A knowledge structuring framework to support the design of social media for online deliberation

Andrew Hilts

A thesis submitted in conformity with the requirements for the degree of Master of Information
Faculty of Information
University of Toronto

©2011 Andrew Hilts
A knowledge structuring framework to support the design of social media for online deliberation

Andrew Hilts

Submitted for the Degree of Master of Information
Faculty of Information
University of Toronto

August 2011

Abstract

Although online social media have achieved great success in several notable instances, the design of such systems remains an art. In some specialized areas, such as online deliberation systems for participatory democracy, experiences from many projects have been reported in the research literature. Designers can benefit from knowledge accumulated from these experiences. However, the knowledge is dispersed and not organized for ready access by practicing professionals. This thesis proposes a framework for structuring and codifying design knowledge from published studies to help designers make design choices that will attain design objectives.
Acknowledgements

I would first like to thank my supervisor on this project, Professor Eric Yu, for his encouragement, insightful critiques and valuable suggestions throughout the course of this project and my entire time at the Faculty of Information. To Professor Kelly Lyons, my second reader, I extend my gratitude for her support throughout the project, and especially in helping me to craft the application example presented in this work. I am grateful to Professor Steve Easterbrook for his services as external examiner, and for his help during Abayomi King and I’s initial look at the Climate Collaboratorium in early 2010.

Several other members of the Faculty of Information community have helped me in diverse ways as I completed this thesis. Karen Louise Smith has always been supportive of my work, and I am grateful to have found someone who shared an interest in online democratic participation. I would like to thank Lysanne Lessard for her leadership of the Faculty’s working group on qualitative methods, as well as all members of that group and interested peers who attended my presentation for the group in the summer of 2010. Thank you to Jennifer Horkoff, for introducing me to the Q7 language. Abayomi King, Stephen Tracy, Arlo Murphy, Grant Patten and Alex Wong have all helped support my work over the past two years, and I am thankful for that. Thank you to the participants of the First International Workshop on Modeling Social Media (MSM’10), including anonymous reviewers for their valuable feedback during the project’s early stage.

I am privileged to have worked with Professor Yuri Takhteyev on our geography of open source developers project this past year. Thank you for many interesting discussions and for the financial support this role as a research assistant provided me. Additionally, I would like to thank the other source of my financial support, the University of Toronto Mississauga’s Communication, Culture and Information Technology program; especially to Joel Moody and Lisa Peden, under whom I worked as a teaching assistant. I am also grateful to the University’s John M. Kelly library, which
provided me with a comfortable environment for conducting the majority of the work on this project.

I am finally grateful to Cathy, for her love and support over the years. Your contribution cannot be overstated.
# Contents

1 Introduction .............................................. 1
   1.1 The importance of designing for the public sphere .................. 3
   1.2 Problem Statement .................................. 5
   1.3 Purpose Statement .................................. 8
   1.4 Research Questions .................................. 9
   1.5 Chapter summaries .................................. 10

2 Literature Review ....................................... 12
   2.1 Online Deliberation ................................ 12
   2.2 Design knowledge .................................. 17
   2.3 Approaches to design knowledge reuse ......................... 18
   2.4 Goal-oriented Requirements Engineering ....................... 27

3 Research Methodology .................................... 34
   3.1 Design Science Research in Information Systems ................. 34
   3.2 Document Coding .................................... 39
   3.3 Iterative Design .................................... 43
   3.4 Semi-structured Interviews ................................ 43

4 Design Iterations ......................................... 47
   4.1 Early work .......................................... 47
   4.2 Versions of the GO-DKL framework .......................... 52
   4.3 Meta Model .......................................... 52
   4.4 Analysis Methodology ................................ 57
5 The GO-DKL Framework 70
   5.1 Purpose of the framework .............................................. 70
   5.2 Conceptual Model ......................................................... 72
   5.3 Codification Process .................................................... 75
   5.4 Usage guide: the framework as design springboard ................. 77

6 Application Example 88
   6.1 The FaceBook DSS Project ............................................. 88
   6.2 Project Definition ....................................................... 89
   6.3 Goal Associations and Selections ..................................... 89
   6.4 Impacted and Related Goal Analysis .................................. 92
   6.5 Project Model Configuration .......................................... 93
   6.6 Generated Reports and Models ....................................... 94
   6.7 Discussion ............................................................... 98

7 Discussion of interview findings 100
   7.1 Online deliberation system designer practices ..................... 100
   7.2 Practitioner perspectives of the framework ....................... 105
   7.3 Summary of interview findings ...................................... 110

8 Conclusions 111
   8.1 Strengths and contributions .......................................... 111
   8.2 Limitations .............................................................. 112
   8.3 Future Work ............................................................ 113

Appendices 115
   A1 The Knowledge Base .................................................... 115
   A2 MySQL records to tree-list representation .......................... 116
   A3 Tree-list evaluation label propagation ................................ 118
   A4 MySQL records to Q7 file .............................................. 119

Bibliography 121
List of Figures

2.1 Example NFR softgoal interdependency graph. Based on Chung et al. (2000). . . . . 31
2.2 Evaluation Propagation Rules Showing Resulting Labels for Contribution Links.  
   From Horkoff and Yu (2009). . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 32
2.3 Basic i* syntax . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 33
3.1 Hevner et al.’s information systems research framework . . . . . . . . . . . . . . 35
4.1 The MSM approach: Qualitative abstraction, codification and modeling of aca-
   demic literature . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 51
4.2 Third iteration of meta model . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 54
4.3 Project component of meta model . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 56
4.4 The tabular approach to exploring the data. Yellow = selected; red = negative
   claims; green = positive claims . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 58
4.5 Project goal graph with no intuitive relationships; note the lack of intersecting links
   in the lower areas (lack of clear alternatives) . . . . . . . . . . . . . . . . . . . . . . . 63
4.6 Climate Collaboratorium project goal graph with all intuitive relationships . . . . 65
7.1 Practitioner relevancy rating of design issues; GO-DKL aggregate goals in bold . . 105
List of Tables

2.1 Approaches to design knowledge reuse ......................................................... 27
2.2 Contribution links used in the NFR framework .............................................. 30
3.1 Interview activities ......................................................................................... 44
3.2 Codebook for interview transcripts (design practice) ........................................ 46
3.3 Codebook for interview transcripts (demo evaluation) ........................................ 46
4.1 Major versions of the GO-DKL framework and browser ................................. 52
4.2 GO-DKL terminology ..................................................................................... 59
4.3 Goal classifications for the Collaboratorium Case Study ................................. 61
6.1 Goal associations and inclusions for the FaceBook DSS project ......................... 91
7.1 Interview participants ...................................................................................... 101
1

Introduction

At its best, design is an activity that effectively balances feasible creativity with innovative problem solving. When analysing a problem and evaluating creative solutions, designers rely on the knowledge derived from both their own experiences and the experiences of fellow designers and researchers in their area(s). This latter knowledge source is often expressed in design guideline documents, which essentially pertain to assessments of past design successes and failures, from which notable phenomena are codified. By accumulating a body of knowledge pertaining to their subject matter, a community of practitioners can help steer the development of their craft.

However, emerging domains of design practice often do not have the full benefit of accumulated guidelines\(^1\). Indeed, a community of knowledge sharing and corresponding shared language with which to discuss the problem domain are immature at this stage. New findings by

\(^1\)This will be exemplified in the following section.
schrödinger scholars in the area are therefore relatively more difficult to discuss, disseminate and synthesize in a manner useful to practitioners operating in the domain. By communicating design insights with one another, researchers and practitioners working in a growing domain can share their early findings and help to reduce the repetition of design failures and to build on past successes, and ideally encourage the development of novel solutions based on a firm, shared understanding of the problem domain.

There have been many research projects and popular efforts to conceptualize, codify, retrieve, represent, situate and evaluate a domain’s design knowledge (several of which are addressed in Chapter 2 of this work. Essentially, these frameworks provide a means of support to assessing the relationship(s) between design artefacts and their application environment. If presented in an easily accessible form, initiatives such as these have the potential to encourage the development of a community of design practice, centered around a shared knowledge resource focussed on designing for an application domain.

These efforts, however, are necessarily coupled with their application domains; methods of structuring and analysing design knowledge should be tailored towards the design knowledge utilized in the area. This includes the types of problems generally addressed, degree of stakeholder involvement, typical work practices and processes, maturity of the application domain, and more.

This thesis is an attempt at developing a framework for structuring and analysing a domain’s design knowledge. It is conceptually based on goal-oriented requirements engineering principles, specifically the notion that design can benefit from a systematic analysis of the relationship
between design artefacts and stakeholder objectives. We pair this perspective with a focus on design knowledge sharing and recontextualization.

The work’s resultant GO-DKL framework GO-DKL framework is specifically focussed on the domain of online deliberation, an introduction to which we now turn.

1.1 The importance of designing for the public sphere

The “public sphere” is an idea popularized by German philosopher Jürgen Habermas (1998). Contrasting with the “private sphere” of one’s home, the public sphere is meant to be a place where citizens may freely discuss societal issues without interference from dominant groups. These discussions are to be rational, reflective and respectful; people should be open minded about other viewpoints, yet critical of all perspectives – including their own (Janssen and Kies, 2005). Such a reasoned process has been thought to encourage clear, logical and consensual decisions about societal issues.

However, some critics argue that this idealized format privileges a certain kind of rationality that can result in the marginalization of alternative viewpoints less expressible in dominant terms. By focusing on logical debate alone, more tacit communicative practices such as storytelling are disadvantaged, and in the process valuable knowledge could be omitted from consideration. Similarly, the focus on consensus has been charged as artificial; ‘true’ democracy arguably embraces diverse opinions and confrontations about key issues (Dahlberg, 2007).
As the web exploded in the nineteen-nineties, some cyber-utopians claimed that this newly popular communications channel, with its decentralized structure and marginal publishing costs, promised a revitalization of the “public sphere”. Where previously issues such as gender, geographic location, age and other characteristics might have stood in the way of equal participation, the anonymity of the ‘net purportedly erased these differences. People could focus on their opinions and share perspectives, information and knowledge. A new mode of democratic engagement and collective decision-making is the goal that many designers, developers and researchers have been pursuing. Governments and civic groups have sponsored many initiatives including, but not limited to systems premised on “e-participation”, “online consultation”, and “argument mapping”.

In spite of the quantity and scale of many efforts, on-line platforms for such practices have not generated the paradigm-altering results that had been imagined. Myriad social, psychological and technical issues have emerged in these arenas. The design of these forums for debate and dialogue have emerged as highly important issues, as subtle changes in the configuration of a system can have large impacts on the practices it supports (Wright and Street, 2007). For example, how can a designer democratically balance providing special content editing privileges to a class of moderator users while protecting the free speech of others? Wright (2009) found this to be challenging in his case study. Another concern has to do with producing actionable results through dialogue; how can a group of users best collaborate together? Which tools and tool configurations can help them?

While these cyber-utopian imagined futures are proving currently unfeasible, by prioritizing
good design and knowledge sharing in this area, a community of researchers and practitioners can hopefully improve these systems for democratic participation and help empower citizens to have a larger voice in how their communities are governed, how “wicked problems” might be tackled, or how new issues can be raised. Research is making headway into many of these areas, alternative system designs are being proposed and evaluated experimentally or in case studies. An annual conference in “online deliberation” showcases many of these endeavors (Davies and Peña Gangadharan, 2009)

1.2 Problem Statement

Current e-democracy practitioners recognize the importance of design for e-participation systems, yet often have difficulty in making informed design decisions. Some best practices have emerged, and various participation process guideline repositories and reference websites are active\(^2\). While the problems online deliberation websites face are fairly well understood, design solutions are not clearly defined. For example, a survey conducted and reported by CIVICUS\(^3\) found that many online participatory governance (PG) practitioners found that a “Lack of knowledge, skills and tools” about PG initiatives was the second most frequently mentioned obstacle or constraint faced by them (CIVICUS, 2007)\(^4\).

This aligns with a concern raised by Stephen Clift, executive director of E-Democracy.org\(^5\)

---

\(^2\) Several are discussed in this section
\(^3\) A prominent international group focusing on citizen participation
\(^4\) Lack of financial resources was the top obstacle
\(^5\) An online community whose mission is to harness online tools to strengthen public life
who asked “how do we invest in online approaches, tools, and ideas that bring together millions
of people with differing views to solve common problems or at least learn from each other”? He
went on to note that E-Democracy.org’s current approach to identity management in an online
deliberation environment, a “mix of local with real names and strong civility [...] is only a wee
part of the answer, what else is?” (Clift, 2011). Here, Clift called for alternative design strategies
to help promote healthy dialogue and deliberation – yet he also hints at a relative lack of
knowledge regarding how to design a successful e-participation tool. Similarly, Davies notes that
online deliberation researchers face difficulty in synthesizing and building upon the wide range of
findings of other scholars (2009). Davies and Peña Gangadharan have attended to this issue by
editing a compilation of conference proceedings, Online Deliberation: Design, Research, and
Practice (2009). Their work presents a wide range of empirical and theoretical analyses, yet
recommendations for designers of online deliberation systems generally remain unsynthesized
and scattered across individual studies and contexts.

These concerns persist in spite of several publicly accessible catalogues of e-participation tool
descriptions and case studies of their application. An example of such a catalogue that is currently
available and continually under development is “ParticipateDB”\(^6\), which bills itself as an attempt
to “build a comprehensive guide to the many online tools for public participation and related
forms of citizen engagement” (ParticipateDB, 2011). This is similar to “Participedia”\(^7\), which
also presents a directory of participation methods in the form of a wiki. Visitors to these websites
may browse through lists of engagement tools, but to find suitable tools is a laborious task; when

\(^7\)See http://participedia.com.
potential solutions are only presented by their names, the visitor is forced to look through many potentially irrelevant tools prior to finding one that might be useful.

