiation.

9.2 Answers to Research Questions

In this section, we provide summaries of the answers to the research questions we asked in Chapter 1 (Section 1.2). We discuss the answers to the four research questions next:

RQ1. What **processes and tools** are currently used by people negotiating qualitative (textual) agreements, and what are the strengths, weaknesses and requirements gaps in these?

To answer RQ1, we divided it into five objectives. We summarize the answers to each of them separately from our GT and survey results. We have earlier discussed the details of these results in the individual chapters.

a. **Current process of qualitative negotiation**: Electronic contractual documents are the media and instruments for the kinds of negotiation we have studied. Each document is created and shared by a party with the other party or parties during negotiation. The receiving party or parties could accept the document as it is shared (this rarely happens), edit the document to represent their positions or proposed compromises using either a change-tracking mechanism, marking text (in bold, italics or underline), surrounding text in some form of bracket, or using other markup as specified in the protocols set for the negotiation. There is almost always back and forth editing of the document between the negotiating parties until agreement is reached (as discussed in Section 4.1).
b. **Current tools used by the people in qualitative negotiation**: MS-Word is the common text editor tool use by the negotiators we interviewed to perform the editing discussed in "a". Other categories of tools are used by the negotiators for:
    i. Transmission of documents e.g. email services, sheets of paper
    ii. Sharing of documents e.g. DocuShare, Drive, Dropbox, Google Drive, SharePoint, Sync etc.
    iii. Teleconferencing e.g. Zoom, Microsoft Teams, etc
    iv. Presentation and Accounting, e.g. spreadsheets
c. **Strengths of the current processes and tools**: MS-Word is the tool use by all the negotiators whose data were collected via interview or survey during this research. Simplicity and ease of use are the attribute of MS-Word that makes it the choice of negotiators. According to the negotiators, they do not need to undergo any training before making use of MS-Word, as a majority of negotiators already use the software.
d. **Weaknesses of the current processes and tools**: The results of the GT analysis revealed that the existing tool, MS-Word, used by the negotiators does not have

sufficient capabilities, particularly the capability to manage, view, search and compare history, notes, and background information regarding agreements (as discussed in Section 5.4) and in particular to compare multiple versions. Negotiators face difficulty to perform these specific operations that qualitative negotiation requires. Typically, they resort to manual processes to carry out such operations.

e. **Requirements gaps of the current processes and tools**: Existing research efforts have been focused on *quantitative* negotiation, as opposed to qualitative negotiation. As a result, existing tools have been designed to focus on quantitative negotiation (as discussed in Chapter 2).

RQ2. What would be an appropriate **metamodel** (abstract syntax) that would capture the structure and content of qualitative (textual) agreements and negotiations in general, enabling tools to be created that would in turn enable agreements to be better analyzed and negotiated?

As discussed in Chapter 7, our research has resulted in design of a comprehensive metamodel for qualitative negotiation.

RQ3. How can we **improve the process** of creating, modifying and negotiating qualitative agreements through technology?

a. **Advocacy for qualitative research**: There is a need to study and understand qualitative negotiation, so as to spur development of models, metamodels and tools that will effectively support the process. This thesis and resulting papers will hopefully spur such research.

b. **Tool development**: There is a need for development of open-source tools that implement qualitative negotiation support tool use cases and requirements such as those we discussed in Chapter 5, and prototyped in Chapter 8. Tool developers would benefit from a thorough understanding of the processes of qualitative negotiation, such as those we derived in our grounded theory analysis and presented in Figure 13. It would be ideal of all of such tools had consistent models so they can interoperate, such as using a common metamodel like

that which we defined in Figure 27. These artifacts all serve as baselines for future researchers and developers that may want to work on textual negotiations.

RQ4. Why are sophisticated and **existing negotiation tools not being universally used**?

From our literature reviews, GT and survey results, we discovered the following as the reasons why the existing sophisticated negotiation tools are not being universally used:

a. **Access Restriction**: Existing tools are not open source and only accessible through subscription which restrict their access to the negotiators (as discussed in Section 5.3). Most current likely cannot cost-justify a specialized tool, especially since all parties would have to agree on the same tool (or at least a tool that would interoperate with Word files as being used by other parties). Any new tool should therefore be open source and freely available.
b. **Resistance to Change**: Negotiators have for many years used MS-Word. Despite the limitations of MS-Word, limiting them from performing some specific qualitative negotiation operations, they are reluctant to try out another tool. Some of them said training would be a key reason for such reluctance. Any new tool, such as that which we proposed in this thesis, would need to be as similar to Word as possible, and to be able to import and export from work with ease.
c. **Security Concerns**: Negotiations have to be conducted under confidential conditions, so negotiators are concerned about the access authorisation and integrity of the data committed to the specialized negotiation tools. When discussing whether or not they would be willing to use such tools, one of them said, "…if [provided] it will get past [the] kind of security classification problem-P#2" that is required in qualitative negotiations. Any new tool would have to carefully consider security.
d. **Lack of Information**: Many of the negotiators that we talked to have never heard anything about specialized negotiation tools before the interview session. For instance, an interviewee said, "I have never heard of them. I don't know anything about them Maybe they will be helpful … But they have never done any marketing to make them gain attention" P#2. In order for a new tool to become widely used it would have to gain a

foothold in the negotiation community, and being easy to use while solving current problems might help spur the process towards that.

9.3 Summary of Contributions

The key contributions of this thesis are listed next:

1. **Gap identification:** Existing efforts have primarily focused on quantitative negotiation where the main subject of agreement is numeric, such as prices in supply chain or service chain negotiations. However, there are many gaps in understanding how software could assist text-intensive negotiators, such as for treaties or labor-management agreements; there is a need for models, theories and tools for this type of negotiations, which we refer to as *qualitative* negotiation. These gaps are discussed in our literature review (Chapter 2)
2. The **notable observation that MS-Word remains the common support tool** used by most qualitative negotiators in multiple domains. This is despite the limitations it poses, such as the inability to rapidly analyse the history of a clause in an agreement.
3. A **model of issues in qualitative negotiation** arising from the grounded theory results. This model is a source of information to understand negotiation and the type of operations that are carried out during negotiation (Section 4.2).
4. A **comprehensive set of use cases for software support of the qualitative negotiation process** (Sections 5.1 and 5.2). These were triangulated from all the research methods.
5. A set of **requirements for needed features for a qualitative negotiation tool**. These build on the use cases (item 4 above). They are described in Section 5.4, and validated by a survey of 36 negotiators in Chapter 6.
6. A **negotiation metamodel**, that provides a source of information about qualitative negotiation (Chapter 7). It can be used for generation of negotiation data models, or even of code through a tool such as Umple. The metamodel is published in UmpleOnline and is accessible there as an example of the class diagrams listed on Umple (https://cruise.umple.org/umpleonline/ On the page that appears, select Examples/Class Diagrams/Negotiation).

7. **An open-source prototype** tool (SoarNego) that implements some key features listed in item 5 above. Prior to this research we were not aware of any open-source tool to support qualitative negotiations. This tool is discussed in Chapter 8.

9.4 Threats to Validity and How We Addressed the Threats

Both qualitative (interview) and quantitative (survey) data used for this research were collected from different sources, analysed, and interpreted sequentially. We also created models (process model, metamodel, tool prototype) that we analysed with respect to the other data. Inherently, "these different ways of gathering information can supplement each other and hence boost validity and dependability of the data"(Zohrabi, 2013).

We conducted each activity of our research with validity in mind, as discussed in the relevant chapters. "Validity is concerned with whether our research is believable and true and whether it is evaluating what it is supposed or purports to evaluate" (Zohrabi, 2013). Next, we examine four sources of potential threats to the validity of the results of mixed method research approach used in this research. The analysis is guided in part by a study of mixed method research by Zohrabi M. (Zohrabi, 2013).

9.4.1 Content Validity

This type of validity relates to whether we adequately and effectively investigated the breadth of the issues we are investigating. In other words, did we capture and address essentially all the important challenges faced in the process of qualitative negotiation for which technological assistance might prove useful.

To address this, we conducted interviews with negotiators and distributed the survey to a wide range of negotiators. We targeted negotiators who do multiple types of qualitative negotiation, including labor negotiation, and inter-governmental negotiation. We also made sure we balanced the data collected by interviewing representative from both sides in the context of union-management negotiation.

We prepared semi-structured interview questions in advance for the participants in which we explored a wide range of issues gathered earlier in the supervisor's experience of negotiation. The questions were drafted by the researcher, and jointly reviewed by both the researcher and the

supervisor in multiple passes to ensure the question are clear and unambiguous. The passes continued as new ideas arose during early interviews. In each pass, some questions were reformulated, or split into two or more questions for clarity. During the interviews, we reinterpreted questions that were suspected not to be clear to the interviewee, and we asked additional questions to ensure we understood the interviewee's response and the context of the answer. The subject matter we gathered from the interviewees was thus much broader than the initial set of questions.