A more sophisticated approach is taken by “People & Participation”\(^8\); it features an advanced search method that asks users to describe their relevant method in terms of expected outcome, costs, number of participants, participant types, time requirements, and level of participation. Following such a search, the visitor is presented with a list of suitable tools. However, the fit between the user’s criteria and the list of tools is not made explicit, and so it is difficult to assess the relevancy of each tool to a particular search criterion. In other words, these above websites do not provide a systematic means of comparing the different tools, or the specific features that the tool is comprised of. That being said, while there is room for improvement, interviews with online deliberation practitioners conducted in this thesis found that the above are among the few websites currently attempting to help designers make ‘tool selection’ decisions in this domain.

It appears as though a major problem in this domain has to do with a lack of awareness of, and an inability to satisfactorily retrieve relevant design knowledge. This issue hampers the progress of online deliberation systems, as if they had effective tool support, practitioners could likely better learn from each other’s successes and failures, as well as capitalize on recent research in related fields (Davies, 2009). Providing a means of synthesizing this domain’s knowledge and a platform for the effective retrieval of this knowledge could therefore help mitigate this issue.

\(^8\)See http://www.peopleandparticipation.net/.
1.3 Purpose Statement

This research focuses on the development of the GO-DKL framework, which aims to be a means of addressing the online deliberation field’s problem of design knowledge retrieval, synthesis and analysis, as outlined above. This framework is intended to accomplish the following objectives:

- Support the systematic codification and analysis of online deliberation system design knowledge.\(^9\)

- Support the creation of a conceptually organized knowledge base of reputable claims that link design features to stakeholder objectives.

- Support a process for practitioners to retrieve and assess the relevance of design features to their own online deliberation system design projects.

- Help designers to systematically identify criteria for their choice of an off-the-shelf tool, or serve as an early-stage conceptual springboard that would inform more concrete, contextually situated design efforts.

- Explore the role of the GO-DKL framework in a system designer’s overall practice.

In addition, we intend to accomplish the following objectives:

- Iteratively explore various conceptualizations of the design domain and assess these in terms of their suitability to the wider framework and system designer practice.

\(^9\)Which we will define in Chapter 2, the literature review
• Iteratively explore and refine means towards contextualizing the knowledge base items within a design project.

• Explore and assess different techniques for representing the knowledge base contents based on interviews and exploratory case studies.

• Finally, explore the suitability and management of scale regarding goal models that are automatically generated from the knowledge base items.

1.4 Research Questions

To achieve the goals set out above, this study is guided by the following research questions:

1. How can insights presented in narrative text form be succinctly extracted and represented formally in a database structure that supports design activity?

2. What kind of retrieval mechanism would be appropriate for such a knowledge base? How do designers relate these knowledge base items to their own contexts? How can designers assess the credibility or reliability of this knowledge?

3. How can goal models be used to support online deliberation system designer analysis of the retrieved knowledge? When would this be appropriate, and what other representational schemes could be helpful?

4. How do online deliberation systems designers currently leverage past design experience
and published guidelines or cases studies in their design practice? How would the GO-DKL framework help support their design practice?

1.5 Chapter summaries

Chapter 2, the literature review, defines and introduces current issues in online deliberation and outlines some design choices that are often considered by practitioners. The review then defines “design knowledge” and “design knowledge reuse” and outlines current approaches to codifying, cataloguing, retrieving, contextualizing and evaluating past design knowledge. The chapter concludes with a brief overview of goal modeling approaches to design knowledge representation.

Chapter 3 presents the methodology used in this study, which is a mix of an iterative design science approach, qualitative document coding and semi-structured interviews.

Chapter 4 discusses the iterative development of the framework. It begins with a discussion of early work preceding the project, and documents the problems and solutions encountered and developed.

Chapter 5 describes the framework in detail. The codification method, analysis method, and the various iterations we went through are explained through an account of a scenario based on real issues involving a climate change debate website, the MIT Climate Collaboratorium.  

Chapter 6 presents a complete, end-to-end application of the GO-DKL framework based on

---

10 Now called the Climate Colab, and found at http://climatecolab.org
another exploratory case study.

Chapter 7, the discussion, describes the results of practitioner interviews and places the framework in the wider context of online deliberation design practice.

Chapter 8 summarizes the findings of the research, the framework’s strengths, its limitations, possibilities for future work and presents concluding remarks regarding the significance of the study.
2

Literature Review

This chapter reviews three distinct areas of scholarly literature in order to contextualize the study presented in later chapters. The first to be discussed is the emerging study of online deliberation, followed by an analysis of several design knowledge reuse projects, and concluding with an overview of several goal-oriented requirements engineering techniques. The latter topic has seen several studies develop representational frameworks for a domain’s design knowledge, work that this thesis continues.

2.1 Online Deliberation

Online deliberation (OD) is a practice of “reasoned, purposeful and interactive” (Davies, 2009), electronically-mediated interpersonal communication. The first characteristic ascribed to OD by Davies is based on the ideal of rational-critical debate. This can denote turn-based, evidence- and
logic-based argumentation and respectful criticism of ideas (Janssen and Kies, 2005). The second characteristic, purposeful, denotes the idea that OD is often situated in some kind of decision-making context where a deliverable, concise summary of the discussion is the major goal among deliberators; For example, this deliverable might be a consensus or agreement statement about a social issue. The last of Davies’ characteristics is ‘interactivity’, which emphasizes the fact that deliberation is based upon interpersonal exchange, and ideally, a process of listening, speaking, critically building on each other’s ideas towards a consensus or recognition of diverse perspectives.

There are numerous problems that may emerge when engaging in online deliberation or other group decision making processes. For example, Sunstein’s *Infotopia* (2006) explains how group deliberation can result in skewed decisions based on informational and social pressures, such as valuable information being left unshared, or a fear of countering the dominant viewpoints, which can often lead to the polarization of stated opinions.

Critics of the rational-critical process argue that a logic-based, rational practice of expression marginalizes certain groups based on a homogeneous definition of rationality which ignores cultural variability in rhetoric and discussion protocol (Dahlberg, 2007; Papacharissi, 2004). Another argument posits that consensus-building is effectively undemocratic, as it negates plurality and difference of opinion (Moe, 2008). Thus, various design alternatives can be developed and selected in pursuit of the varying goals within this realm.

The question of “how can we best leverage collective opinions to help make social/political decisions” is treated in this thesis as a highly context-dependent issue; the different practices of
various stakeholders can shift requirements for a successful consultation process in variable manners. Indeed no single process is guaranteed to align with the practices, goals and constraints of different contexts. A situated understanding of the pros and cons of certain deliberative configurations is more appropriate.

2.1.1 Design Issues

This section serves two purposes; first, it outlines some of the many concerns that online deliberation systems designers address. Second, it provides a glimpse into the sort of knowledge that has been codified and stored in the framework’s knowledge base. Indeed, the following paragraphs are organized based on several design issues that the GO-DKL framework uses to guide designer retrieval and selection of relevant concepts from the knowledge base. These issues have been inductively identified through the document coding process described in chapter 3. Finally, the following descriptions should help the reader better understand the application example presented in Chapter 6.

A major defining characteristic of online deliberation (particularly conducted in political contexts) is a focus on normative ideals relating to the process of communication and decision-making that comprise the deliberation. These ideals may focus on fairness, trustworthiness, the protection of minority opinions, a focus on consensus building, sincerity, respectful interaction, and more. One example of a tradeoff that can emerge with respect to these concerns is the negative effect that prioritization of content in pursuit of consensus has on the diversity of views (Lev-On and Hardin, 2008). Furthermore, a focus on conflict reduction might
damage the sincerity of any perceived agreement among participants (Afshar et al., 2009).

Balancing and prioritizing these concerns is an important step in the design of a deliberative process and its underlying tool support.

The above normative concerns are intertwined with the desired outcomes relating to the representation, organization, and navigation of the discussions themselves. Certain designs better support desired representational outcomes. For example, storytelling and tacit interaction is better accommodated by a design prioritizing narrative-based textual conversation, while reasoned criticisms of proposals may benefit from highly-structured argument mapping techniques (Introne, 2009).

In certain deliberative processes, community building is not a desired outcome. This is especially true in processes that endeavor to be focused explicitly on decision making. Nevertheless, platforms for interaction can variably encourage community development, and depending on its desirability, can be configured appropriately with respect to this concern. For instance, granting special moderator privileges to some users might help regulate discussion and keep it rational and critical, but at the same time, this power helps to disseminate and create community norms of acceptable behaviour (Gurzick and Lutters, 2009). If knowledge sharing is a goal of the deliberation, then supporting users in identifying valuable interaction partners or valuable users might help this goal (Chen and Hung, 2002), but would also help users find like-minded others with whom a bond could be forged (Gurzick and Lutters, 2009).

An online deliberation activity may be carried out in order to help shape policy, inform decision-makers about citizen priorities, or enable citizens to come to an agreement, among other
goals. There are many concerns as to how the provided content can be structured or summarized in order for it to function as a reliable indicator of collective opinion. For example, the traditional threaded conversation of a discussion forum is correlated with a lack of consensus-building among users (Dunne, 2009).

The degree to which individuals and groups can develop and maintain identities within a system is also an area of design concern. For instance, Mathieson (2007b) found that incorporating team-oriented icons into a design helped remind users that they were members of a specific group, and in doing so, group identity was strengthened. Additionally, visible indicators of user status in a community can help shape the opportunities for socialization of users with varying levels of status (Nygren, 2010).

All of these concerns matter little if users do not participate in the discussions. There are various facets to this concern, including attracting new users (Lampe et al., 2010), increasing individual contributions (Andrews, 2002), and sustaining and growing an existing user base (Jones et al., 2002). Another key goal is to convert users from ‘lurkers’\(^1\) to full contributors (Gurzick and Lutters, 2009).

Given the variety of design issues at play in this domain, a systematic approach to relating design knowledge around such topics may help to address the field’s knowledge sharing problem identified by Davies (2009) \(^2\).

---

\(^1\)Users who do not provide any content to the web site, but consume and learn from the contributions of others

\(^2\)As outlined in Chapter 1
2.2 Design knowledge

Given the need for online deliberation designers to better take advantage of the knowledge produced by researchers, a means of rendering this information useful to designers is required. Design knowledge has been argued to only be useful to designers if they: choose to access it, are able to access it, and can apply it readily (Lera, 1984). They must also be aware of this knowledge in order for it to be useful. Additionally, novice designers have been found to require support in identifying what they need to know; a problem that a robust repository could help mitigate (Ahmed and Wallace, 2004).

However, a limitation of a repository based on designer input is that the documentation of design decisions is often sporadic and unreliable (Ormerod et al., 1999), and often the creators of design rationale are not those who would directly benefit from it (Grudin, 1988). Even when utilizing design rationale systems, it has been found that designers may often prefer to simply recall previous design work from memory, rather than querying a system (Ormerod et al., 1999). One risk of this is that potential alternative configurations may not be considered. Therefore, there is a need for systematic support to encourage such considerations (Ormerod et al., 1999).

Modeling domain knowledge is one means of providing information that supports designers: Requirement engineering, “particularly in the early stages, involves the handling of uncertain and mutable problem models. Previously developed RE models can be used as tools for managing and reducing uncertainty” (Lam et al., 1998). Many applications in a single domain share a set of common requirements that can be termed ‘domain requirements’. (Lam et al., 1998). For this
A large question remains: “how can we effectively conceptualize, codify, retrieve, represent, situate and evaluate online deliberation design knowledge?” The domain’s design knowledge will be more useful to designers if a satisfactory answer to this question is found. There are many ways of conceiving design knowledge and its reuse, several of which are outlined in the following section.

2.3 Approaches to design knowledge reuse

This section identifies and compares various approaches, from various disciplines, to the representation, management and analytic support focused on design knowledge and its reuse.\(^3\) These approaches usually have central conceptual elements, a means of codification, retrieval, representation, contextualization and evaluation. See the table at the end of this section for a concise summary of our discussion. Elements from these various approaches inform the design of this thesis’ framework, which is described in Chapter 4.

2.3.1 Conceptualizing and codifying design knowledge

In order to systematically build upon past design experiences, a clear model of conceptually important elements pertaining to design processes and artifacts must be defined. Additionally, a

\(^3\)This thesis treats ‘reuse’ flexibly; it stands for ‘the recontextualization and repurposing of design knowledge’. 
means of extracting these concepts and explicitly codifying them is generally required. These elements provide design knowledge with a structure that supports effective retrieval. Several such approaches are described below.

Conceptualization

Design knowledge can be expressed in various forms. The knowledge domain, desired analysis support, knowledge sources, and degree of context-sensitivity all play a part in shaping a useful conceptual model of the design knowledge in question.

A method for codifying design knowledge in the industrial design field outlined in Muller (1996) centres its conceptual focus on the design product ‘form’. Forms are typified according to three feature categories: functional, solution, and behavioural features. Respectively, these relate to a form’s intended capabilities, relationship with its implementation environment, and its symbolic connotations. Another approach derived from interviews with engineers found the participants’ sought-after design knowledge included other elements besides the ‘form’; it categorized the aerospace domain’s design concerns into four concepts: the process itself, the product (or form), the function, and issues (relevant concerns) (Ahmed, 2005).

Design knowledge in architecture has been expressed in ‘precedents’ - design cases divided into conceptual stories, which in turn are made up of an issue, concept, and form (Oxman and Planning, 1994). Note that here, ‘form’ is but one element in this conceptual framework, unlike its central role in the above example, this method is more contextually-situated; it deals with stories set in a certain place and time, and also prioritizes design problems (‘issues’) and
high-level solutions (‘concepts’). This is likely due to architecture focusing on designing in context, but industrial design focusing on the product itself. A recent study in architecture similarly focuses on precedents, but provides seven different models for conceptualizing a precedent based on the contextual factors as well as the design method chosen (Eilouti, 2009).

Similar to the ‘precedent’ concept is a design ‘episode’, which Ormerod et al. (1999) conceives of as a “focus constellation” of factors: components, design activity, rationale, and indexes. These are contextually-related through their focus on a single design question. A stronger focus on the actual activities of the design (lifecycle stage, associated stakeholders, etc.) as well as the rationale underlying the design decisions made, differentiates this approach from the others presented here.

A framework for design reuse in the Human-Computer Interaction discipline (Wahid et al., 2006), divides knowledge into ‘claims’. These conceptual elements are described as items which outline the positive and negative tradeoffs associated with a particular design feature within a usage context.

The requirements engineering discipline presents other conceptualizations of design knowledge. For example, a method of structuring agile software development method design knowledge focuses its conceptual model on ‘method fragments’, which satisfy ‘objectives’ and rely on ‘requisites’ (Esfahani and Yu, 2010). Another approach is “requirements recycling”, centered on reuse of clearly-defined use cases, which are treated as the conceptual ‘parent’ element of codified system requirements (Alexander and Kiedaisch, 2002). This approach, like many requirements reuse methods, focuses on clearly-defined system requirements within an
organization. However, similar to the agile method fragments study (Esfahani and Yu, 2010), the framework developed in this thesis intends to draw upon knowledge at the generic disciplinary level, and furthermore deals with the fuzzier textual expressions of journal articles and book chapters as sources to be codified in a knowledge base.

**Codification**

Codification schemes are dependent upon the domain’s conceptualization and the sources of design knowledge. The following is a brief overview of how the above approaches deal with design knowledge codification.

Muller (1996)’s the context-independent approach outlined above focuses on the typification of (industrial design) products – their abstraction into instances of the three above-mentioned feature types. This typification is argued to enable designers to look at potential solutions from multiple perspectives.

Past knowledge from design cases may framed as precedents that are be decomposed into conceptually related stories that textually and descriptively outline the issues, concepts, and form (Oxman and Planning, 1994). A proposition from the architectural domain includes an analysis-synthesis cycle of precedent-based knowledge recycling (Eilouti, 2009). In this model, precedents are created and codified at the end of this cycle, through a refinement and clarification process of lessons learned from buildings and testing a design prototype.

Another, more context-sensitive method, is through the indexing of design reports. The actual
report documents are indexed as instances of certain domain knowledge categories (Ahmed, 2005). A library of these tagged documents may then be related to one another based on their shared indices.

Ormerod et al. (1999) similarly situate their codification approach in-context. However, it differs in that design knowledge is meant to be codified seamlessly as the work is being done, rather than after the fact. This, coupled with their situated retrieval approach, is meant to make the codification process appear more valuable, as past results are presented contemporaneously with encoding activities. According to this method, as a new situation’s relevant concepts become apparent during the design process, they are to be encoded directly and immediately into the knowledge base.

The previously-mentioned HCI method (Wahid et al., 2006) proposes that designers isolate clear design features and codify their effects as claims. These claims should furthermore be related to others based on their effects.

The codification process in the repository of agile method fragments (Esfahani and Yu, 2010) consists of explicitly relating a method fragment name to its requisites and objectives, as well as storing details about the examined study (such as context and study type). After aggregation of similar elements, these relationships are then stored in tables called ‘generic snapshots’, which depict the interrelationships among various fragments, objectives and requisites. Finally, the use-case-based approach (Alexander and Kiedaisch, 2002) proposes an ‘archaeological’ method involving the extraction of aggregate use cases from groups of explicitly stated system requirements that an organization has previously generated.
2.3.2 Retrieval and Application

In order to be useful to designers, structured design knowledge requires a means of retrieval. The sources mentioned above also present retrieval and representational methods which are discussed below.

Retrieval

Search strategies have been utilized by a number of sources, most basically outlined in Eilouti (2009) and Ahmed (2005). A more robust means of doing so is having a user define a design ‘issue’, ‘concept’ or ‘form’ using a controlled vocabulary, and subsequently browsing the ‘stories’ that address the search term (Oxman and Planning, 1994). A similar approach involves a search for specific attributes of a design feature within a feature typology (Muller, 1996).

However, another line of thought argues that the knowledge repositories on which many design reuse schemes are based are limited by their search-orientation, and the centrality of the reposited knowledge over the user’s context (Wahid et al., 2006). Their alternative proposition is a context-sensitive browsing approach that centres on a user manually connecting various design features from the repository together to form a larger design suited to a specific project. They have created a digital “claims library”, navigable via different types of relationships among claims.

Another browsing approach mentioned in the agile method fragment repository provides valuable flexibility in enabling users to browse the knowledge base through focusing on three major meta concepts (objects, requisites and fragments) (Esfahani and Yu, 2010). Alternatively,
one may simply browse of a list of use cases and their related system requirements (Alexander and Kiedaisch, 2002).

A novel approach by Ormerod et al. (1999) continues with their proposed seamless design reuse framework. As designers are defining a new ‘design episode’ by entering metadata and defining focal questions, options and criteria, the reuse system continually parses this information and automatically presents weighted, relevant past episodes for the designers to examine.

**Representation**

While GO-DKL uses $i^*$ as one representational framework\(^4\), a consideration of alternative approaches to the representation of design knowledge has helped in the development of the GO-DKL browser, the prototype implementation that supports the analytic processes described in chapter 5.

Two approaches represent retrieved design knowledge as *illustrations* of the forms involved. One is simply a browsable list of relevant thumbnail images (Muller, 1996), while a second approach (Oxman and Planning, 1994) focuses on ‘cards’; clickable panels that can emerge from these illustrations. A ‘card’ is available for each ‘story’ related to the precedent, and contains hypertextual links to the cards detailing the ‘concepts’ and ‘issues’ involved (which may link to other stories as well). This presents a network view that affords the exploration of relevant design knowledge.

---

\(^4\) As has the agile method fragments repository (Esfahani and Yu, 2010)
The HCI design claims framework leads to the creation of visual models called ‘claims maps’, which depict the interrelationships among textually-expressed claims (Wahid et al., 2006). This is argued to facilitate the analysis of the effects of including various components in a design. A final representational scheme focuses on use cases, which may be expanded into a hierarchy with system requirements and predecessor systems as child elements of the use cases (Alexander and Kiedaisch, 2002). In this manner, similar requirements are grouped and retrieved together.

**Contextualization**

Retrieved design knowledge must be reapplied in a new context in order to be effectively reused by designers. Below is an outline of several approaches to design knowledge contextualization that have been attempted.

The $i^*$ models that are used to represent agile method fragments are intended to be customized by the framework user, through a process of modifying the relationships between fragments, their objectives and requisites (Esfahani et al., 2010). Adding new requisites and objectives, and interrelating fragments with others is also described as a means of situating the retrieved knowledge. Wahid et al.’s ‘claims map’ can similarly be customized, with designers intended to assemble proto-designs by interconnecting retrieved claims into a larger claims map (2006). Similarly, retrieved use cases and system requirements may be integrated into prototypical requirements documents to suit the current situation (Alexander and Kiedaisch, 2002). Precedent-based design forms can also be informally adapted to suit a context Eilouti (2009), as can the other approaches.
Evaluation

Ormerod et al. (1999)’s evaluation procedure is implicit. Designers may retrieve past knowledge, and determine relevancy based on their own judgement. However the inclusion of previously retrieved knowledge into a new ‘design episode’, coupled with the ability to mark criteria as ‘important’, affects these criteria’s subsequent retrieval weightings. This procedure is implicitly question- and context-based, as related episode components are taken into account as the system encodes each episode. Several approaches involve the focused analysis of alternatives design solutions (Eilouti, 2009; Esfahani et al., 2010). Eilouti (2009) integrates this evaluation procedure into the proposed ‘analysis–synthesis cycle’ whereby evaluated prototypes go on to inform new design precedent reports. Several other approaches utilize informal approaches to assessing the degree to which the designed artifact fits with the application context (Muller, 1996, such as). Esfahani et al. (2010) uses the same evaluation procedure as GO-DKL, which is addressed in detail in the next section. The below table 2.1 summarizes the various approaches to the above topics.

2.3.3 Summary

The above approaches stem from a variety of disciplines and focus on diverse domains. Indeed, the approaches themselves are loosely coupled with their source disciplinea and application domaina. For example, Alexander and Kiedaisch’s approach (2002) is centered on cataloguing an organization’s documented use cases, an IT-specific function meant for a large engineering firm.
Table 2.1: Several approaches to design knowledge reuse

<table>
<thead>
<tr>
<th>Ref</th>
<th>Concept</th>
<th>Codify</th>
<th>Retrieve</th>
<th>Represent</th>
<th>Situate</th>
<th>Evaluate</th>
</tr>
</thead>
<tbody>
<tr>
<td>a</td>
<td>categories</td>
<td>index</td>
<td>search</td>
<td>indexed report</td>
<td>–</td>
<td>–</td>
</tr>
<tr>
<td>b</td>
<td>use case</td>
<td>reverse-engineer</td>
<td>browse</td>
<td>hierarchy</td>
<td>integrate</td>
<td>none defined</td>
</tr>
<tr>
<td>c</td>
<td>precedent</td>
<td>refinement</td>
<td>search</td>
<td>varying</td>
<td>adapt</td>
<td>alternatives</td>
</tr>
<tr>
<td>d</td>
<td>fragment</td>
<td>generic snapshots</td>
<td>browse</td>
<td>i*</td>
<td>customize</td>
<td>alternatives</td>
</tr>
<tr>
<td>e</td>
<td>form</td>
<td>typification</td>
<td>search</td>
<td>pictorial list</td>
<td>informal</td>
<td>analyze ‘fit’</td>
</tr>
<tr>
<td>f</td>
<td>precedent</td>
<td>decomposition</td>
<td>search</td>
<td>cards</td>
<td>–</td>
<td>–</td>
</tr>
<tr>
<td>g</td>
<td>episode</td>
<td>in situ</td>
<td>auto</td>
<td>text</td>
<td>implicit</td>
<td>implicit</td>
</tr>
<tr>
<td>h</td>
<td>claim</td>
<td>feature effects</td>
<td>browse</td>
<td>map</td>
<td>map</td>
<td>informal</td>
</tr>
</tbody>
</table>

Additionally, the visual approaches taken by Oxman and Planning (1994) and Muller (1996) reflect the industrial design and architectural disciplines’ focus on the built, tangible artifact. As such, during the development of the GO-DKL framework, we endeavoured to be aware of these existing knowledge structuring frameworks, building on them where appropriate, and searching out other means of addressing the above issues in a manner suitable for online deliberation systems design.

2.4 Goal-oriented Requirements Engineering

The failure of information systems is often attributed to a misalignment between technical capabilities and environmental context. Simply stated, the technical functions performed do not sufficiently help to achieve the goals or meet the needs of stakeholders. This is one of the
motivations of goal-oriented software engineering. This methodology focuses on the systematic
development of information systems to ensure the achievement of organizational goals.
Furthermore, this approach has been argued to facilitate a clear focus on the problem, rather than
on one particular solution (Wieringa, 2004). System design alternatives are comparatively
analyzed based on their varying contributions to stakeholder goals. This form of tradeoff analysis
forms the basis of system design choices, and can be referred to later on in the system life cycle as
a traceable guide to design rationale \(^5\). Similarly, as organizations are evolving, dynamic entities,
so too are their goals, and so too must the technical solutions be reassessed in light of these
changes.

A common method to represent the relationships among stakeholder goals and technical
capabilities is through goal modeling. This has been recognized as an effective means of
representing organizational-level requirements based on stakeholder needs (Chung et al., 2000;
Dardenne et al., 1993). Goal models often take the form of a hierarchical tree-like graph. Abstract
‘high-level’ goals are decomposed into more specific goals, indicating that these specific goals
together can help to achieve the high-level goal parent. These may alternatively be represented as
binary alternatives; either one or the other must be chosen. Further decomposition of the goals
can link them to operational, technical capabilities. These operationalizations may contribute to
one or many goals in varying ways, forming the basis for tradeoffs analysis.

2.4.1 The NFR Framework

The conceptual model underlying the GO-DKL framework is based on the NFR framework
(Chung et al., 2000), which is leveraged as a means for representing domain design knowledge as
interdependent entities. The NFR framework deals with non-functional requirements (NFRs).
These requirements are conceptualized as qualities that an information system is aspired to

\(^5\)The reasons underlying the decisions made during the design process
feature, such as performance, security, or privacy. In essence, they are goals that cannot be measurably achieved. Thus they are treated as ‘soft goals’ that, following Simon (1996) can only be sufficiently satisfied – ‘satisficed’.

Instead of viewing these nonfunctional requirements as results of the design and development of an information system, the NFR framework proposes that NFRs systematically drive the design process; design choices are based on their perceived impact upon NFRs. Particular kinds of NFRs, in conjunction with domain and technical knowledge may be systematically modeled and analyzed in order to deal with ambiguity, identify tradeoffs and priorities, interdependencies among elements, as well as to guide the actual selection of design alternatives. Following the selection of design alternatives, an analyst may evaluate the effects of a decision alternative on the higher-level goals of the project and potentially reassess certain selected alternatives in light of those findings. These tasks may be performed iteratively.