The breadth in the domain uncovered during the SLR, the grounded theory (as discussed in the last paragraph) and the negotiation experience of the supervisor all helped ensure the metamodel comprehensively covered the needed features.

There remain some limitations in our work, which we will also mention again under 'external validity'. There may be qualitative negotiation issues that could benefit from automated support that are unique to domains where we did not conduct interviews or obtain survey responses.

9.4.2 Internal Validity

Internal validity, also called credibility, relates to the degree to which we have observed and measured what we ought to consider in the domain of qualitative negotiation. We applied the following five methods to boost the internal validity of the research data and tools (Zohrabi, 2013):

a. **Triangulation of methods and data sources**: The concept here is that if one uses multiple methods, from multiple sources, and the results are consistent, this raises confidence in one's conclusions. We addressed this using four methods, that all contributed to the ideas expressed the results such as our proposed features and metamodel:
    - The author's supervisor had been involved in negotiation for many years, including during the period of work on this thesis. His negotiation participation and experience led to a set of ideas that he communicated to the author, and also served to cross-check and correct perceptions the author obtained through the other methods.
    - The **grounded theory** study generated ideas that were core to the final results
    - The **survey** validated ideas from both the grounded theory and action work.
    - The **metamodel design** work uncovered some details that we had not considered.

- The **prototyping** work in the creation of SoarNego helped fine tune some of the results from the other methods.

We discussed at the end of each section the data collection, and how the results of each method used corroborate one another.

b. **Member checks**: The concept here is that following a qualitative data gathering process, the results should be shown to members of the community for validation and further input. We addressed this as follows:
    - We created the interview questions in part based on perceptions obtained through participation of the author's supervisor at negotiation table; the grounded theory work therefore helped validate, correct, and deepen those perceptions. We were careful to ensure, however that we retained the principles of grounded theory, by avoiding contaminating the process by communicating preconceived notions to the interviewees.
    - As the interview process proceeded, we updated the questions based on what we had learned from previous interviews so subsequent interviews could validate what we learned from earlier ones. This is an integral part of the grounded theory process.
    - We piloted the survey with a couple of negotiators and made some adjustments to correct potential confusion before sending it to a large set of people.
    - Finally, we created the survey question directly from the grounded theory results, and then administered the survey to most of the people we interviewed, as well as to others. This gave many people in the negotiation community room to agree or disagree with the correctness and plausibility of the grounded theory results, thus helping validate them.

c. **Long-term observation**: The concept here that, "repeated observations over an extended period of time can naturally enhance the validity of research data and findings" (Zohrabi, 2013). Participation of the supervisor at the negotiation table happened over a lengthy period, so partially addressing this. However, due to the naturally limited timeframe of a PhD thesis, we were not able to perform the other research methods over many years as

would have been ideal. We leave it to future research to conduct further interviews and further surveys, as well as to bring our metamodel prototype tool into real use and observe their effectiveness.

d. **Peer examination**: The concept here is that methods and data, particularly in qualitative research, should examined by multiple researchers to search for potential confounding issues and to ensure more perspectives are considered. We have addressed this as follows:
    - Our SLR protocol and the first draft of its results was examined by peers and a professor in a course.
    - Our grounded theory process and results were peer reviewed in a published paper.
    - Our metamodel was peer reviewed in a submitted paper, with feedback received and incorporated in this thesis, although no paper has yet been published.
    - The author and his supervisor took independent notes during many of the interviews, and separately analysed them, then compared results in order to home in on the results presented in this thesis.
    - A separate PhD student peer-reviewed the survey prior to it being piloted by negotiators.

e. **Avoidance of researcher bias**: The concept here is that it is risky to have a single negotiator do data analysis alone, because that researcher might incorporate personal bias or make mistakes in formulating following the process.

    We addressed this by having the researcher's supervisor involved in the following:
    - Validating the interview protocols.
    - Being involved in many interviews, so as to be able to ask questions from different angles or to help fill in gaps.
    - Independently analysing the interview data.
    - Being involved in creation of the survey, so as to help avoid ambiguity and validate its coverage.
    - Reviewing the metamodel at several stages of its development.

- Reviewing the SoarNego prototype at several stages of its development.

9.4.3 External Validity

External validity, also called transferability, is "concerned with the applicability of the findings in other settings or with other subjects" (Zohrabi, 2013). That is, can we generalise that the results of our research such that it can be claimed to be applicable to qualitative negotiators, other than those we collected data about.

We addressed this as follows:

The scope of data collected from the interviews covered a broader spectrum of negotiation types beyond our grounded theory work, yet the results corroborated and expanded the interview data. This increases confidence that our results can be generalized for other instances of qualitative negotiations in international trade, labor-management, and qualitative commercial negotiations. But it should still be considered to be limited as to new types of qualitative negotiations that we have not covered, such as formal legal negotiations (e.g. in family law).

9.5 Ensuring Future Acceptance of SoarNego by Negotiators

In this section, we discuss human resistance to new technology as a threat to SoarNego, or any other tool that adopts the models we have presented in this this thesis. We also discuss ways to mitigate such resistance.

As discussed in Section 9.2, *ease of use* and human *resistance to change* are key reasons why Microsoft Word is commonly used by all the negotiators we talked to during our research. To ensure that resistance to change does not prevent future adoption of the advances we have proposed, the Technology Acceptance Model (TAM) can provide guidance.

The TAM has been used over the years to explain adoption of technological systems. "Many studies on adopting new technologies have engaged TAM and they all prove that TAM is an extensively appropriate model that describes the reasonable extent of the adjustment in the adoption of technology and use" (Dube et al., 2020).

TAM speculates that technology adoption is determined by perceived ease of use and perceived usefulness (Dube et al., 2020; Lai, 2017). To boost the adoption of SoarNego and other aspects of our work by negotiators, the following is our approach:

1. **Ease of use**: Intentionally, we have designed the SoarNego interface to look like that of Microsoft Word with a few additional buttons. We may also consider the possibility of making SoarNego an add-on tool of Microsoft Word in the future. It will also be important to conduct usability studies to optimize ease of use.
2. **Usefulness**: The features we have embedded in SoarNego are explicitly designed for usefulness, based on our research results and particularly the validation of these results in our survey. However, the challenge will be to validate that usefulness in practice by evolving the tool and beta testing.

In addition to the above factors, another key issue will be *raising awareness*. We already discovered that negotiators are not aware of commercial tools, so it may even be more challenging to raise awareness of an open-source technology that does not have a large corporate budget. As discussed in Section 8.4, we will showcase the usefulness of SoarNego during demonstrations to negotiators, and hope that awareness then spreads among their networks of colleagues. We will also consider organizing workshops. Also, in addition to the publication listed in Section 1.4, more of our results will be published.

9.6 Future Work and Recommendations

In the following, we discuss some potential avenues for further research and extension of the contributions of this study as discussed in the earlier chapters.

Exploration of artificial intelligent (AI) methods will further harness the effectiveness of the features defined in Chapter 5 which will enhance user experience of the prototype we have developed. For instance, there could be a customised interface recommendation based on the prior features used in the process of negotiation by a user, a chat board module could be built into the current prototype as an alternate access to the features defined in Chapter 5.

We suggest relevant intensive negotiation bodies such as UN, EU, WTO, etc should fund an open-source project that will produce industry standard version of the metamodel and prototype built in this study. Such tool would have significant positive impact in the negotiation domain and make life easier for negotiators.

One of the limitations of the prototype developed in this study is processing of the pre-existing contractual agreements printed on paper or in pdf format. It would be useful to add a parser

that would capture such existing agreements and populate a configured data model instance of the metamodel presented in Chapter 7. This would allow future negotiators to use some of the features we have proposed, such as the ability to compare versions of clauses. It would also set up a system for the next round of negotiations, such as updates to treaties, or subsequent rounds of union contracts after they inevitably expire. It should be noted, however, that data about earlier proposals and notes would not be feasible to capture.

Our final suggestion is that others doing requirements analysis of socio-technical domains perform grounded theory analysis similar to what we have presented the study.