In order to accelerate the analysis, catalogues of design knowledge may be developed for future reference. For example, the NFR framework proposes catalogues for each NFR type (eg: usability). These include the concepts and terminology associated with the type. Other proposed catalogues include method catalogues and correlation catalogues; the latter recording tradeoffs among various soft goals. Developing new catalogues of domain knowledge has been the subject of several related theses developed at the University of Toronto. For example, Kabir (2008) concisely reorganized and represented web accessibility guidelines using the NFR framework. Similarly, Moayerzadeh (2008) represented service-oriented architecture (SOA) design guidelines using the framework.

Softgoal interdependency graphs are the NFR framework’s main representational vehicle for modeling design knowledge. Essentially, these are goal graphs like those described above. However, they are centered on softgoals and their interrelationships. Softgoals are represented as nodes in the graph, connected via links. These contribution links can either refine a soft goal into
Table 2.2: Contribution links used in the NFR framework

<table>
<thead>
<tr>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AND (OR)</td>
<td>Parent node is satisficed if all (one) of the children is satisficed</td>
</tr>
<tr>
<td>Makes (breaks)</td>
<td>Parent node is satisfied (denied) if child node is satisficed</td>
</tr>
<tr>
<td>Helps (hurts)</td>
<td>Parent node is satisfiable (deniable) if child node is satisficed</td>
</tr>
<tr>
<td>Unknown</td>
<td>No value propagation; useful for preliminary relations</td>
</tr>
</tbody>
</table>

more concrete or operational ones, or relate soft goals with one another. Table 2.2 presents the NFR frameworks various contributions links in more detail.

The below Figure 2.1 exemplifies the concepts discussed above. It is a hierarchical graph beginning with three abstract goals at the top, and decomposing into refined goals or contributing elements (including goals and operationalizations). Design alternatives (the operationalizations) are located at the bottom of the graph and contribute to a variety of soft goals (ie: they can have more than one parent node). The checkmarks indicate that this graph has been subjected to the evaluation procedure described next.

As mentioned above, evaluating these graphs of interconnected design concerns can help designers to make choices that respect organizational goals and needs. The qualitative and interactive evaluation procedure outlined in Horkoff and Yu (2009) begins with the selection of candidate design alternatives (operational, implementable technology solutions). On the graph, this selection would be indicated by a checkmark - labeling the design alternative’s node as ‘satisfied’. Once a sufficient number of design alternatives have been selected in this way, the analyst would assess how the selected nodes contribute to their parent nodes (via contribution links).

Based on the type of contribution link, the parent node would inherit an ‘evaluation label’ from its child. Consider an example from the online deliberation domain, if the design alternative Anonymous messages is selected, and it contributes via a helps link to Reduce social
Figure 2.1: Example NFR softgoal interdependency graph. Based on Chung et al. (2000).

pressures, the latter node would be marked as ‘partially satisfied’. However, if Anonymous messages also contributes via a hurts link to Flame wars be avoided, then the latter node in this case would be marked as ‘partially denied’. See Figure 2.2 below for a comprehensive outline of the evaluation label propagation rules.
2.4.2 Agent-oriented modeling and the *i* framework

This section briefly outlines the premise and syntax of *i* (Yu, 2009), a requirements engineering modeling approach related to the NFR framework; *i* similarly focuses on soft goals and their contributions from design features. However, *i* is not limited to operationalized design features and soft goals, its unique syntax includes both ‘hard’ and ‘soft’ goals, tasks (similar to operationalizations) and resources. Hard goals are clearly-defined objectives that can be measurably achieved, while soft goals are fuzzier and have to do with quality attributes. Tasks are processes undertaken by actors, and resources are items that can be used by tasks or delivered in order to satisfy goals.

These elements are related through links similar to the NFR framework, but *i* is guided by a different focus; it is conceptually focused on interdependent actors. Each actor has internal goals, performs certain tasks and may create or provide resources. Actors depend on one another for these resources to be provided, tasks to be carried out, and goals to be achieved. Essentially, one actor can affect the satisfaction of another actor’s goals. By assessing alternative tasks (or system design features) and their varying impacts on actor goals and those of their dependents, an analyst may perceive the social effects of different business processes, technical solutions or dependency relationships. In this manner, *i* is actor, or agent-oriented in addition to being goal-oriented,
while the NFR framework is only goal-oriented. As $i^*$ has more mature tool support than the NFR framework, the $i^*$ syntax was employed for the representation of the NFR goal graphs\(^6\) developed during this thesis work.

Figure 2.3: Basic $i^*$ syntax

A goal-oriented conceptual model of design knowledge – coupled with complementary retrieval and representational tools – may be an effective means toward a system designer’s systematic consideration of the social possibilities afforded by design features. While the agile method fragment design knowledge repository discussed in the preceding section (Esfahani et al., 2010) presented a goal-oriented conceptual model and represented repository items using $i^*$, their retrieval method is not clearly described nor has it been subject to any empirical study. Previous theses focusing on the goal-oriented codification, representation and analysis of both design patterns (Moayerzadeh, 2008) and accessibility guidelines (Kabir, 2008) do not define a systematic means of cataloguing, retrieving and contextualizing the information. As such, this thesis takes the opportunity to explore, refine and empirically assess an interface to a goal-oriented design knowledge repository, which this document next presents.

---

\(^6\)This was relatively straightforward, given the similarities between $i^*$ and NFR.
Research Methodology

This chapter discusses the various methods employed in this research. The framework presented in later sections of this thesis emerged from this theoretical base. The overarching approach taken for this thesis was based on Hevner et al. (2004). The work of other scholars guided the approaches to document codification and interview transcription and analysis.

3.1 Design Science Research in Information Systems

The design and development of the GO-DKL framework was guided by Hevner et al.’s (2004) model of information systems design research. That model called for the consideration of both knowledge from the environment (people, organizations, and technologies), and the Information Systems (IS) discipline’s knowledge base (theoretical foundations and methodologies). This knowledge then informed the creation of an artifact to be evaluated. By following these guidelines, Hevner et al. (2004) posited that IS researchers can be positioned to conduct and report high-quality, relevant research.
By explicitly outlining the characteristics of robust information systems research, the core guidelines of Hevner et al.’s framework provide a good foundation for this research project’s methodology. These guidelines will now be presented alongside an explanation of how this research attempts to include them in the process.

### 3.1.1 Guideline 1: Design as an Artifact

The design artifact is positioned as the ‘core subject matter’ of the IS field, and must serve some purpose, for instance by addressing an important organizational problem. Hevner et al. describe a design artifact as “not only instantiations [...] but also the constructs, models, and methods applied to the development and use of informations systems” (2004, p. 82).
By this definition, this thesis research shall produce several interrelated design artifacts: An instantiation in the form of a prototype design knowledge retrieval system demonstrates feasibility, a model of online deliberation design knowledge represents the problem, and methods for populating and analyzing such a knowledge base both help to support design work.

3.1.2 Guideline 2: Problem Relevance

The design artifact must interact with the application environment in a relevant manner. Behavioural science literature relating to this environment should inform the creation of this artifact to help ensure relevancy. Additionally, a constituent community of stakeholders engaging in design work in the environment should be recognized and targeted in the research. Essentially, the design artifact should help serve their goals and needs. It should enable their problems to be effectively addressed.

The thesis engages with the known problem of the disparate treatment of online deliberation design knowledge, as identified by Davies (2009). The thesis attempts to address this problem by codifying the domain knowledge found in academic publications related to the target community of online deliberation system designers. From this, relevant stakeholder goals and technical capabilities will be included in a design artifact. Furthermore, interviews with practitioners in the field will help to elicit the degree to which the developed GO-DKL framework effectively addresses problems that they face over the course of their design work.

3.1.3 Guideline 3: Design Evaluation

In order for the findings of design science research to be convincing and reliable, the design activities should be subject to careful evaluation procedures. The application environment informs some of the criteria used in evaluation. For example, any new IT artifact must effectively integrate
into existing organizational processes and be accepted by stakeholders. The evaluation phase also provides feedback to inform the iterative design of artifacts, so that they may be incrementally improved. Evaluation methodologies are typically drawn from the IS knowledge base.

The evaluation of this project’s design artifacts will be less rigorous than those called for by Hevner et al.. For instance, whereas observational evaluation of an implemented artifact would call for a case study, the prototype in this thesis is not yet implemented in a usage context. Therefore, an informal demonstration to potential users followed by an interview where the users are asked to assess the artifact was carried out. To complement these assessments, a descriptive evaluation in the form of an application example was conducted.

3.1.4 Guideline 4: Research Contributions

A research contribution in design science must show that an artifact, when implemented, provides significant value to actors in the application environment. Alternatively, novel models and constructs that build upon past work in the knowledge base may be useful to researchers. If the artifact is implementable, models demonstrate representational fidelity, and a heretofore unsolved problem is addressed, then the design research is said to contribute to the field.

The new and interesting contributions arising from this work will include a prototype implementation (though significantly, not embedded in situ), as well as a model of domain knowledge which would demonstrate a degree of fidelity in an instantiation’s capacity to effectively support design work. Interviews shall help to determine the value provided to actors in the application environment.
3.1.5 Guideline 5: Research Rigor

Rigorous research is based on carefully selected and executed research methods. In the case of Hevner et al.’s guidelines, this pertains to both development and evaluation of the IT artifact.

The document coding process undertaken for this thesis carefully follows a qualitative social science coding best practices. Additionally, the most recent prototype of the GO-DKL browser described in this document has been subjected to a rigorous evaluation based on a demonstration to stakeholder and subsequent interviews.

These methodologies will be more thoroughly explained later in this chapter.

3.1.6 Guideline 6: Design as a search process

Hevner et al. describe design as a “search process to discover an effective solution to a problem” (2004, p. 88). Ideally, a complete model of the problem(s) and all potential solutions would guide the development of the artifact. However, many problems are fuzzy or wicked and thus developing a complete model is impractical. In these cases, ‘satisficing’ solutions (Simon, 1996) are desired. Using evaluation heuristics to assess iterative solution attempts is recommended.

The development of the GO-DKL browser prototype proceeded iteratively. The first two of the three major iterations were intuitively evaluated by the researcher and his supervisor based on the prototypes’ capacity to engage with the problems at hand. The third iteration was evaluated by practitioners who provided a great deal of feedback to guide future artifact iterations.

\footnote{Sufficiently satisfactory; “good enough”}
3.1.7 Guideline 7: Communication of Research

The final guideline presented by Hevner et al. suggests that research be presented with both technical and managerial audiences in mind. To accommodate the former, descriptions of any artifact should be detailed enough to ensure that they would be implementable by developers. To serve the latter group, the research communication should clarify how the artifact can be applied within a context in order to provide value to an organization.

Through the use of process and data models, as well as detailed screenshots and appendices outlining code and queries, this thesis paper clarifies how the analysis prototype could generally be implemented. By reporting on the results of practitioner interviews, managers may consider how others have assessed the prototype, and make a decision based on those findings.

3.2 Document Coding

The process by which scholarly publications dealing with online deliberation system design were coded and prepared for entry into a database followed several stages. A clear description of these process stages is essential for ensuring the transparency and reproducibility of the research (Wilson, 2009). First, general selection criteria were established and refined as publications were found. Second, a coding scheme was iteratively developed. Finally, a coding process was also iteratively developed alongside the scheme.
3.2.1 Selection Criteria

The selection criteria is a formal specification of the bounds within which documents are eligible for inclusion in the coding process. The criteria should emerge naturally from a study’s research questions (Wilson, 2009). The research questions driving the selection criteria used in this study are the following:

- How can insights presented in narrative text form be succinctly extracted and represented formally in a database structure that supports design activity?
- How can goal models be used to support online deliberation system designer analysis of the retrieved knowledge?

Three key factors that emerge from these questions are “insights that support design activity”, “goals”, and “online deliberation systems”. Due to these factors, the following eligibility criteria were developed:

- Publications shall mention design features.
- Publications shall mention stakeholder goals.
- Publications shall relate design features to stakeholder goals.
- Publications shall report on systems relating to online deliberation.

In order to obtain sources that might be useful to designers of online deliberation systems, many scholarly databases were queried. The goal of these queries was to obtain high-quality academic literature that detailed empirical studies of online deliberation systems.

Below is a list of the databases that were searched. The first two databases listed were queried in order to obtain a broad range of results. The remaining provided access to specialized information systems and computer science publications:

- Google Scholar
- Scholar’s Portal

2 And subsequently, the framework’s knowledge base
The following search queries were performed:\3:\n
- “online deliberation”
- “social media” design
- design deliberation
- e-democracy
- e-participation
- discussion forums
- democratic blogging
- online commenting

Unfortunately, as discovered by the researcher, and noted by a practitioner in one of this project’s interviews, there is a relative dearth of design-oriented e-democracy studies. Many online deliberation or e-democracy studies have been more focused on the resultant social interaction and political dynamics of the system, rather than the role of design. Due to this, many retrieved documents were excluded (per the eligibility criteria). To help offset the lack of highly specific literature, the search was expanded to related social media studies from HCI and CSCW fields (though the amount obtained was still smaller than hoped).

See Appendix A1 for a complete list of publications used in the knowledge base.

3.2.2 Coding Scheme

The coding scheme outlines what it is that the researcher identified in each document. Deciding on the items to include in the scheme requires a strong understanding of the domain’s literature\footnote{Or combinations thereof}.
A sense of what the common variables that appear in the literature can be obtained by coding a sample of preliminary documents (Brown et al., 2003). Following this method, the coding scheme is inductively defined as the landscape of concerns dealt with in the literature becomes clearer.

This project has followed that method; a coding scheme was iteratively refined over the course of three batches of documents. Another factor that played a role in the development was one of the project’s research questions: “how can goal models be used to support online deliberation system designer analysis of the retrieved knowledge?” Based on the desire to answer this question, the coding scheme was influenced by the representational constructs currently employed in goal models. Specifically included was the notion of design features contributing towards stakeholder goal achievement. As such, design decisions, contribution relationships and goals were included in the coding scheme.

Iterations of the scheme were informally evaluated based on their perceived utility in helping to answer the research questions. Section 4.3.1 outlines the development of this coding scheme in more detail.

### 3.2.3 Coding Process

The coding process generally remained consistent over the course of several iterations of the coding scheme. First, bibliographic details and a short description of the document are recorded. Next, passages in the document that appear to exemplify concepts from the coding scheme are highlighted and ‘tagged’ as instantiations of the concept. Once coding is complete, instantiated concepts are extracted and loaded into a database. Finally, similar concepts are merged together.

This process was carried out using QSR NVivo 8, a qualitative analysis program centered on documents and conceptual nodes. Passages in documents can be tagged as any number of
user-definable nodes. In this project, nodes included coding schema concepts as well as their instantiations. For example, a passage would be coded both at the meta concept level, Design Feature and subsequently coded as one instantiation of that meta concept: invite politician to participate. NVivo supports custom queries on all elements within a ‘project’. These queries supported exporting the data required to populate the knowledge base.

### 3.3 Iterative Design

Following Hevner et al.’s (2004) guidelines, the framework development proceeded iteratively in a design-evaluate cycle. This iterative approach applies to both the development of the framework’s underlying meta model, as well as the knowledge analysis method. For the latter, wireframes and basic web applications were rapidly developed and informally evaluated. Initial design work was data-driven, while later iterations focused more on the analysis process itself.

See the next chapter for a detailed explanation of the framework’s development.

### 3.4 Semi-structured Interviews

In order to evaluate version 3 of the GO-DKL browser, semi-structured interviews of e-democracy practitioners were conducted. The results from these sessions help shape an understanding of the framework’s utility, limitations, and directions for future work.
Table 3.1: Interview activities

<table>
<thead>
<tr>
<th>Activity</th>
<th>Duration</th>
<th>Total Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>Introductions</td>
<td>5 mins</td>
<td>0:05</td>
</tr>
<tr>
<td>Brief discussion of thesis framework; Q&amp;A</td>
<td>5 mins</td>
<td>0:10</td>
</tr>
<tr>
<td>Interview about participant design practice</td>
<td>20 mins</td>
<td>0:30</td>
</tr>
<tr>
<td>Demo of the prototype tool</td>
<td>10 mins</td>
<td>0:40</td>
</tr>
<tr>
<td>Interview about the tool and participant perception of it</td>
<td>15 mins</td>
<td>0:55</td>
</tr>
<tr>
<td>Wrap-up</td>
<td>5 mins</td>
<td>1:00</td>
</tr>
</tbody>
</table>

3.4.1 Participants

Participants were recruited using two methods. The primary population that was sent a recruitment notice were members of a mailing list of e-democracy designers. Potential participants were asked to describe a project that he/she was/is involved with that could be used as a test project during a demonstration of the framework.

The participants were required to be e-democracy practitioners. In this study, an e-democracy practitioner was defined as an individual meeting one of the following criteria: The participant was a member of the e-democracy mailing list OR the participant was referred to the researcher through a mailing list member. Participants were required to have been part of a project that designed systems for publicly accessible, electronically-mediated interpersonal discussion pertaining to social or political issues, a definition based on Davies (2009). The participants were also required to be located in the Greater Toronto Area or able to participate in a web-based Skype call. Six interviews were conducted, five of which were conducted through Skype.

Interviews generally lasted one hour and were structured as shown in Table 3.1.
3.4.2 Transcription and Analysis

Interviews were audio recorded and transcribed by the researcher within 48 hours of the interview. These transcriptions were then read over and general themes discussed by the participants were identified. A list of themes was produced for each transcription. These lists were then merged into a master list, including only recurring themes present in more than one list. From this list, a coding scheme was developed, following Weston et al. (2001) and Burnard (1991). This interview coding scheme was influenced a priori by the project’s focus on goal-oriented relationships, and therefore the notions of ‘goal’ and ‘design’ were included at the beginning of the process.

Two separate coding schemes were developed for the interview proper and GO-DKL prototype demonstration portions of the practitioner sessions. These schemes are respectively shown in tables 3.2 and 3.3.

Interview participants were asked to describe common tasks undertaken during the course of their design work, the context in which the designers worked as well as the application context, various problems, goals and design solutions engaged with, as well as their views on the role of design knowledge in their practice.

During and following the application demonstration, participants discussed their interpretation of the prototype, expressed critiques, made suggestions and reflected on the relevance of the framework and its elements to any potential use cases they could imagine.

Each transcript was coded according to this scheme. From there, instances of each code item identified in any of the transcripts were amalgamated and analyzed. The perspectives of each interview subject, relating to the same topic – the focal code item – were then able to be identified. These are discussed in detail in Chapter 7.

There are several limitations to the execution of this approach, derived from the best practices
3 Research Methodology

<table>
<thead>
<tr>
<th>Design Activity</th>
<th>Context</th>
<th>Knowledge Source</th>
<th>Design Issues</th>
<th>Work</th>
</tr>
</thead>
<tbody>
<tr>
<td>Conceptual</td>
<td>Users</td>
<td>Inspiration</td>
<td>Goal</td>
<td>Business model</td>
</tr>
<tr>
<td>Prototyping</td>
<td>Client</td>
<td>Documentation</td>
<td>Design Feature</td>
<td>Org. size</td>
</tr>
<tr>
<td>Biz Analysis</td>
<td>Environment</td>
<td>Tacit</td>
<td>Evolution of</td>
<td></td>
</tr>
<tr>
<td>Iterative</td>
<td>C.O.P.</td>
<td>Publications</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Tradeoffs</td>
<td></td>
<td>People</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Evolution</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Interpretation of feature</th>
<th>Critique</th>
<th>Suggestion</th>
<th>Use Case</th>
<th>Relevance of feature</th>
</tr>
</thead>
<tbody>
<tr>
<td>of process</td>
<td>Concept</td>
<td>Feature</td>
<td>Scenario</td>
<td>of feature</td>
</tr>
<tr>
<td></td>
<td>Vocabulary</td>
<td>Process</td>
<td>User</td>
<td>of process</td>
</tr>
<tr>
<td></td>
<td>Process</td>
<td>Usability</td>
<td></td>
<td>of concept</td>
</tr>
<tr>
<td></td>
<td>Feature</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 3.2: Codebook for interview transcripts (design practice)

advised in Weston et al. (2001). Primarily, the sole developer of the code book and coder was the researcher. There were no checks with other researchers to help ensure that the scheme identified the most important themes present in the transcripts. Additionally, the researcher did not ask any of the participants if the coding scheme was appropriate nor if the coding items associated with passages from the participants’ interview transcript seemed appropriate to that participant. In light of the careful and complete description of the methods outlined above and the limited scope of this research project, the researcher is nevertheless confident that the resulting analysis is adequate for the purposes of evaluating the general utility of the framework.
Design Iterations

This chapter presents the informal, iterative development of the GO-DKL framework. This work represents the bulk of the time spent on the project, as fairly labour-intensive development efforts were undertaken during the course of later iterations. The discussion begins with a description of the researcher and supervisor’s early work predating the thesis project’s specific research question, and ends with the motivations for the current version 3 of the framework.

4.1 Early work

The idea for structuring online deliberation design knowledge in a goal-oriented format first surfaced in a small project for the supervisor’s course INF2183: Knowledge management and systems (Hilts and King, 2010). Results from this project went on to be published in the First International Workshop on Modeling Social Media, (MSM ‘10; see Hilts and Yu, 2010).

The approach outlined in that publication – hereafter referred to as the MSM approach – was focused on a single exploratory case study. This case was modeled after MIT’s Climate
Collaboratorium\textsuperscript{1}. The goal of the project was to demonstrate how goal and agent-oriented modeling techniques can be used to represent design knowledge in the domain’s scholarly literature.

\subsection*{4.1.1 The Climate Collaboratorium}

The application domain modeled in the MSM approach was based on a study of MIT’s Climate Collaboratorium (Hilts and King, 2010)\textsuperscript{2}, an online community that aims to function as a “Wikipedia of the climate change debate”.

Users of the Collaboratorium are tasked with the creation of plans to mitigate the effects of climate change. The website interfaces with sophisticated climate change simulation models, and user plans are based on the manipulation of certain input parameters to the simulation models.

The plans are then debated and compared by the wider community, who ideally leverage their collective intelligence in order to elect feasible plans. Threaded discussion forums and voting play a large role in the decision-making process, as do argument summarization and formalization.

Hilts and King (2010) interviewed Collaboratorium stakeholders, such as the web developer, project manager, moderator and potential users. From these interviews, goals and problems were identified that the website had inadequately met. These goals were then used to guide the approach outlined below.

Two major themes dominate the identified goals of the Collaboratorium: plan understanding and community development and retention. These themes can be specified into clearer goals:

\textbf{Plan Understanding}

\textsuperscript{1}Now called the Climate CoLab. http://www.climatecolab.org
\textsuperscript{2}Unpublished paper. Contact this document’s author to request a copy.
• **Descriptions, impacts, and models be understandable:** Knowledge about user-created plans needs to be easily obtainable. This includes clear plan descriptions, more concrete explanations of the forecasted impact of plans, and an accessible explanation of the climate simulation models used to generate the forecasted impacts.

• **Plan comparison be facilitated:** As users are tasked with debating, evaluating and electing climate mitigation plans, the plans themselves must be comparable, ideally on numerous different facets. Comparing the relative strengths and weaknesses of plans is invaluable, and should be as clear and accessible as possible.

• **Community’s assessment of plans be emphasized:** While comparing plans, it would be useful for users to be presented with the community’s assessment of the two plans on various criteria.

• **Scientific accuracy and accessibility be balanced:** The climate simulation models are currently treated as black boxes, and plans themselves are often merely quantitative input parameters to those models. Explicating the meaning of those parameters and the reasons behind the selection of their value would be helpful.

**Community development and Retention**

• **Users be incentivized to participate:** The Collaboratorium needs to provide a sense of value to its users and the contributions they make to the website when developing plans, debating, and deciding.

• **Niche group development be facilitated:** users with shared or complementary interest should be aided in finding one another and creating plans together. If successful, robust communities of practice could support the website’s users in creating strong plans. These communities could furthermore strengthen the bonds among users, leading them to continue using the website.

• **User opinions be considered by plan makers:** In order to more fully leverage the collective intelligence of its user base, the Collaboratorium should encourage plan makers to consider any reasonable suggestions or objections made to a plan; indeed this consideration should be communicated explicitly to users. In this manner, a reciprocal relationships of respect and trust can be developed.

• **Selected plans have real policy influence:** In order for the Collaboratorium’s users to feel motivated and incentivized to contribute to the community, the perception that the elected plans be considered in real policy decisions is an important one.
4.1.2 The MSM approach

The MSM approach proposed a method of codifying and representing passages from design-oriented documents that explore the relationships among stakeholders and technology. These codified passages were then concisely represented using the $i^*$ framework. $i^*$ was chosen as a representational framework due to its efficacy in representing the relationships between and among stakeholders and technical solutions (Pastor et al., 2011). The intention behind this method was to facilitate the analysis of a domain’s research from a systems requirements engineering perspective that focuses on social relationships.

The approach began with a small review of social science, Information Systems, CSCW and HCI literature related to e-participation and online deliberation. Each relevant document was read carefully, and relevant passages that outlined relevant stakeholders involved in such systems and — importantly — their motivations were excerpted. Special attention was paid to motivations relevant to the Collaboratorium project; therefore, passages dealing with information and opinion sharing were sought.

In order to concisely represent a range of related motivations, $i^*$ concepts were identified within and among the selected literature passages. Figure 4.1 depicts how these passages became codified as $i^*$ elements and depicted using that modeling notation. Most of the resulting model elements in this example are softgoals and their interrelationships (expressed as contribution links and dependencies). These goals and contribution links were contained within the relevant actor. The definition of several intuitive relationships among goals and actors was required in order to ‘flesh out’ the model. The researcher’s knowledge of the domain guided this process. This process was complemented by the goals identified from the interviews discussed above.

Once completed, the model may be assessed using the qualitative evaluation procedure outlined

---

3 Similar to the approach taken in this project, though less formal.
The MSM approach: Qualitative abstraction, codification and modeling of academic literature in Horkoff and Yu (2009). This approach calls for the selection of design alternatives that the analyst believes will help achieve the project’s goals. The impact of these choices are elicited from the values of the contribution link connecting model elements \(^4\).

The MSM approach can guide the development of \(i^*\) models of sociotechnical relationships that are contextually situated and may help designers to make decisions about system features. However, we recognized that it does little to address Davies’ (2009) concerns of collaborative knowledge sharing across the discipline in order to help move the field forward. This limitation, as well as the overall approach taken, inspired the subsequent development of this thesis project.

\(^4\)as per the more detailed explanation in the literature review
4.2 Versions of the GO-DKL framework

We now turn to a description of the actual development of the GO-DKL framework conducted over the course of this thesis project. Three major versions of the framework were developed, each introducing new features or reconceptualizations that built on lessons learned from earlier versions, as shown in Table 4.1. The development of these versions is described in the following sections.

4.3 Meta Model

A meta model has been the conceptual foundation of the GO-DKL framework’s treatment of design knowledge throughout its development. This model delineates various kinds of knowledge entities and their relationships to one another. The meta model evolved from the document coding scheme used in version 1 of the framework into a more holistic representational tool used in version 3; it evolved by also including entities having to do with a designer’s analysis of the codified knowledge. See the next chapter for a concise description of the meta model’s current form.