References

Adler-Nissen, R., & Drieschova, A. (2019). Track-Change Diplomacy: Technology, Affordances, and the Practice of International Negotiations. *International Studies Quarterly*, *63*(3), 531–545. https://doi.org/10.1093/ISQ/SQZ030

Adolph, S., Kruchten, P., & Hall, W. (2012). Reconciling perspectives: A grounded theory of how people manage the process of software development. *Journal of Systems and Software*, *85*(6), 1269–1286. https://doi.org/10.1016/j.jss.2012.01.059

Baccarin, E., Madeira, E. R. M., Medeiros, C. B., & der Aalst, W. M. P. (2011). SPICA'S MULTI-PARTY NEGOTIATION PROTOCOL: IMPLEMENTATION USING YAWL. *INTERNATIONAL JOURNAL OF COOPERATIVE INFORMATION SYSTEMS*, *20*(3), 221–259. https://doi.org/10.1142/S0218843011002237

Baccarin, E., Madeira, E. R. M., Medeiros, C. B., & Van Der Aalst, W. M. P. (2011). Spia's multi-party negotiation protocol: Implementation using YAWL. *International Journal of Cooperative Information Systems*, *20*(3), 221–259. https://doi.org/10.1142/S0218843011002237

Buzan, B. (1981). Negotiating by Consensus: Developments in Technique at the United Nations Conference on the Law of the Sea. *American Journal of International Law*, *75*(2), 324–348. https://doi.org/10.2307/2201255

Carmel, E., Herniter, B. C., & Nunamaker, J. F. (1993). Labor-management contract negotiations in an electronic meeting room: A case study. *Group Decision and Negotiation*, *2*(1), 27–60. https://doi.org/10.1007/BF01384402

Carvalho, L., Scott, L., & Jeffery, R. (2005). An exploratory study into the use of qualitative research methods in descriptive process modelling. *Information and Software Technology*, *47*(2), 113–127. https://doi.org/10.1016/J.INFSOF.2004.06.005

Cellary, W., & Picard, W. (1998). Web-Based Business-to-Business Negotiation Support. *Undefined*.

Chakraborty, S., & Dehlinger, J. (2009). *Applying the Grounded Theory Method to Derive Enterprise System Requirements*. https://doi.org/10.1109/SNPD.2009.102

Charmaz, K. (2006). *Constructing Grounded Theory A Practical Guide through Qualitative Analysis. London Sage Publications. - References - Scientific Research Publishing*. https://www.scirp.org/(S(351jmbntvnsjt1aadkposzje))/reference/referencespapers.aspx?referenceid=1690485

Chiu, D. K. W., Cheung, S. C., Hung, P. C. K., Chiu, S. Y. Y., & Chung, A. K. K. (2005). Developing e-Negotiation support with a meta-modeling approach in a Web services environment. *Decision Support Systems*, *40*(1 SPEC. ISS.), 51–69. https://doi.org/10.1016/j.dss.2004.04.004

Choi, H. R., Kim, H. S., Hong, S. G., Park, Y. J., Park, Y. S., & Kang, M. H. (2005). Implementation of framework for developing multi-agent based automated negotiation systems. *ACM International Conference Proceeding Series*, *113*, 306–315. https://doi.org/10.1145/1089551.1089609

Chun Tie, Y., Birks, M., & Francis, K. (2019). Grounded theory research: A design framework for novice researchers. *SAGE Open Medicine*, *7*, 2050312118822927. https://doi.org/10.1177/2050312118822927

Coleman, G., & O'Connor, R. (2007a). Using grounded theory to understand software process improvement: A study of Irish software product companies. *Information and Software Technology*, *49*(6), 654–667. https://doi.org/10.1016/j.infsof.2007.02.011

Coleman, G., & O'Connor, R. (2007b). Using grounded theory to understand software process improvement: A study of Irish software product companies. *Information and Software Technology*, *49*(6), 654–667. https://doi.org/10.1016/J.INFSOF.2007.02.011

Corbin, J., & Strauss, A. (2012). Basics of Qualitative Research (3rd ed.): Techniques and Procedures for Developing Grounded Theory. *Basics of Qualitative Research (3rd Ed.): Techniques and Procedures for Developing Grounded Theory*. https://doi.org/10.4135/9781452230153

Dawadi, S., Shrestha, S., & Giri, R. A. (2021). Mixed-Methods Research: A Discussion on its Types, Challenges, and Criticisms. *Journal of Practical Studies in Education* , *2*(2), 25–36. https://doi.org/10.46809/jpse.v2i2.20

Dube, T., Van Eck, R., & Zuva, T. (2020). Review of Technology Adoption Models and Theories to Measure Readiness and Acceptable Use of Technology in a Business Organization. *Journal*

of Information Technology and Digital World, *02*(04), 207–212. https://doi.org/10.36548/JITDW.2020.4.003

Fei, Y., & Chen, W. (2007). A multi-agent, multi-object and multi-attribute intelligent negotiation model. *Proceedings - Fourth International Conference on Fuzzy Systems and Knowledge Discovery, FSKD 2007*, *3*, 440–446. https://doi.org/10.1109/FSKD.2007.64

Georgieva, S., & Allan, G. (2008). Best Practices in Project Management Through a Grounded Theory Lens. *Electronic Journal of Business Research Methods*, *6*(1), pp43-52-pp43-52. https://academic-publishing.org/index.php/ejbrm/article/view/1223

Glaser, B. G., & Strauss, A. L. (2017). Discovery of grounded theory: Strategies for qualitative research. *Discovery of Grounded Theory: Strategies for Qualitative Research*, 1–271. https://doi.org/10.4324/9780203793206

Griffel, F., Tu, M. T., Munke, M., Merz, M., Lamersdorf, W., & Silva, M. M. da. (1997). Electronic contract negotiation as an application niche for mobile agents. *Proceedings First International Enterprise Distributed Object Computing Workshop*, 354–365. https://doi.org/10.1109/EDOC.1997.628376

Halaweh, M. (2012). A Case Study of Using Grounded Theory-Based Technique for System Requirements Analysis. *Journal of Information Systems and Technology Management*, *9*(1), 23–38. https://doi.org/10.4301/s1807-17752012000100002

Hevner, A., & Chatterjee, S. (2010). *Design Science Research in Information Systems*. 9–22. https://doi.org/10.1007/978-1-4419-5653-8_2

Hoda, R. (2022). Socio-Technical Grounded Theory for Software Engineering. *IEEE Transactions on Software Engineering*, *48*(10), 3808–3832. https://doi.org/10.1109/TSE.2021.3106280

Hoda, R., & Noble, J. (2017). Becoming Agile: A Grounded Theory of Agile Transitions in Practice. *Proceedings - 2017 IEEE/ACM 39th International Conference on Software Engineering, ICSE 2017*, 141–151. https://doi.org/10.1109/ICSE.2017.21

Hoda, R., Noble, J., & Marshall, S. (2013). Self-organizing roles on agile software development teams. *IEEE Transactions on Software Engineering*, *39*(3), 422–444. https://doi.org/10.1109/TSE.2012.30

Jarke, M., Jelassi, M. T., & Shakun, M. F. (1987). Mediator: Towards a negotiation support system. *European Journal of Operational Research*, *31*(3), 314–334. https://doi.org/10.1016/0377-2217(87)90041-5

Kenny, M., & Fourie, R. (2015). Contrasting classic, straussian, and constructivist grounded theory: Methodological and philosophical conflicts. *Qualitative Report*, *20*(8), 1270–1289. https://doi.org/10.46743/2160-3715/2015.2251

Kersten, G. E. (1985). NEGO — Group decision support system. *Information & Management*, *8*(5), 237–246. https://doi.org/10.1016/0378-7206(85)90001-1

Kersten, G. E., & Lai, H. (2008a). Negotiation Support and E-negotiation Systems. In *Handbook on Decision Support Systems 1* (pp. 469–508). Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-540-48713-5_23

Kersten, G. E., & Lai, H. (2008b). Negotiation Support and E-negotiation Systems. *International Handbooks on Information Systems*, 469–508. https://doi.org/10.1007/978-3-540-48713-5_23

Kersten, G. E., & Noronha, S. J. (1999). WWW-based negotiation support: design, implementation, and use. In *Decision Support Systems* (Vol. 25, Issue 2, pp. 135–154). North-Holland. https://doi.org/10.1016/S0167-9236(99)00012-3

Kersten, G., & Lai, H. (2021). *E-Negotiations: Foundations, Systems, and Processes*. https://doi.org/10.1007/978-3-030-49629-6_22

Kersten, G., & Lo, G. (2001). Negotiation Support Systems and Software Agents in E-Business Negotiations. *ICEB 2001 Proceedings*.

Kitchenham, B., Pearl Brereton, O., Budgen, D., Turner, M., Bailey, J., & Linkman, S. (2008). Systematic literature reviews in software engineering – A systematic literature review. *Information and Software Technology*, *51*, 7–15. https://doi.org/10.1016/j.infsof.2008.09.009

Klenk, A., Beck-Greinwald, A., Angst, H., & Carle, G. (2012). Iterative multi-party agreement negotiation for establishing collaborations. *Service Oriented Computing and Applications*, *6*(4), 321–335. https://doi.org/10.1007/s11761-012-0119-x

Lai, P. C. (2017). The Literature Review of Technology Adoption Models and Theories For The Novelty Technology. *JISTEM - Journal of Information Systems and Technology Management*, *14*(1), 21–38. https://doi.org/10.4301/S1807-17752017000100002

Lethbridge, T. C., Forward, A., Badreddin, O., Brestovansky, D., Garzon, M., Aljamaan, H., Orabi, A. H., Orabi, M. H., Abdelzad, V., Adesina, O., Alghamdi, A., Algablan, A., Zakariapour, A., Lethbridge, T. C., Forward, A., & Badreddin, O. (2021). Umple: Model-driven

development for open source and education. *Science of Computer Programming*, *208*, 102665. https://doi.org/10.1016/j.scico.2021.102665

Leyva-Moral, J. M., Palmieri, P. A., Loayza-Enriquez, B. K., Vander Linden, K. L., Elias-Bravo, U. E., Guevara-Vasquez, G. M., Davila-Olano, L. Y., & Aguayo-Gonzalez, M. P. (2021). 'Staying alive' with antiretroviral therapy: a grounded theory study of people living with HIV in Peru. *BMJ Global Health*, *6*(10), e006772. https://doi.org/10.1136/BMJGH-2021-006772

Maglyas, A., Nikula, U., & Smolander, K. (2013). What are the roles of software product managers? An empirical investigation. *Journal of Systems and Software*, *86*(12), 3071–3090. https://doi.org/10.1016/j.jss.2013.07.045

Masood, Z., Hoda, R., & Blincoe, K. (2022). Real World Scrum A Grounded Theory of Variations in Practice. *IEEE Transactions on Software Engineering*, *48*(5), 1579–1591. https://doi.org/10.1109/TSE.2020.3025317

Moher, D., Liberati, A., Tetzlaff, J., & Altman, D. G. (n.d.). *Preferred reporting items for systematic reviews and meta-analyses: the PRISMA statement David Moher and colleagues introduce PRISMA, an update of the QUOROM guidelines for reporting systematic reviews and meta-analyses*. https://doi.org/10.1136/bmj.b2535

Nyhart, J. D. (J. D., Antrim, L., Capstaff, A. E., Kohler, A. D., & Leshaw, D. (1978). *A Cost model of deep ocean mining and associated regulatory issues*. https://repository.library.noaa.gov/view/noaa/10011

Ohta, R., Matsuzaki, Y., & Itamochi, S. (2021). Overcoming the challenge of COVID-19: A grounded theory approach to rural nurses' experiences. *Journal of General and Family Medicine*, *22*(3), 134–140. https://doi.org/10.1002/JGF2.410

Peffers, K., Tuunanen, T., Rothenberger, M. A., & Chatterjee, S. (2007). A Design Science Research Methodology for Information Systems Research. *Journal of Management Information Systems*, *24*(3), 45–77. https://doi.org/10.2753/MIS0742-1222240302

Rangaswamy, A., & Shell, G. R. (1997). Using computers to realize joint gains in negotiations: Toward an "electronic bargaining table." *Management Science*, *43*(8), 1147–1163. https://doi.org/10.1287/mnsc.43.8.1147

Reeves, D. M., Wellman, M. P., & Grosof, B. N. (2002). Automated negotiation from declarative contract descriptions. *Computational Intelligence*, *18*(4), 482–500. https://doi.org/10.1111/1467-8640.00199

Schmidt, K., & Simone, C. (1996). Coordination Mechanisms: Towards a Conceptual Foundation of CSCW Systems Design*. In *Computer Supported Cooperative Work: The Journal of Collaborative Computing* (Vol. 5). KluwerAcademic Publishers.

Schoonenboom, J., & Johnson, R. B. (2017). How to Construct a Mixed Methods Research Design. *Kolner Zeitschrift Fur Soziologie Und Sozialpsychologie*, *69*(Suppl 2), 107. https://doi.org/10.1007/S11577-017-0454-1

Schoop, M. (2020a). Negoisst: Complex Digital Negotiation Support. *Handbook of Group Decision and Negotiation*, 1–19. https://doi.org/10.1007/978-3-030-12051-1_24-1

Schoop, M. (2020b). Negoisst: Complex Digital Negotiation Support. *Handbook of Group Decision and Negotiation*, 1–19. https://doi.org/10.1007/978-3-030-12051-1_24-1/COVER

Schoop, M., Jertila, A., & List, T. (2003). Negoisst: A negotiation support system for electronic business-to-business negotiations in e-commerce. *Data and Knowledge Engineering*, *47*(3), 371–401. https://doi.org/10.1016/S0169-023X(03)00065-X

Schoop, M., & Quix, C. (2001). DOC.COM: A framework for effective negotiation support in electronic marketplaces. *Computer Networks*, *37*(2), 153–170. https://doi.org/10.1016/S1389-1286(01)00213-4

Searle, J. R. (1969a). Speech Acts. In *Cambridge: Cambridge University Press*. Cambridge University Press. https://doi.org/10.1017/CBO9781139173438

Searle, J. R. (1969b). *Speech Acts; An Essay in the Philosophy of Language.* Cambridge University Press American Branch, 32 East 57th Street, New York, N.Y. 10022 ($7.50).

Sedano, T., Ralph, P., & Peraire, C. (2017). Software Development Waste. *Proceedings - 2017 IEEE/ACM 39th International Conference on Software Engineering, ICSE 2017*, 130–140. https://doi.org/10.1109/ICSE.2017.20

Shorten, A., & Smith, J. (2017). Mixed methods research: expanding the evidence base. *Evidence-Based Nursing*, *20*(3), 74–75. https://doi.org/10.1136/EB-2017-102699

Sikolia, D., Biros, D., Mason, M., & Weiser, M. (2013). Trustworthiness of Grounded Theory Methodology Research in Information Systems. *MWAIS 2013 Proceedings*. https://aisel.aisnet.org/mwais2013/16

Staron, M. (2019). Action research in software engineering: Metrics' research perspective (invited talk). *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial*

Intelligence and Lecture Notes in Bioinformatics), *11376 LNCS*, 39–49. https://doi.org/10.1007/978-3-030-10801-4_4/FIGURES/5

Stol, K. J., Ralph, P., & Fitzgerald, B. (2016). Grounded theory in software engineering research: A critical review and guidelines. *Proceedings - International Conference on Software Engineering*, *14-22-May-2016*, 120–131. https://doi.org/10.1145/2884781.2884833

Waterman, M., Noble, J., & Allan, G. (2015). How much up-front? A grounded theory of agile architecture. *Proceedings - International Conference on Software Engineering*, *1*, 347–357. https://doi.org/10.1109/ICSE.2015.54

Xie, A.-S., Li, Y.-J., Wang, H., Li, J.-F., & Shang, W. (2003). A genetic algorithm approach to Web-based NSS. *Proceedings of the 2003 International Conference on Machine Learning and Cybernetics (IEEE Cat. No.03EX693)*, *3*, 1862-1865 Vol.3. https://doi.org/10.1109/ICMLC.2003.1259800

Yuan, Y., Rose, J., & Archer, N. (1998). A Web-Based Negotiation Support System. *Electronic Markets*, *8*(3), 13–17. https://doi.org/10.1080/10196789800000033

Zohrabi, M. (2013). Mixed method research: Instruments, validity, reliability and reporting findings. *Theory and Practice in Language Studies*, *3*(2), 254–262. https://doi.org/10.4304/TPLS.3.2.254-262

Appendix 1: Grounded Theory Questions

We present our grounded theory (GT) method in Section 3.1, and the results in Chapter 4. In this appendix we give two versions of our interview questions; the second version evolved from the first in order to incorporate and deepen understanding of themes that emerged in early interviews.

A1.1 GT questions used in early interviews

The following is the first version of the interview rubric we used for the first part of our grounded theory study, as discussed in Section 3.1.3.

Interviewee: __
Date: __
Notes taken by: __
0. Preliminaries

REMEMBER TO TURN ON ZOOM RECORDING
a) … introduce ourselves, and confirm their name and organization
We are going to ask you some question today about software support for negotiation.

a) Are you OK if we record this session? [if no, then will have to email consent form and rely on notes].

b) We will show you on the screen an informed consent form [share screen]. Please confirm orally that you provide you consent to this process?

Thank-you

[unshare screen]

Ok, let's get started.

1. When negotiating a new (agreement/contract/treaty), or modifications to one, what TOOLs do you use among your OWN team members (privately) to prepare proposals to present to other parties? [on paper, in a word processing document, in [specialized tool]]

2. What do you like best about the tools you use to discuss proposals?
3. What drawbacks are there of the tools, in other words, what problems do you face when preparing to negotiate [with regard to the software, and keeping track of your proposals]?

4. [if not answered] When privately discussing proposed changes, how do you keep track of the changes you are thinking about or proposing?

5. How do you communicate your proposed contract changes (i.e. your 'positions', ‘suggested new wording’, ‘concerns’) to other parties? (what tools? Orally?)

6. And similarly, how are the other party’s proposals or counteroffers communicated to you?

7. As compromises or partial agreements are worked out during negotiations, how do you keep track of these (in a live meeting / afterwards)?

8. And how do you ensure that the other parties and you have a consistent view of the evolving agreement (including a view of the changes, so the old and new versions can be compared)?

9. How do you keep track of alternative proposed changes, or conflicting proposed changes? (e.g. ‘we would agree to X if you agree to Y, but we would agreed to W if you agree to Z).