<table>
<thead>
<tr>
<th>Version</th>
<th>Central focus</th>
<th>Browser features</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Visualize relationships</td>
<td>Proto-design assembly</td>
</tr>
<tr>
<td>2</td>
<td>Designer project as starting point</td>
<td>Goal-driven browsing</td>
</tr>
<tr>
<td>3</td>
<td>Contextualization and control over results</td>
<td>Project model configuration</td>
</tr>
<tr>
<td>3.1</td>
<td>Summary of design recommendations</td>
<td>Reporting mechanisms</td>
</tr>
</tbody>
</table>

Table 4.1: Major versions of the GO-DKL framework and browser
4.3.1 Evolution of the coding scheme

The coding scheme\(^5\) began with a basic objective – to extract relationships between design features and stakeholder goals from each publication. It was well understood that each publication was situated in a unique context. Thus, before any tests were undertaken, the coding scheme’s first iteration was configured as follows:

- Publication details
- Context description
- Design Decisions
- Goals
- Actors
- Relationships among actors, goals and design decisions

The first coding iteration of 20 publications found that design alternatives were often not directly related to a high-level stakeholder goal. Intermediary goals or tasks often were used as a means of demonstrating how exactly a design decision might impact goals. For example, anonymity might encourage frequent posting, which in turn could help to achieve the goal participation be increased. These intermediary elements were subsequently added to the coding scheme’s second iteration.

At this stage in the research, the coding process stipulated that each of the 20 documents to be coded should be examined carefully and completely during the course of each individual step in the process (shown below), so that each step’s domain concept is adequately focused on. The process attempts to provide a time for reflection so that proper instantiated concepts may be coded by first coding the meta concepts separately from their instantiations. This process was designed to help ensure consistent coding across documents.

There were three major steps in this first iteration:

\(^5\) As introduced in Section 3.2.2
The next iteration of 20 documents yielded the insight that a source often relates multiple design features or goals to one another as part of a larger network of relationships. By recognizing this, another entity in the meta model was developed. The notion of a ‘claim’ was used as a structuring mechanism for a subset of relationships found in the publication. This conceptualization recalls the notion of design ‘precedents’ made up of relationships between ‘issue’, ‘concept’, and ‘form’ (Oxman and Planning, 1994). Therefore, the third iteration of the domain coding scheme and meta model conceives of design knowledge as being located in publications, expressed in claims that are situated in a context and consist of design features that contribute to goals which are held by actors, as shown in Figure 4.2.

Following the introduction of the “claim” concept to the coding scheme, the coding methodology likewise shifted away from the individual meta-concept-driven coding practice.
(scan entire document for metaconcept, repeat), to a claim-by-claim coding methodology. In practice, this consists of first identifying a claim based on the presence of one of the meta concepts, followed by the identification of those concepts relating to the first identified concept in that claim. The next chapter – which concisely describes version 3 of the GO-DKL framework – provides a clear example of this process.

### 4.3.2 Expansion to support analysis work

Following the analysis methodology’s first design iteration, the need for a formal model to represent knowledge from a designer’s project became apparent\(^6\). At a basic level, storing the project’s name and description were required. As a goal-directed browsing method was undertaken in that methodology’s second iteration, the project component of the meta model was developed to include a designer’s initial project goals and their associations with library items, as well as selected goals from the library for inclusion into the project. The below list outlines the basic project data concepts included in the project component of the meta model:

- Project Name
- Description
- Initial (Project) Goals
- Included Concepts

An interface between a designer’s initial goals and library items was deemed necessary in order to support association between the two items. In order to simplify the complexity of the large array of library items, a means of structuring, or classifying them was required. This classificatory structure emerged, based on ‘aggregate goals’, categorical goals that library goals were associated with. The researcher inductively developed these aggregate goals by qualitatively examining every goal in the knowledge base and looking for clusters of related items. The

\(^6\)See the following section for more details.
common feature of these clusters was then determined and used as the label for an aggregate goal. For example all viewpoints be considered and agreement be sincere were both classified as ‘normative goals’, and thus associated with the aggregate normative ideals be upheld. This process thus enables a prototype system to retrieve a list of library goals that have been associated with the same aggregate goal that the designer has associated with his / her initial goal. See the next chapter for a list of aggregate goals.

Once the analysis methodology included user-customizable project models, more data requirements emerged. In order for designers to reconfigure the contribution links between included library elements, a project model-specific data store was required. Therefore, nodes and relationships from the library needed to be similarly expressed within the project model expressed as ‘project model relationship’. This enables designers to create as many unique reconfigurations of the library data as they desire, on multiple projects.
4.4 Analysis Methodology

The discussion now turns to the incremental development of the framework’s analysis methodology. Questions regarding analysis process, desired representational mechanisms, and user input requirements were addressed as the GO-DKL browser – the analysis prototype – was refined.

4.4.1 V1: Tabular approach

Version 1 of the GO-DKL browser prototype provided a complete overview of the knowledge base’s claims from which users could select items relevant to their own context, shown below in 4.4. This selection process recalls the work of Wahid et al. (2006), specifically their focus on designer construction of ‘proto-designs’ from codified knowledge. The framework’s tabular presentation could be sorted by most elements in the current meta model. In this manner, all claims related to a single goal or design alternative could be assessed together and chosen for inclusion in a project. The table could be filtered by aggregate goal, to provide some scale management. Selected items would then be grouped into a project, an overview of which could then be analyzed. This overview was simply a list of selected items paired with another list of items that had been shown in the literature to be related to the selected items.

However, upon reflection, we deemed that this process had several limitations. First, scale issues forced us to reconsider presenting a complete overview of the knowledge base; even with the possibility to filter the items by aggregate goal, the tabular structure itself appears imposing in the sheer amount of different data items presented. Second, we felt that additional support for the user to associate his/her contextual knowledge with knowledge base items would increase applicability and help reduce the user’s cognitive load by enabling users to filter results into manageable, relevant subsets. A final limitation was the general directionless approach enabled
by this interface to the knowledge library. It did not clearly support a systematic analysis process, but rather an unstructured one.

Figure 4.4: The tabular approach to exploring the data. Yellow = selected; red = negative claims; green = positive claims

4.4.2 V2: Goal-driven approach

In keeping with the goal-oriented design of the knowledge base itself, a goal-directed analysis procedure was deemed to be a sensible means of mitigating the above-mentioned concerns regarding the version 1’s tabular approach. This section describes the rationale behind the design choices involved in version 2 of the GO-DKL framework. Table 4.2 provides a definition of the
<table>
<thead>
<tr>
<th>Term</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project</td>
<td>A design project that utilizes the GO-DKL knowledge base</td>
</tr>
<tr>
<td>Initial goal</td>
<td>The initial goals for the project</td>
</tr>
<tr>
<td>Aggregate goal</td>
<td>High-level concept used to link initial goals to related library goals</td>
</tr>
<tr>
<td>Library goal</td>
<td>Extracted from the literature, stored in the GO-DKL knowledge base</td>
</tr>
<tr>
<td>Included goal</td>
<td>A library goal that has been selected and included into a project</td>
</tr>
<tr>
<td>Impacted and related goal</td>
<td>Other library goals that are impacted by suggested design features for included goals</td>
</tr>
<tr>
<td>Project model</td>
<td>A subset of relationships between design features and goals, based on the included goals of the project.</td>
</tr>
<tr>
<td>Goal graph</td>
<td>A graphical representation of the relationships found in the project model</td>
</tr>
<tr>
<td>Model slice</td>
<td>A subset of the project model, centered on a focal goal</td>
</tr>
</tbody>
</table>

Table 4.2: GO-DKL terminology

various goal types referred to in the subsequent sections. It serves to clarify and distinguish the terminology associated with the GO-DKL framework.

Goal association

As a means for beginning a context-sensitive, goal-directed analysis of the knowledge base, it was determined that the designer should define a project, including major conceptual project goals, prior to exploring the data itself. The designer would thus begin to explore the knowledge base with his / her project clearly in mind. The interface between designer goals and the knowledge base items occurs at the next stage - the associative stage. This is similar to Oxman and Planning’s (1994) retrieval approach whereby designers define their focal issue through a controlled vocabulary. Go-DKL Version 2 adopted a similar process: the designer classifies his / her initial project goals based on the inductively developed aggregate goals and subsequently selects relevant retrieved library items. In this manner, a designer is intended to recognize how any selected goals were relevant to his / her project; these retrieved goals are instances of a class
that the designer manually selected and associated with his / her project.

In order to exemplify this process, the Collaboratorium initial project goals defined at the start of this chapter, in section 4.1.1, page 48, will be used as sample data. Following the definition of the goals, the designer next associates them with the aggregate goals. Table 4.3 presents a summary of the goal classifications of the Collaboratorium initial project goals, along with rationale for those choices. We classified the initial project goals informally, based on our understanding of the project.
<table>
<thead>
<tr>
<th>Initial Goal</th>
<th>Aggregate Goal</th>
<th>Rationale</th>
</tr>
</thead>
<tbody>
<tr>
<td>Descriptions, impacts, and models be understandable</td>
<td>participation</td>
<td>Understanding leads to participation</td>
</tr>
<tr>
<td></td>
<td>personalization</td>
<td>Understanding varies per user</td>
</tr>
<tr>
<td></td>
<td>accessibility</td>
<td>Understanding is an accessibility issue</td>
</tr>
<tr>
<td>Plan comparison be facilitated</td>
<td>community building</td>
<td>A community that critiques each other’s plans is important.</td>
</tr>
<tr>
<td></td>
<td>information</td>
<td>Comparison is easier if comparable plans are easily found.</td>
</tr>
<tr>
<td></td>
<td>management</td>
<td>Comparison should be a discussion.</td>
</tr>
<tr>
<td></td>
<td>discussion support</td>
<td>The comparison should produce a clear deliverable.</td>
</tr>
<tr>
<td></td>
<td>deliverable</td>
<td>Comparisons can’t be done unless users are engaged.</td>
</tr>
<tr>
<td></td>
<td>engagement</td>
<td></td>
</tr>
<tr>
<td>Community’s assessment of plans be emphasized</td>
<td>normative</td>
<td>Community assessments should be inclusive of all viewpoints.</td>
</tr>
<tr>
<td></td>
<td>community building</td>
<td>Emphasizing communal, rather than individual assessment is important.</td>
</tr>
<tr>
<td></td>
<td>information</td>
<td>We need to effectively gather information to emphasize a community assessment.</td>
</tr>
<tr>
<td></td>
<td>management</td>
<td>The assessment should be concise.</td>
</tr>
<tr>
<td></td>
<td>deliverable</td>
<td></td>
</tr>
<tr>
<td>Scientific accuracy and accessibility be balanced</td>
<td>trust</td>
<td>The balance needs to be trustworthy so plans are still reputable.</td>
</tr>
<tr>
<td></td>
<td>information</td>
<td>Managing information in an accessible way is a challenge.</td>
</tr>
<tr>
<td></td>
<td>management</td>
<td>This is clearly an accessibility issue.</td>
</tr>
<tr>
<td></td>
<td>accessibility</td>
<td></td>
</tr>
<tr>
<td>Users be incentivized to participate</td>
<td>community building</td>
<td>Participation and community are intertwined</td>
</tr>
<tr>
<td></td>
<td>participation</td>
<td>This is obviously a participation issue.</td>
</tr>
<tr>
<td></td>
<td>engagement</td>
<td>Engaging with one another is another important component.</td>
</tr>
<tr>
<td>Niche group development be facilitated</td>
<td>community building</td>
<td>Special interest groups could become vibrant communities</td>
</tr>
<tr>
<td></td>
<td>identity</td>
<td>These groups have distinctive identities</td>
</tr>
<tr>
<td></td>
<td>group support</td>
<td>Group support technologies are essential</td>
</tr>
<tr>
<td>Selected plans have real policy influence</td>
<td>deliverable</td>
<td>Plans must be clear and concise and emphasize relevance to policy issues</td>
</tr>
</tbody>
</table>

Table 4.3: Goal classifications for the Collaboratorium Case Study
4 Design Iterations

Retrieving and Analyzing Impacted and Related Goals

A common problem for design stakeholders is an inadequate assessment of the domain in which a software project will be implemented (e.g. Curtis et al., 1988). This unclarity or incomplete understanding of how a project fits into the wider domain can limit the efficacy of the developed artifact. Due to this, several requirements engineering and design science methodologies attempt to help stakeholders clearly elicit goals and define domain models. The following discussion outlines this project’s attempt to follow that tradition.

Once a designer has selected several library goals to include in his / her project, the system is capable of retrieving every design feature that has an impact on those goals. These design features, however, also contribute to unselected goals. Thus, it would be sensible to list these potentially relevant goals to the designer, and also to provide a sense of a) why those goals might be relevant; and b) whether those goals would be aided or hurt by the retrieved design features.

These goals are labelled as impacted and related goals, because they have been indirectly associated with the project. Colour coding of the goals helps to indicate the type of contribution the impacted and related goals are found to receive from the retrieved design features. The prototype system then provide designers the option to include these goals in the project as they did with the goals selected earlier. In this manner, related goals within the application domain can be considered.

4.4.3 Dealing with goal models

Goal models can be generated from a designer’s included goals to pursue a more systematic analysis of the knowledge base. These models are created from the selected goals and retrieved relationships. There is a well-tested and clearly-defined approach to evaluating goal models (discussed in Chapter 3), and so the major challenge for this project was in delineating the scope
of the model. Essentially, this challenge consisted of balancing the trade-offs between model scale and analytic utility.