10.How do you keep track of and communicate your argumentation that you make to the other party about why you want certain changes, or why they should accept your proposals? (i.e. your rationale for proposals)

11. Similarly, how do you keep track of their argumentation and rationale?

12. How do you separate confidential argumentation and strategy from argumentation that you present to other parties?

13. [if not answered] What tool support, if any, do you have for helping you manage your negotiation strategy (e.g. prioritizing issues, setting aside contentious points, items you might be willing to concede, items that are absolutely critical, etc.)

14. [if not answered] Can you think of any software features or improvements you would like to have to help
- Prepare your position
- Communicate to the other party
- Keep track of the the argumentation, the status of the agreement and its changes

15. [if not answered] What are the greatest difficulties you face in managing the document being negotiated, the positions of the parties and the changes?

16. [if not answered] Has negotiating online during the pandemic caused any changes to how you use software tools to support negation?

17. [if not answered] Are there any other features we haven't discussed that you would like to see in a tool that could help you negotiate agreements better than you can now?

18. Is there anyone else you think we would benefit from talking to?

19. Is there anything else you would like to add?

Thank-you.

We do intend to create a survey where we will ask a larger group of negotiators questions such as prioritizing their needs for negotiation software. We hope you would be Ok to receive that in a few months.

If you are interested, we will provide you with our published results when we have completed this research.

A1.2 GT questions used in later interviews

Interviewee: __
Date: __
Notes taken by: __

0. Preliminaries

REMEMBER TO TURN ON ZOOM RECORDING

a) … introduce ourselves, and confirm their name and organization

We are going to ask you some question today about software support for negotiation.

a) Are you OK if we record this session? [if no, then will have to email consent form and rely on notes].

b) We will show you on the screen an informed consent form [share screen]. Please confirm orally that you provide you consent to this process?

Thank-you

[unshare screen]

Ok, let's get started.

1. When negotiating a new (agreement/contract/treaty), or modifications to one, what TOOLs do you use among to store or manage the agreement including proposed wording or changes [on paper, in a word processing document, in [specialized tool]]

2. [if not answered] How do you communicate/send/share your proposed contract changes (i.e. your 'positions', 'suggested new wording', 'concerns') to other parties? (what tools? Orally?)

3. [if not answered] how are the other party's proposals or counteroffers communicated to you?

4. [if not answered] When privately discussing proposed changes prior to meeting with other parties, how do you keep track of the changes you are thinking about or planning to propose?

5. [if not answered] How do you keep track of negotiation strategy, including alternative proposed changes, or conflicting proposed changes? (e.g. 'we will present P if they first drop Q', or 'we would agree to X if you agree to Y, but we would agreed to W if you agree to Z).

5A. [if not answered] What tool support, if any, do you have for helping you manage your negotiation strategy (e.g. prioritizing issues, setting aside contentious points, items you might be willing to concede, items that are absolutely critical, etc.)

6. [if not answered] How do you keep track of and communicate your argumentation or rationale that you want to explain to the other party about why you want certain changes, or why they should accept your proposals? (i.e. your rationale for proposals)

7. [if not answered] Similarly, how do you keep track of their argumentation and rationale?

8. [if not answered] As compromises or partial agreements are worked out during live negotiations, how do you keep track of these (in a live meeting / afterwards)?

9. if not answered]How do you separate confidential argumentation and strategy from argumentation that you present to other parties?

10. [if not answered] And how do you ensure that the other parties and you have a consistent view of the evolving agreement (including a view of the changes, so the old and new versions can be compared)?

11. What do you like best about the tools you use to track documents being negotiated?

12. What drawbacks are there of the tools, in other words, what problems do you face [with regard to the software, and keeping track of your proposals and those of the other parties]?

13. We are going to suggest some potential features in negotiation software. Can you indicate whether you think it would be helpful:

a) Ability to quickly go back to previous versions, and quickly see differences between versions or search for wording

b) Ability for any piece of wording to see its 'history' (date of change, which side asked for it; rationale for it; what alternatives were discussed)

c) Ability to hide/show details such as change history, rationale, etc. to manage the complexity

d) Ability to see private notes on the shared document that the other party cannot see

e) Instant editing of proposals, and sign-off of agreements, by both parties in a common document in a live meeting.

14. [if not answered] Can you think of any other software features or improvements you would like to have to help

- Prepare your position
- Communicate to the other party
- Keep track of the argumentation, the status of the agreement and its changes

15. [if not answered] Has negotiating online during the pandemic caused any changes to how you use software tools to support negation?

16. Is there anyone else you think we would benefit from talking to?

17. Is there anything else you would like to add?

Thank-you.

We do intend to create a survey where we will ask a larger group of negotiators questions such as prioritizing their needs for negotiation software. We hope you would be Ok to receive that in a few months.

If you are interested, we will provide you with our published results when we have completed this research.

Appendix 2: Survey Questions

The following are the questions we asked our survey participants, as discussed in Section 3.2.2. Questions 14 and 18 were made mandatory for the participants to answer. Both questions are marked with "*" sign.

Informed consent for this survey

Thank you for agreeing to participate in our survey, which aims to gather information on your experience with negotiations and negotiation tools. We are attempting to design better negotiation technology and we wish to determine the weaknesses of existing tools, and what features negotiators would most like to have available in tools.

There are no known risks from participating in the study.

This survey is being supervised by Dr. Timothy Lethbridge.

The survey should only take about 15 minutes to complete. The questions are anonymous, and all your responses will be kept strictly confidential. Data will be stored in Canada, and will be deleted after five years.

If you have any questions with regards to the ethical conduct of this study, you may contact the Protocol Officer for Ethics in Research, University of Ottawa: ethics@uottawa.ca.

We really appreciate your input!

Do you consent to participate in this survey?

By clicking "Yes, I Consent" below, you consent that you are willing to answer the survey, but you retain the right to withdraw by not clicking 'submit' at the end.

Main survey

The first two questions relate to the way you prepare and discuss changes to existing contract wording in your team **before transmitting it to the other party or parties**

1. **Within your team,** which method do you use in a word processor such as Microsoft Word to mark *proposed deletion* of a part of contractual text? Select all that apply:

- [ ] Using the word processor's **change tracking** mechanism
- [ ] Using the word processor's **commenting** mechanism
- [ ] **Striking through**
- [ ] **Asterisk** beside the location where deletion is to be made
- [ ] Marking text to delete in **bold**
- [ ] Marking text to delete in **italics**
- [ ] Marking text to delete in a specific **color**
- [ ] Any form of **brackets** around text to be deleted
- [ ] **Underlining** text to be deleted
- [ ] N/A (we don't show text to be deleted in any special way)
- [ ] N/A (we don't use a word processor)
- [ ] Other (please specify)

2. **Within your team,** which method do you use in a word processor such as Microsoft Word to *mark proposed addition* of a part of contractual text? Select all that apply:

- [ ] Using the word processor's **change tracking** mechanism
- [ ] Using the word processor's **commenting** mechanism
- [ ] **Asterisk** beside the location where addition is to be made
- [ ] Marking text to add in **bold**
- [ ] Marking text to add in **Italics**
- [ ] Marking text to add in a specific **color**
- [ ] Any form of **brackets** around text to be added
- [ ] **Underlining** text to be added
- [ ] N/A (we don't show text to be added in any special way)
- [ ] N/A (we don't use a word processor)
- [ ] Other (please specify)

The next two questions relate to the format used when **transmitting proposed contract changes to the other party or parties**

3. **When exchanging proposals or counter-proposals with the other party,** which method do you use in a word processor such as Microsoft Word to **mark proposed deletion** of a part of contractual text? Select all that apply:

- [ ] Using the word processor's **change tracking** mechanism
- [ ] Using the word processor's **commenting** mechanism
- [ ] **Striking through**
- [ ] **Asterisk** beside the location where deletion is to be made
- [ ] Marking text to delete in **bold**
- [ ] Marking text to delete in **italics**
- [ ] Marking text to delete in a specific **color**
- [ ] Any form of **brackets** around text to be deleted
- [ ] **Underlining** text to be deleted
- [ ] N/A (we don't show text to be deleted in any special way)
- [ ] N/A (we don't use a word processor)
- [ ] Other (please specify)

4. **When exchanging proposals or counter-proposals with the other party**, which method do you use in a word processor such as Microsoft Word to *mark proposed addition* of a part of contractual text? Select all that apply:

- [ ] Using the word processor's **change tracking** mechanism
- [ ] Using the word processor's **commenting** mechanism
- [ ] **Asterisk** beside the location where addition is to be made
- [ ] Marking text to add in **bold**
- [ ] Marking text to add in **Italics**
- [ ] Marking text to add in a specific **color**
- [ ] Any form of **brackets** around text to be added
- [ ] **Underlining** text to be added
- [ ] N/A (we don't show text to be added in any special way)
- [ ] N/A (we don't use a word processor)
- [ ] Other (please specify)

5. At what **stages** of the negotiation do you **draft proposals for new contractual text or for changes to existing text** (select all that apply)?

- [ ] Before the first interaction with the other party to talk about fresh negotiations
- [ ] After having an initial conversation with the other party, but before negotiations get formally underway
- [ ] During ongoing negotiation

6. Which of the following **most closely describes your recent experience of preparing first draft** of proposals for contractual text **within your own team** prior to presenting to the other party?

- ◯ A common document is shared with team members and each of the team members makes their contributions directly, and everybody can instantly see the changes being made or proposed by others (e.g. Google Docs, Office 360)
- ◯ A document is created by a team head with initial suggestions, shared with team members who edit individual copies; the contributions are later consolidated by the team head
- ◯ Team members submit handwritten or printed documents that are harmonized by the team head
- ◯ Each team member submits an electronic document that is harmonized by the team head or designated team member.
- ◯ Other (please specify)

7. Within your team and during a live virtual negotiation, which of the following would you likely use to **privately communicate information with your teammates**? Select all that apply:

- [ ] Texting, including SMS or instant message software such are WhatsApp, Slack, Microsoft Teams, Zoom chat, Apple messages, etc.
- [ ] Email
- [ ] Telephone call
- [ ] Signal openly (e.g. orally, or body language) that you want to switch to a private caucus, and only communicate private information while in such a private caucus
- [ ] The built-in private communication mechanisms of a commercial negotiation tool
- [ ] Not applicable: Such private communication does not occur
- [ ] Other (please specify)

8. How likely is it that you would do **minor editing** of the contractual text, such as adjustment of a few words, **in collaboration with the other party in a live negotiation session?**

◯ Never ◯ Rarely ◯ Sometimes ◯ Often ◯ Always

9. How likely is it that you would do **major editing** of the contractual text such as adding or rewriting entire clauses **in collaboration with the other party in a live negotiation session?**

◯ Never ◯ Rarely ◯ Sometimes ◯ Often ◯ Always

In the next two questions we would like to find out how you caucus privately with your team members when conducting negotiations online

10. How likely is it that when conducting **virtual negotiation** using a tool like Zoom or Microsoft Teams you would adopt **Breakout rooms to enable private caucus meetings for each party**?

- ◯ We have never participated in *virtual* negotiations
- ◯ Never (although we have participated in virtual negotiations)
- ◯ Rarely
- ◯ Sometimes
- ◯ Often
- ◯ Always

11. How likely is it that when conducting **virtual negotiation** using a tool like Zoom or Microsoft Teams, parties would employ **completely separate meetings when they want to caucus** with their own team members?

- ◯ We have never participated in *virtual* negotiations
- ◯ Never (but we have participated in virtual negotiations)
- ◯ Rarely
- ◯ Sometimes
- ◯ Often
- ◯ Always

12. **Within your team,** how do you use a word processor's **track changes feature** to mark **proposed changes to contractual text** when gathering team members' suggestions?

- ◯ We **don't use** track changes for this
- ◯ We **don't use** track changes for this **very much**
- ◯ We use track changes primarily to mark **small** proposed changes, but **not lengthy** changes (we mark lengthy changes some other way)
- ◯ We use track changes primarily to mark **lengthy** proposed changes, but **not small** changes (we mark small changes some other way)
- ◯ We use track changes to mark **all or almost all** proposed changes

13. **When exchanging proposals/counter-proposals with the other party,** how do you use a word processor's **track changes feature** to mark **proposed changes to contractual text** that you are formally proposing?

- ◯ We **don't use** track changes for this
- ◯ We **don't use** track changes for this **very much**
- ◯ We use track changes primarily to mark **small** proposed changes, but **not lengthy** changes (we mark lengthy changes some other way)
- ◯ We use track changes primarily to mark **lengthy** proposed changes, but **not small** changes (we mark small changes some other way)
- ◯ We use track changes to mark **all or almost all** proposed changes

* 14. As a negotiator, to what extent do you think you would **benefit from each of the following potential features in negotiation software?**

	Definitely	Probably	Possibly	Not likely (but does not bother me)	Not likely (and would prefer **not** to have it available)
Ability to **instantly go back to see previous versions** of wording (either from earlier in current negotiations, or in previous	◯	◯	◯	◯	◯

contracts)					
Ability to **compare wording as it existed in selected different versions** of the document (so additions and deletions, and who proposed them are highlighted)	○	○	○	○	○
Ability to **search through previous versions** (all or selected ones) for certain wording that may have once been present	○	○	○	○	○
Ability to **see on request, for any piece of wording, its 'history'** (date of change, which party asked for it; rationale for it; what alternatives were discussed)	○	○	○	○	○
Ability to **hide/show details such as change history, rationale,** etc. to manage the complexity of the document being viewed	○	○	○	○	○
Ability to **filter** document wording so as to **show or hide only the changes made or proposed** by certain individuals or parties	○	○	○	○	○
Ability to create and later examine **private notes** on the document that the other party cannot see	○	○	○	○	○
Ability by both parties to perform **edits to a common document in a live meeting**.	○	○	○	○	○
Ability to **sign off partial or complete agreements** by both parties in a common document in a live meeting.	○	○	○	○	○

15. Which of the following tools do you use to **share electronic documents within your team?** Select all that apply:

- [ ] DocuShare
- [ ] Dropbox
- [ ] Box
- [ ] Google Drive
- [ ] Microsoft Share Point, OneDrive or Teams
- [ ] Sync
- [ ] A common drive shared among computers in a local network
- [ ] Within a commercial negotiation tool
- [ ] None of the above
- [ ] Other (please specify)

16. How do you **transmit** proposals to the other party? Select all that apply:

- [ ] Email, as attachment
- [ ] Email, in the body of the email
- [ ] Email, as a link to the proposal
- [ ] Instant messaging, as attachment
- [ ] Instant messaging, in the body of the message
- [ ] Instant messaging, as a link to the proposal
- [ ] On a thumb drive or other physical media
- [ ] On paper
- [ ] Using the facilities of commercial negotiation tool
- [ ] Other (please specify)

17. In what **media** do you transmit proposals to the other party? Please select all that apply.

- [ ] Word processing documents
- [ ] Plain text
- [ ] PDF files
- [ ] Paper
- [ ] In the internal format of a commercial negotiation tool
- [ ] Other (please specify)

* 18. How **satisfied or dissatisfied** are you with the following aspects of your current **negotiation process?**

	Very satisfied	Satisfied	Neither satisfied nor dissatisfied	Dissatisfied	Very dissatisfied
Your ability to **retrieve information about clauses and agreements made between parties in the past**, including the evolution history of clauses	○	○	○	○	○
The **tool(s)** you use for **sharing documents** among team members that you selected in Q15	○	○	○	○	○
The process you use to **transmit proposals to the other party**	○	○	○	○	○
Your ability to conduct **virtual (online) negotiations** with the other party using tools such as Zoom or Microsoft Teams.	○	○	○	○	○
Your ability to **manage information other than the contract document** itself, such as the negotiation mandate; meeting agendas, personal notes, and resource material to reference or to back up proposals during a live negotiation session.	○	○	○	○	○
Your ability to **access notes taken during past negotiations** as a reference and in preparation for the future negotiations	○	○	○	○	○

19. How did your experience change regarding the following due to the move from mostly **in-person negotiations** to mostly **online-negotiations** during the Covid-19 pandemic (you may ignore this question if you did not make such a change in your process)

	Much better online	Somewhat better online	Not much difference	Somewhat better in person	Much better in person
Speed of reaching agreement	○	○	○	○	○
Effort to reach agreement	○	○	○	○	○
Overall cost of the process of reaching agreement	○	○	○	○	○
Quality of agreement following negotiation	○	○	○	○	○
Ability to read facial expressions, body language or other non-verbal information from the other party about whether they like or dislike something	○	○	○	○	○
Finding times to meet	○	○	○	○	○

20. To what extent have you considered or used the following **commercial negotiation tools?**

	Never heard of it	Never considered it	Tested it but did not adopt it	Used it for active negotiations but stopped using it	Currently use it
Concord	○	○	○	○	○
Mitratech ContractRoom	○	○	○	○	○
DocuSign CLM (Formerly SpringCM)	○	○	○	○	○
Sirionlabs	○	○	○	○	○
Icertis	○	○	○	○	○
Juro	○	○	○	○	○
Apttus	○	○	○	○	○
Avvoka	○	○	○	○	○
Conga	○	○	○	○	○
ContractPod	○	○	○	○	○

Other (please specify and indicate how much you use it)

21. If you are not using a commercial tool for negotiations, what are the main reasons that make you avoid using such a tool?