**Intuitive relationships**

Early models generated from the knowledge base – limited only to knowledge extracted from scholarly sources – did not facilitate the type of analysis usually attributed as a central benefit of goal models. That is, they did not facilitate the identification of alternative solutions to a single problem. In goal-oriented terms, the framework cannot generate clear alternative means to a single end based solely on the literature. This is due to the fact that each publication in the domain generally deals with separate goals situated in a specific context, as well as the relatively small number of publications codified. Many contextual factors in each specific publication may contribute to a goal’s achievement; and thus to maintain that the knowledge in the base is contextually situated, we do not claim that the knowledge is a solid means toward an end. It is up to the designer to configure the model to suit the variables present in his or her context.

**Figure 4.5:** Project goal graph with no intuitive relationships; note the lack of intersecting links in the lower areas (lack of clear alternatives)

The need to connect the relationships outlined in disparate publications emerged as a clear design challenge at this stage. We wanted to facilitate comparison between different design
features reported in the literature, yet also maintain that the knowledge is contextually situated. This was part of the motivation behind the creation of various claim types\(^7\), one of which we call ‘intuitive relationships’. These intuitive relationships link previously unrelated items together, based on an intuitive understanding of the domain. We defined these based on our domain knowledge. As such, the claims are distinct from the literature-based claims, yet they are necessary to function as a kind of glue between the literature items.

The inclusion of intuitive relationships permits the generation of robust models that feature multiple contributions to single nodes on the goal graphs, which permits grouping branches of the graph into clear alternatives. However, it also introduces a scale problem to the models; while in theory these choices simplify the analysis of design alternatives, the sheer number of relationships present in the model results in unmanageable complexity. Models take hours to properly analyze using this approach. Developing a scale management strategy was therefore deemed necessary.

**Scalability problem**

The knowledge base, once it included intuitive relationships – even at an early stage of 20 sources – was capable of producing large, almost impenetrable goal models that offered little utility to designers. These goal models are meant to be analyzed by selecting a focal node in the graph, and assessing the various contributions it receives from other nodes. However, as shown in figure 4.6, identifying those linked nodes proves to be an immense task when the sheer number of links that intersect across one another serve to obfuscate those relationships. This is a common problem with these types of goal models (see Pastor et al., 2011). Therefore, we determined that managing the scale of models is of great concern with regard to the usefulness of version 2 of the framework. With consideration to the above discussion of intuitive relationships as well as the process by which impacted and related goals are retrieved, we will now discuss several

\(^7\)A more detailed explanation is located in the subsequent chapter.
approaches to filtering the items that are used to populate goal graphs.

The first approach would be to not include any intuitive relationships (as shown in Figure 4.5), but this approach suffers from the lack of alternatives analysis support as mentioned above. The second approach is filtered retrieval, wherein the design features retrieved to the designer are only those related to the included goals via actual publications (omitting intuitive relationships). However, the intuitive relationships that those design features shared with other items in the knowledge base would also be retrieved. With this approach, the middle of the goal graph becomes the most complex, with relatively clearer lower areas denoting the design features (see Figure 4.7).

The third approach was to retrieve all design features (including those connected through intuitive relationships), as well as all intuitive relationships to higher-level goals – in other words, not addressing the problem. This approach suffers from the above-mentioned scale problem, but might be appropriate for smaller models.

Another technique investigated was “goal slicing”. This work was based on Leica (2005), who
proposed a method of analysing a subset of a larger $i^*$ graph. The subset is delineated based on the immediate relations of a single focal goal in the model. In Leica’s framework, a designer chooses a single included goal to investigate, and the system retrieves all related design features AND the other included goals impacted by those features. This approach can enable analysts to focus intently on the general environment of one goal, repeating these slice analyses throughout the context of the larger model. See the Chapter 5’s figure 5.7, which depicts a goal slice.

These two techniques – filtering and slicing – are complementary to each other. A goal slice may be filtered based on any of the three methods described above. By supporting these various model scale management techniques, these features are hoped to provide additional control as to how the designer / analyst interacts with the generated models. We hope that this would help make the retrieved models more usable, though during this project we did not have the time nor resources to properly investigate their utility and usability.

Figure 4.7: Climate Collaboratorium project goal graph with filtered intuitive relationships
Goal model configuration

Managing the scale problem might have been simplified had the framework included ‘pre-made’, legible goal model slices for each library item, as in Esfahani and Yu (2010). These models could be individually analyzed or combined to form a larger project model. However, this approach was not pursued for two reasons. The primary reason has to do with a different scalability concern; that of the knowledge base itself. Theoretically, as the knowledge base grows, each new item would require the manual creation of a new model. This extra step of human intervention would be time consuming and limit the rate at which the knowledge base could expand.

Another important concern was the degree of context-independence of the library items. Based on an recognition that each project is uniquely situated (see Suchman, 2007), the GO-DKL framework endeavours to limit the degree to which the system frames its contents as objective knowledge applicable across contexts. Instead, several design choices have been made to enable contextualization of the knowledge. Including an initial project goal definition and association stage is one means for designers to contextualize retrieved items. Additionally, our decision to omit static models from the knowledge base led to the adoption of the alternative - configurable models. The first attempt at configurable models were those generated automatically by version 2 of the GO-DKL browser. This version, however, suffered from the model scale problem. We now turn to GO-DKL version 3, which features our second approach to configurable models that also address this scale concern.

4.4.4 V3: Project Models

A more radical attempt at dealing with the scale problem inherent in retrieving all relationships relevant to a project was to move away from the traditional representation of goal models. As the above figures have demonstrated, the identification of alternatives is quite difficult when the goal
graph is densely interconnected. While the model slicing methods discussed above help reduce this complexity, there are still many intersecting lines, and design features that contribute to the same goal might be spread across the topology of the graph. To address this concern, we represent the goal model in a web browser-based, collapsible and expandable hierarchical tree-list. A designer may select a focal aggregate goal, and ‘drill down’ by expanding the selection to reveal all included goals in the model that are associated with the aggregate goal. The designer may repeat this process by expanding those goals to eventually reveal the design alternatives that contribute to the satisfaction of the focal goal. In this manner, the designer may quickly isolate and analyze goals of interest.

This representation of the model is also configurable. A designer may modify the contribution value or a relationship between elements based on his / her intuition and understanding of the application context. Relationships and nodes may be excluded from the model. By excluding these irrelevant components of the project model, the designer may work towards a contextualized and focused project model.

However, this browser-based approach does not easily facilitate tradeoffs analysis. This is in contrast to less complex goal graphs, which can facilitate this process by depicting how various design alternatives to a focal goal might impact other goals in the model. Interrelationships may be depicted clearly as links between nodes. The tree-list representation described above cannot easily do this; one goal and its contributing child items must be focused on at a time. However, to mitigate this limitation, we have developed an automated evaluation procedure that propagates the effect of including or excluding a certain design alternative across the entire model. In this manner, a designer may change focus to another goal, and immediately perceive the effect of an earlier change, based on a change in colour of various affected goals across the model.

Furthermore, in order to complement a variety of design practices, this configurable project

8See appendices A2 and A3 for technical details.
model can generate the goal model slices discussed above. Therefore, the two approaches can work together; a designer may reduce the model scale in-browser and reconfigure relationships as need be, prior to a more formal analysis procedure using a modeling tool. The next chapter will describe these components in more detail.
The GO-DKL Framework

This chapter provides a complete description of the *Goal-oriented design knowledge library* (GO-DKL) framework and GO-DKL browser prototype, both of whose development was described in the preceding chapter. The first sections discuss the codification of the literature and its underlying model. The latter sections describe the procedures for analysing this codified knowledge. To begin, we re-introduce the purpose of the framework.

5.1 Purpose of the framework

This framework is meant to provide a platform for designers to explore past ideas and refine their understanding of their current project through exploration and selection of interesting goals that have been extracted from published literature, codified and stored in a knowledge base. Different design features (EG: moderation, displaying a user’s real-name, user profiles, comment threading, etc) are related to these goals, and the designer may use a prototype tool to select and analyse the impact of various configurations of these design options.
This selection and analysis process results in the creation of models of a designer’s project. Conceptually, these models are a network of interconnected goals and selected design features. The model may be represented in various formats, each with its own benefits and limitations. With these facilities at their disposal, designers should be able to assess consequences and tradeoffs associated with potential system designs.

Figure 5.1 depicts the processes by which knowledge is extracted from sources, codified and stored in a knowledge base. These are shown in the upper half of the diagram, and have been largely carried out by the researcher. The lower half of the diagram depicts the processes by which a designer associates knowledge base items, includes those selections in his/her project, and furthermore analyzes and configures those items and relationships to create a working project model.
5.2 Conceptual Model

The codification of published design knowledge is guided by a conceptual model derived from the contents and structure of design-oriented online deliberation publications. As shown in the preceding chapter, the research has iteratively explored the form of the information presented in these publications, and has identified key concepts that when related to one another, form a claim about the relationship between design features and stakeholder objectives.

This conceptual model is divided into three main areas. The first area is a library of published knowledge, the second is based on a design project, and the third is based on the role of an intermediary in connecting published knowledge to design projects. See Figure 5.2 for a complete graphical representation of this model.

5.2.1 Library data

The published data is centered around the concept of a claim, which can be divided into one of the following five claim types:

- **Empirical Observation**: A claim made by the researchers pertaining to a relationship directly observed in the study’s context.
- **Evidence-based theory**: A scholar’s citing of another theory in order to hypothesize about a relationship in the study’s context.
- **Guideline**: A claim that is well established in the literature, and treated as a commonly accepted principle that is fairly context independent.
- **Normative Theory**: A publication espousing or drawing upon the philosophical ideals of what a relationship in the study’s context should resemble.
- **Intuitive**: A claim connecting various items in the knowledge base developed by a knowledge intermediary, based on his/her domain knowledge and intuition.

Claims are furthermore made up of (library) relationships between domain concepts. Relationships are valued: they can be one of many types (based on $i^*$ syntax; see chapter 2):
• Helps (+)
• Hurts (-)
• Makes (++)
• Breaks (–)
• Unknown (?)
• And (AND)
• Or (OR)

The domain concepts are linked together by relationships and can be one of the following types:

• Design feature: Regarding the configuration or inclusion of a functional capability within a system
• Goal: Intentional desire of an actor. Represents a desired ’end’, and not the means towards it.
• Actor: Something (or someone) that acts upon the context in some way. The actor must exert some level of autonomy and is likely in an interdependent relationship with other actors.

5.2.2 Project data

Designers define a project by outlining a set of initial goals they would like to achieve. Relevant concepts from the library data are stored as included concepts. Based on the relationships in the knowledge base among these included concepts, a project model is generated.

The project model is made up of every project relationship for that project, and represents the overall, interrelated contributions of various design features to the various included goals of the project. This relational data can then be represented and analyzed, and attractive design alternatives chosen (see the upcoming section 5.4 for more details).
5.2.3 Intermediary data

In order for the library and project data to be usably linked to one another, an intermediary must define certain associative elements situated between these areas. For example, various ‘aggregate goals’ have been defined that can be seen as “super classes” of a subset of library goal concepts. At the same time, a designer associates his/her project’s initial goals with one or more aggregates. This enables the retrieval of relevant library concepts.

Aggregate goals are developed by domain experts by grouping related library goals together and inductively defining a highly conceptual goal that encompasses the shared qualities of the grouped goals. The following list presents the aggregate goals used in the evaluation of the framework.

- **Accessibility be rigorous**: Issues surrounding accessibility.
- **Activities be trustworthy**: Relating to participants placing their trust in the website.
- **Community be developed**: Issues regarding the development of a strong, lasting community.
- **Deliverable creation be facilitated**: Regarding discussions producing some kind of concise result or summary.
- **Discussions be facilitated**: The platform’s ability to foster discussion among users.
- **Group activity be facilitated**: The formation and facilitation of small discussion groups
- **Identity be manageable**: Providing users with a degree of control over their identities in this software.
- **Information be managed effectively**: Issues regarding how information is represented, retrieved, and user behaviour monitored
- **Normative ideals be met**: Social/moral norms including but not limited to fairness, diversity, consensus, politeness
- **Participation be broad and sustainable**: Concerns surrounding attracting new users and creating sustainable participation
- **System be personalizable**: User customization of the software.
- **Usability be rigorous**: Issues surrounding ease of use.
Another area where intermediary data is required is in the definition of intuitive relationships among library concepts. In order for various sources’ reported claims and concepts to be usably interlinked with one another, concepts from one publication should be linked to concepts from another, where appropriate. This activity should only be undertaken by a domain expert. These intuitive relationships are likely more debatable than standard library relationships, and thus should be clearly differentiated from the others. As the knowledge base grows in size, this process may hopefully diminish in importance and necessity.

5.3 Codification Process

At this stage, a relevant publication is read over and marked up based on the library concepts mentioned above. Once many documents have gone through this process and their results have
been stored in a database, a knowledge base emerges.

When codifying documents, we look for claims. These claims are made up of relationships between design features and goals which are held by actors. Phrases in the publication’s text that appear to indicate a design feature, goal or actor are highlighted and stored as concept instances. Similarly, passages outlining a relationship are highlighted and stored as a type of relationship between two concepts. Relationships are also stored as a part of a larger claim.

Based on the example shown in Figure 5.3, the codification process would store approximately the following in the knowledge base:

- Claim 1 (Guideline)
  - Relationship 1
    - Personalization (design alternative)
    - helps (relationship type)
    - Access to interesting Content (goal)
  - Relationship 2
    - Access to interesting Content (goal)
    - helps (relationship type)
    - Participation be increased (goal)
  - Relationship 3
    - User (actor)
    - has goal (relationship type)
    - Access to interesting Content (goal)

This process is repeated for every document to be codified. QSR NVivo was used for this process. Please see appendix A1 for a complete list of publications used in GO-DKL version 3’s knowledge base.
5.4 Usage guide: the framework as design springboard

The second major component of the framework is support for a designer to contextualize and analyse the knowledge base contents. This process involves several steps and was first depicted in figure 5.1. The list below outlines these steps, the analytic process through which designers associate library knowledge with their project, assess the resulting project model, and configure the model to their context. The sub lists represent iterative activities that may be repeated and refined as necessary.