22. If you are using a commercial negotiation tool, please describe the feature or features that you find most attractive in the tool

23. In what role do you primarily conduct negotiation?

- ◯ As a chief negotiator
- ◯ As a person who engages with counter-parties in negotiation sessions, but is not the chief negotiator
- ◯ As a member of a negotiation team, but who generally attends live negotiations, but does not speak directly to counter-parties during such sessions
- ◯ Other (please specify)
- ◯ As a member of a negotiation team who does not attend live negotiations
- ◯ Prefer not to answer

24. What type of negotiation do you have recent experience with? (Select all that apply)

- ☐ Labor negotiation on the union side
- ☐ Labor negotiation on the management side
- ☐ Negotiation of inter-governmental agreements or treaties (e.g. trade, climate change etc.)
- ☐ Negotiation among public-sector organizations (other than intergovernmental or labor-management agreements)
- ☐ Commercial negotiation as the buyer
- ☐ Commercial negotiation as the vendor
- ☐ Negotiation of legal agreements not included above (e.g. among litigants, for family law, intellectual property protection etc.)
- ☐ Prefer not to answer
- ☐ Other (please specify)

25. What do you consider to be your level of experience in conducting negotiations?

- ◯ Very high (an expert who has been involved in negotiation for at least 5 years, on many lengthy contracts)
- ◯ High
- ◯ Medium
- ◯ Low
- ◯ Very low (a beginner who has only recently participated in my first negotiations)

26. Please provide any other comments you may have on this topic.

Appendix 3: Negotiation Metamodel Source Code and Instructions

Appendix 3.2 has the code listing of the negotiation metamodel discussed in Chapter 7. This code is also available on GitHub folder https://github.com/umple/agreementNegotiation/tree/main/negotiationMetamodel/src and the set of instructions to run it in Umple are listed in Appendix A3.4.

A3.1 File name: Negotiations.ump

This file is the core metamodel containing the class diagram, represented in the Umple modeling language. The code for this can be found directly on Github at:

https://raw.githubusercontent.com/umple/agreementNegotiation/main/negotiationMetamodel/src/Negotiations.ump

Diagrams generated from this can be seen in Figure 27 and subsequent figures. Appendix A.3.4 is used to specify the details from this file that will be present in each diagram.

```
1   // This model is a metamodel for
2   // negotiation. It can be used for
3   // the development of negotiation tools
4   // or negotiation domain-specific
5   // languages
6
7   // All negotiations have as their
8   // objective the creation or update of
9   // an Agreement. This could be
10  // an international treaty, a
11  // commercial agreement for purchase
12  // and sale, a labour-management
13  // collective agreement, or anything
14  // else
15  //
16  // Note that the title of the
17  // agreement is determined by
18  // the AgreementVersion, since it
19  // can change
20
21  class Agreement {
22    // An internal ID that can be used by an API
23    // to allow for queries about this
24    // specific agreement, irrespective of version
25    autounique internalAgreementID;
26
27    // Agreements are updated over time
```

```
28      // so the latest approved version is
29      // actually the most important
30      1 -- * AgreementVersion;
31
32      displayColor orange;
33    }
34
35    // The legal entities involved in
36    // negotiation are the Parties
37    // and the Users who are involved
38    // in negotiation for one of the
39    // parties
40    class LegalEntity {
41      abstract; // has no subclasses
42      name;
43      address;
44      emailAddress;
45
46      displayColor yellow;
47    }
48
49    // All negotiations involve at least
50    // two parties. These could be
51    // sovereign states, corporations,
52    // unions or individuals
53    class Party {
54      isA LegalEntity;
55      0..1 -- * UserRole;
56
57      displayColor yellow;
58    }
59
60    // A party has UserRoles. The reason for
61    // tracking these is so that we know
62    // who is making proposals, or signing
63    // of on parts of the agreement
64    // Some UserRoles such as Mediator
65    // may be not be associated with a specific
66    // Party
67    class UserRole {
68
69      enum Role {
70        ChiefNegotiator,
71        MemberAtTable,
72        Mediator,
73        Arbitrator,
74        Other
75      }
76      Role role;
77
78      displayColor yellow;
79    }
80
81    // A user may have multiple roles
82    // For example they may be involved in multiple negotiations
83    // for different agreements
```

```
84  class User {
85    isA LegalEntity;
86    userid;
87
88    1 -- * UserRole;
89
90    displayColor yellow;
91  }
92
93
94  // A VisibilityControlledItem is
95  // something that only certain
96  // LegalEntitles are allowed to see
97  // These include Versions of Proposals
98  // or Clauses that may be incomplete or
99  // being prepared by one party; or Notes
100 class VisibilityControlledItem {
101   abstract;
102
103   // Internal ID of the item that can be
104   // referred to by an API that needs to
105   // query specify Versions or Notes.
106   autounique internalItemID;
107
108   // An item may or may not be tagged
109   // with an author
110   0..1 authoredItem -- 0..1 User author;
111
112   // It may be that only one person
113   // such as a person creating a draft
114   // can currently see a version, or
115   // maybe everybody in a party, or
116   // maybe all parties, or maybe just
117   // the chief negotiator and an
118   // assistant
119   *  controlledItem -- 1..* LegalEntity
120     allowedToSee;
121
122   displayColor darkgoldenrod;
123 }
124
125 // An ItemVersion contains
126 // a set of ClauseVersions
127 // (containedClauses)
128 //
129 // Its two subclasses are
130 // either an edited version of the
131 // agreement (AgreementVersion) as
132 // a whole, or else an edit version
133 // of a Clause (ClauseVersion).
134 //
135 // By virtue of the fact that an
136 // ItemVersion is also a
137 // VisibilityControlledItem, it
138 // may be informal, private to
139 // one party, or visible at the
```

```
140  // negotiation table to all parties
141  //
142  // Any change to any contained
143  // clause results
144  // in a new ItemVersion even
145  // if it is a temporary edit
146  class ItemVersion {
147    isA Version;
148    abstract;
149
150    // An AgreementVersion should have a
151    // title but a ClauseVersion may
152    // or may not have a
153    // title to help identify it
154    lazy title;
155
156    // The changeDescription is
157    // optional, but could be used to
158    // identify information such as
159    // that it was presented to a party
160    // or is subject to ratification etc.
161    changeDescription;
162
163    0..1 container -> * ClauseVersion containedClauses;
164
165    displayColor lightsalmon;
166  }
167
168  // An ItemVersion is made up
169  // of ClauseVersions, and the
170  // difference between ItemVersions
171  // is determined by the
172  // differences in the ClauseVersions
173  //
174  // A new ClauseVersion is created
175  // every time the id, title, text or
176  // subclauses are edited
177  //
178  // A ClauseVersion is only part of an
179  // agreement version if it is a
180  // subclause of another ClauseVersion
181  // and/or is a top level clause of an
182  // AgreementVersion
183  //
184  // Moving a ClauseVersion from one
185  // parent to another means changing
186  // both parents
187  class ClauseVersion {
188    isA ItemVersion;
189
190    // The ID of a clause will normally
191    // be a number or a letter, or
192    // for subclauses, a string of these,
193    // often separated by dots.
194    // It will be displayed to the user
195    // it can be changed such as when
```

```
196     // re-ordering, insertion of new
197     // clauses or reparenting take place
198     // At the top level, the version might
199     // include a string such as 'Article'
200     //
201     // The title would appear after the id
202     id;
203
204     // The text is the essential legal
205     // text of the clause
206     // It uses Markdown format
207     // To allow for italics etc.
208     text; // Markdown format
209
210     // A clause may or may not have lower
211     // level clauses called subclauses.
212     // This is maintained in the
213     // containedClauses association
214     // inherited from ItemVersion
215     // Note that if we want to find the
216     // parent we have to navigate down
217     // from the top level
218     // AgreementVersion because otherwise
219     // every change to a clause would
220     // require creating new versions of
221     // its subclauses
222
223     displayColor lightsalmon;
224   }
225
226   // AgreementVersion is a special case
227   // of ItemVersion for the top level
228   // agreements
229   class AgreementVersion {
230     isA ItemVersion;
231
232     // Each agreement version is
233     // between 2 or more parties
234     // which are defined below
235     * -- 2..* Party;
236
237     // An agreement may have a
238     // descriptive Version label that is
239     // made public
240     versionLabel;
241
242     displayColor lightsalmon;
243   }
244
245   // Clauses are the core elements of
246   // an agreement
247   //
248   // Each time a clause is changed
249   // a new ClauseVersion is created.
250   // The history of a clause can
251   // be explored by reviewing its
```

```
252 // versions
253 class Clause {
254   // The internalClauseID is used
255   // to tie together all the versions
256   // of a clause so their history
257   // can be tracked even if the clause
258   // is radically changed over time
259   // This can be used by an API to refer
260   // to a clause irrespective of version
261   autounique internalClauseID;
262
263   1 -- * ClauseVersion;
264
265   displayColor lightcoral;
266 }
267
268 // A Note is information attached
269 // to a Version or Proposal
270 // Since it is visibility controlled
271 // it can be private to just one
272 // user, or to a party, or to
273 // everybody.
274 //
275 // A Note is not a legal part of the
276 // agreement, but may help people
277 // to record what others think,
278 // or why the change is proposed.
279 // Sometimes old notes can be used
280 // to help resolve disputes
281 // about interpretation
282 class Note {
283   isA VisibilityControlledItem;
284
285   // A note is about some subject;
286   // a note can even be about another
287   // note, such as a private
288   // observation about some rationale
289   * -- 1 VisibilityControlledItem
290     subject;
291
292   // Arbitrary text associated with
293   // the item
294   markdownText;
295
296   // By default there is no need
297   // to record a note type, so it
298   // would be Unknown. But if the note
299   // provides evidence or justification
300   // it can be tagged as Rationale,
301   // and if the other party has reacted
302   // in some way, it can be tagged
303   // as Feedback_FromOtherParty to
304   // allow for better searches
305   enum NoteType {
306     Unknown,
307     Rationale_Private,
```

```
308       Rationale_ToReveal,
309       Rationale_Revealed,
310       Rationale_FromOtherparty,
311       Analysis_AboutOtherParty_Private,
312       Feedback_ToReveal,
313       Feedback_Revealed,
314       Feedback_FromOtherParty,
315       ProcessObservations,
316       Data,
317       Other
318     }
319     NoteType noteType;
320
321     displayColor goldenrod;
322   }
323
324   // A Proposal is a set of ItemVersions
325   // that can be linked by 'and', 'or' or
326   // 'excusive or'
327   //
328   // It represents ideas for changes to
329   // an agreement
330   //
331   // The simplest case could be a change
332   // to just one clause
333   //
334   // But many Proposals have multiple
335   // elements, which might be linked
336   // as follows
337   //
338   // Most sets of elements are linked
339   // by 'and' meaning a conjunct
340   //   a conjunct means that all
341   //   changes are requested to be made
342   //   together
343   //
344   // But some elements could be
345   // alternatives with exclusive or
346   // 'either this or that,
347   // but not both'.
348   //
349   // Others could be 'disjuncts'.
350   // 'Any of these ... it does
351   // not have to be all'
352   //
353   // A final approved proposal has to
354   // be a conjunct, or just hava a single
355   // element
356
357   class Proposal {
358     isA Version;
359
360     // This enum describes the kind of
361     // Proposal
362     enum BooleanLink {
363       conjunct,
```

```
      alternative,
      disjunct
    };
    BooleanLink booleanLink;

    // The elements can be
    // ClauseVersions or
    // Proposals (i.e. sub-proposals)
    0..1 -> * Version elements;

    displayColor cornflowerblue;
  }

  // Version is the abstract superclass
  // of Proposal and ItemVersion
  class Version {
    isA VisibilityControlledItem;
    abstract;

    Date versionDate;
    Time versionTime;

    // Each version is a result of
    // changes to a previous version
    * -- 0..1 Version basedOn;

    displayColor lightsalmon;
  }
```

A3.2 File name: NegotiationsItemVersionStateMachine.ump

This file is the state model of ItemVersion. A diagram of this is presented in Figure 36. The file can be found online on Github at:

https://raw.githubusercontent.com/umple/agreementNegotiation/main/negotiationMetamodel/src/NegotiationsItemVersionStateMachine.ump

```
1  // Definition of the state machine for the
2  // ItemVersion class
3  class ItemVersion {
4    sm {
5      // Created by an author
6      active {
7        initiated {
8          present-> underNegotiation;
9        }
10
11       // One or more proposals have been made
```

```
12          // and presented to the other Party(s)
13          underNegotiation {
14            agreedAsDraft -> underVerification;
15          }
16
17          // parties agreed, except for error correction
18          underVerification {
19            agreedSubjectToPartyVote -> tentativelyAgreed;
20          }
21
22          // Tentatively agreed subject to votes
23          tentativelyAgreed {
24            agreedAsFinal -> finalized;
25            stakeholdersVoteAgainst -> underNegotiation;
26          }
27
28          // A formal agreement
29          finalized {
30          }
31          withdraw -> noLongerConsidered;
32        }
33
34        // A version that is permanently off the table
35        noLongerConsidered {
36        }
37      }
38    }
39
```

A3.3 File name: NegotiationsProposalStateMachine.ump

This file is the state model of class Proposal. A diagram of this is presented in Figure 35. This can be seen directly in Github at:

https://raw.githubusercontent.com/umple/agreementNegotiation/main/negotiationMetamodel/src/NegotiationsProposalStateMachine.ump

```
1   // Definition of the state machine for the
2   // Proposal class
3   class Proposal {
4     sm {
5       // Created by an author
6       active {
7         // Constructed, including
8         // sets of ItemVersions
9         // possibly linked by logic operators
```

```
10        initiated {
11          internallyPresent-> underInternalConsideration;
12          withdraw -> obsolete;
13          edit -> initiated;
14        }
15
16        // Being considered by the team in a party
17        underInternalConsideration {
18          present-> underNegotiation;
19          returnToAuthorForRefinement -> initiated;
20          withdraw -> obsolete;
21          considerCounterProposal -> superseded;
22          edit -> underInternalConsideration;
23        }
24
25        // The proposal is live 'on the table'
26        // But no new version of clauses have
27        // yet become the new baseline
28        underNegotiation {
29          agreedAsDraft -> incorporated;
30          takeToCaucus -> underOtherPartyConsideration;
31          considerCounterProposal -> superseded;
32          reject -> underInternalConsideration;
33        }
34
35        // The proposers are waiting to hear back
36        // from the other party
37        underOtherPartyConsideration {
38           presentCounterProposal -> underInternalConsideration;
39           reject -> underInternalConsideration;
40        }
41      }
42
43      // The proposal is now historic
44      // As edits based on it to current
45      // ItemVersion has reached state
46      // agreedAsDraft
47      // The changes made may or may not
48      // end up in a final document
49      incorporated {
50      }
51
52      // The proposal is no longer going to
53      // be returned to the table
54      obsolete {
55      }
56
57      // The originating team received a
58      // counterproposal and consider that
59      // the to be the lead proposal in this
60      // topic.
```

```
61        superseded {
62        }
63    }
64          }
```

A3.4 File name: Filters.ump.

This file is used to automatically generate the figures in Chapter 7. It can be seen directly on Github at:

https://raw.githubusercontent.com/umple/agreementNegotiation/main/negotiationMetamodel/src/Filters.ump

To use this file to generate different diagrams, the following command-line commands can be used, where umple2svg is a shell script available in the dev-tools directory of Umple:

- Figure 27 umple2svgcd Filters.ump -s hideattributes
- Figure 28 umple2svgcd Filters.ump -s hideattributes filterX1
- Figure 29 umple2svgcd Filters.ump filterX2
- Figure 30 umple2svgcd Filters.ump filterX3
- Figure 31 umple2svgcd Filters.ump -s hideattributes filterX4
- Figure 32 umple2svgcd Filters.ump filterX3
- Figure 33 umple2svgcd Filters.ump filterX6
- Figure 34 umple2svgcd Filters.ump filterX7

```
1   filter x {
2     include Nothing;
3   }
4
5
6   // Figure X1. Shows Agreement and neigbours
7   // use -s hideAtributes to avoid showing them
8   filter X1 {
9     include Agreement, Party, AgreementVersion;
10    hops {super 0;}
11  }
12  mixset filterX1 {
13   filter {includeFilter X1;}
14  }
15
16  // Figure X2. Shows parents of AgreementVersion
17  // enables explaining of attributes
18  filter X2 {
19    include Agreement, AgreementVersion;
20  }
21  mixset filterX2 {
22   filter {includeFilter X2;}
23  }
```

```
24
25  // Figure X3. Simple hierarchy of LegalEntity
26  filter X3 {
27    include LegalEntity, UserRole;
28    hops {sub 1;}
29  }
30  mixset filterX3 {
31   filter {includeFilter X3;}
32  }
33
34  // Figure X4. VisibilityControlledItem
35  filter X4 {
36    include LegalEntity, User, VisibilityControlledItem;
37  }
38  mixset filterX4 {
39   filter {includeFilter X4;}
40  }
41
42  filter X5 {
43    include VisibilityControlledItem;
44    hops {sub 1;}
45  }
46  mixset filterX5 {
47   filter {includeFilter X5;}
48  }
49
50  filter X6 {
51    include ItemVersion, Clause, ClauseVersion;
52    hops {super 0;}
53  }
54  mixset filter3 {
55   filter {includeFilter X6;}
56  }
57
58
59  filter X7 {
60    include Proposal, Version;
61    hops {super 0;}
62    hops {sub 0;}
63  }
64  mixset filterX7 {
65   filter {includeFilter X7;}
66  }
67
68  use Negotiations.ump;
69  use NegotiationsItemVersionStateMachine.ump;
70
71
72  filter SM1 {
73    include ItemVersion;
74  }
```

```
mixset filterSM1 {
 filter {includeFilter SM1;}
}


//                                                                  use
https://raw.githubusercontent.com/umple/agreementNegotiation/main/
negotiationMetamodel/src/Negotiations.ump;


```