1. Define project
2. Select one initial goal to analyze
   2.1. Associate goal
   2.2. Select library goal
3. Browse impacted and related goals
4. Create project model
5. Analyse project model
   5.1. Configure model
      5.1.1. Analyze goal slice
      5.1.2. Configure goal contributions and related concepts
   5.2. Analyze Project Model
   5.3. Generate Report(s)
5.4.1 Define Project

To begin the process, the designer provides a project name, short description, and list of initial goals to the knowledge base’s front-end system interface\(^1\). These goals should be stated in concise prose (no more than 6-7 words). These goals should be limited to high-level goals for the deliberative process, rather than specific technical solutions or ideas. In other words, the desired outcome is what is important to input at this stage, rather than the means.

5.4.2 Select one initial goal

Next, the designer chooses one initial project goal to analyze in detail. The following stages are then repeated for each initial goal.

Associate initial goal with aggregates

The first stage of analysis is to associate the chosen initial goal with aggregate goals. Recall that these aggregate goals act as categories for the goals in the knowledge base, and were defined by an intermediary (in this case, we performed the role). In the below example, shown in Figure 5.4, the designer has associated his initial goal Niche group development be facilitated with the aggregates Community be developed, Discussions be facilitated and Group activity be facilitated.

\(^1\)The prototype GO-DKL browser was developed for this project, using HTML, CSS, JavaScript, PHP and MYSQL.
Browse and select from retrieved goals

The system next retrieves lists of library goals, each being centered on one of the aggregate goals associated with the initial project goal. The designer then explores the retrieved goals, perhaps examining their descriptions and publication of origin (see Figure 5.5). Finally, the designer includes those that seem relevant to the current initial project goal (in this case: Niche group development be facilitated).

This association and selection process occurs for each project goal and may be iterated as desired.

5.4.3 Browse impacted and related goals

Once all initial goals have been associated and relevant library goals stored as included goals, the system retrieves and presents the designer with a list of goals that have not been included, but might be of interest to the project. They are called impacted and related goals, and are related to the project via the project’s associated design features. In order to calculate this list of impacted
and related goals, the system assesses every relationship in the knowledge base involving a design feature contributing to any of the included goals. This list of contributing design features is then queried for additional library goals that have not been stored as included goals. These retrieved goals are impacted and related goals, presented in Figure 5.6.

The impacted and related goals are divided based on the type of contribution they receive from the design features.\(^2\) The prototype system employs a color scheme to clearly differentiate the types of impacted and related goals; red goals are negative, orange have received contradictory evidence, and green are positively impacted related goals.

The designer is asked to consider these goals and whether any of them would be relevant to the

\(^2\)Specifically, the system will examine each stored relationship between the design feature and library goal. Each relationship type (help, hurts, etc) is given an integer (helps +1, hurts -1, etc). If the sum of all relationship integers for a given set of relationships between feature and goal is above 0, it is labeled as positively impacted. Less than 0, it is labeled negatively impacted. If 0, it is labeled Contradictory evidence. For example, negatively impacted related goals are contributed to negatively by the project’s associated design features.
The objective of this stage is to help identify unforeseen consequences of various design features.

### 5.4.4 Create Project Model

Once the designer has completed an iteration of goal association and stored a satisfactory amount of included goals, he/she may next develop a project model.

A project model is generated automatically and initially depicts the following:

1. Every included aggregate goal
2. Every included goal associated with any of those aggregate goals.
3. Every relationship in the library involving those included goal receiving a contribution from another goal or design feature.
4. The contributing goal or design feature and its contribution value (helps, hurts, etc)
The resulting model can be conceived of as a tree of goals, with high level aggregate goals at the top of the tree, which are situated above connected library goals, which in turn are above connected design features. This model includes all the knowledge base’s relationships between the included design features and the included library goals. This model can therefore be used as the basis of detailed analysis of the impact of selecting various configurations of design features on included goals.

5.4.5 Analyse project model

The designer next selects the model in order to view and configure it. The initial representation of the model is an interactive tree list. The tree list’s top level shows the included aggregate goals. When a user clicks on an aggregate goal, all included library goals associated with that aggregate become visible. Each goal may be clicked on, and their contributors shown. For example, Figure 5.7 shows an expanded view of contributors: starting at the lowest point in the hierarchy, interweave content and discussion and its siblings contribute to evaluate content, which in turn contributes to decision making be facilitated, and so on.

Beside each item (goal or design feature) is a dropdown list. The initial value of this list reflects the relationships between the item and its parent item as reported in the literature and stored in the knowledge base. As mentioned above, the relationship values include ‘help’, ‘hurt’, and several other concepts. This list representation is intended to allow designers to easily explore the relationships between design features and their included project goals.

While traversing this list, the designer may examine the description of included design features and goals, as well as the claims from which they were derived. This might be done in order to assess the fit of that concept in relation to the project’s application context. Items in the list may

---

3See Appendix A2 and A3 for technical details.
be deselected, eliminating them from any reports or graphical models generated from the overall project model.

This list is not the only representational possibility for the project model. Indeed, in order to have a more holistic view of the effects of design features on various goals, the designer may click on the \( Q7 \) (Yu et al., 2007) button beside each included goal (shown in Figure 5.7) to be provided with a graphical model centered around that goal. See appendix A4 for a description of the conversion process. An example of a generated model slice is shown in Figure 5.8. Contributing design features are included, as are any other included goals affected by those design features. In this manner, a designer may assess the design features that might contribute to an important goal, while also being able to see the possible consequences those design features have on other goals. This is not easy to do through the list view. This kind of model “slice” enables focused analysis on a single goal and its “surrounding environment”, and is based on the work of Leica (2005).

The modeling paradigm used here is based on the NFR and \( i^* \) frameworks. As mentioned in the section 5.2, we depict design knowledge as valued relationships between design features and goals. Design features are depicted as \( i^* \) tasks, while goals are modeled as \( i^* \) ‘soft goals’.

When the designer has a good understanding of the consequences of various design
alternatives, he/she may reconfigure the project model based on that understanding and the specific unique details of the project’s context. While using the list representation that acts as an interface to the knowledge base, a designer may alter the contribution links between elements from their original retrieved state to a value that the designer feels is more appropriate for his/her context. This process would be guided by the designer’s own knowledge of his/her context.

In the example shown in Figure 5.9, the designer modifies the contribution of evaluate content to decision making be facilitated from ‘helps’ to ‘AND’. The ‘AND’ option would be selected if the designer feels that contribution must be selected alongside another contribution. See below that expand spaces for interaction is also marked as ‘AND’. Note that evaluate content is currently green, implying it would be satisfied by its contributing children.

The next Figure 5.10 shows the designer deselecting various concepts. These were deemed irrelevant to the current project. Note that the contribution type of group memory recording has been changed to ‘?’ and the colour of evaluate content has changed to orange. This colour shift notes that with only group memory recording selected, the goal evaluate content receives contradictory contributions. The other two potential contributions, interweave content and discussion and incremental formalization have been deselected and thus do not contribute

Figure 5.8: A modeled goal slice in OpenOME
Once various contributions have been reconfigured and certain goals omitted as need be, the user may use the Q7 button to generate a goal graph of the entire project in order to assess the

---

4See Appendix A3 for a description of the technical process underlying the evaluative labeling of project model elements based on child element contributions.
entire model. Every design alternative and goal in the project model is linked to others based on their contribution relationships. In this manner, a complete picture of the interrelationships and various impacts design alternatives might have on project goals can be assessed.

This analysis would be performed by importing the Q7 file into the Open OME tool, likely following the evaluation process described in Horkoff and Yu (2009). This process is interactive and qualitative, as opposed to the automatic and quantitative evaluative labeling used in the GO-DKL browser. Following the former process affords a reflective consideration and selection of alternatives with respect to the larger picture that is lacking from the analysis afforded by the GO-DKL browser.

Once a suitable configuration of design alternatives has been identified (based on iterative model development), the designer may formulate specific system requirements or develop rapid prototypes depending on the design methodology employed. The analysis process identified above should be used as a springboard for design ideas, and must be supplemented by in situ analysis.

### 5.4.6 Project Reports

Based on the interviews discussed in Chapter 8, the capability to generate two kinds of reports has been added to the framework. As it would take significant effort to analyze the entire project’s goal graph, the prototype GO-DKL browser can generate a simpler list of design features and their contributions to various project goals. This list would enable quick skimming and intuitive decisions by the designer, rather than detailed and systematic analysis.

The second report format is a simple bibliography of the publications referenced in the project model. It is ordered by the number of references found for a specific publication. Links to the original source publication are provided, so that a designer may learn more about the specific
relationships and contextual details presented in that publication. To help this process, a list of a project’s included goals that are found in the publication is presented beside the citation.
6

Application Example

A clear application of the GO-DKL framework’s analysis procedure is presented in this chapter. This is intended to function as a ‘descriptive evaluation’ of the framework (Hevner et al., 2004), and thus demonstrate how the various aspects of the procedure may be used to support a real-world design project. To accomplish this, the chapter outlines a hypothetical example project, follows the analysis procedures described in the preceding chapter, and finally discusses the project-specific results of using the framework\(^1\).

6.1 The FaceBook DSS Project

A group decision support system integrated into FaceBook is a project proposed as a means of studying who use social media to support decision-making and how. The mode of interaction proposed is deliberative; users will collaboratively discuss and debate issues and ideas in an online environment. The below list presents several design-oriented goals for the project:

\(^1\)The example uses data obtained from the knowledge base of publications (listed in Appendix A2) that were collected using the eligibility criteria outlined in Chapter 3 and codified according to the GO-DKL process
• Participation be incentivized: Users need motivation to contribute. Friends should easily contribute to each other’s questions.

• External information be considered: Decisions need to consider external information, which likely come from a variety of sources with different degrees of reliability. Users should be able to reference and make sense of this external information and effectively leverage it in their decision-making processes.

• Degree of consensus be obtained: Contributors to a question need to work together to build some kind of consensus or agreement about the topic at hand. Mechanisms to help encourage and support this behaviour will be required.

• Social pressures be reduced: In order to promote high-quality decision-making, social pressures such as minority anxiety or the ‘bandwagon effect’ should be minimized.

These high-level goals were decided upon based on knowledge of group decision-making behaviour, decision support systems as well as an understanding of FaceBook as a social platform. They serve as the data that the designer first inputs into the prototype tool to begin the analysis procedure, described next.

## 6.2 Project Definition

A designer and analyst on the ‘FaceBook DSS’ team, “Jane”, is tasked with exploring and analyzing the knowledge base. She begins this process by defining the project. Using the GO-DKL browser, Jane enters a name, description and the four above-mentioned goals into a form. Once submitted, Jane selects “FaceBook DSS” from the project list.

## 6.3 Goal Associations and Selections

Situated in the ‘FaceBook DSS’ project space, Jane next selects degree of consensus be obtained from the list of initial project goals. She then begins associating the selected goal with
the library’s aggregate goals. Jane decides that degree of consensus be obtained is related to both deliverable creation be facilitated and discussions be facilitated. She selects those two aggregate goals, and moves on to the next phase.

Jane next browses two lists of library goals – each list focused on one of the selected aggregate goals from the previous phase. While examining the discussions be facilitated list, Jane notices the goal emergent collaboration and decides to learn more about it. An additional information dialog box reveals to her that the goal was extracted from an empirical observation presented in the publication by Schwabe et al. (2010), and is defined as “the process of interaction should be evolvable and the system should support these dynamic processes of collaboration.”

Jane decides that this goal is relevant to the achievement of her initial project goal degree of consensus be obtained, and so, selects that goal. From this selection, the system stores a relationship between degree of consensus be obtained, discussions be facilitated, and emergent collaboration. Next, she selects several other goals from this same list: decision be reached, discussion be focused, knowledge is gained, and summarization be simple.

Jane then moves on to the second aggregate goal she associated with degree of consensus be obtained – deliverable creation be facilitated. She similarly explores the retrieved list of goals, learns more where need be, and selects the goals decision making be facilitated, discussion reach conclusion, foster discussion of differences and finally sensitive issues be surfaced.

Once satisfied with her selections relevant to her initial goal degree of consensus be obtained, Jane repeats this process for her other three goals. She notices that several library goals in the retrieved lists are already selected. She notes that she had previously associated them with another goal, and makes her new selections. The below Table 6.1 presents the goal associations and selected library goals relevant to each remaining initial project goal.
## 6 Application Example

<table>
<thead>
<tr>
<th>Project Goal</th>
<th>Association</th>
<th>Included Goals</th>
</tr>
</thead>
<tbody>
<tr>
<td>Degree of consensus be obtained</td>
<td>Deliverable Creation be facilitated</td>
<td>decision making be facilitated, discussion reach conclusion, foster discussion of differences, sensitive issues be surfaced</td>
</tr>
<tr>
<td></td>
<td>Discussions be facilitated</td>
<td>decision be reached, discussion be focused, emergent collaboration, knowledge is gained, summarization be simple</td>
</tr>
<tr>
<td>Participation be incentivized</td>
<td>Community be developed</td>
<td>build common ground, determining relevant conversation to join be simple, identify valuable participants, referring to others be simple, summarization be simple, tracking relevant messages be simple</td>
</tr>
<tr>
<td></td>
<td>Participation be broad and sustainable</td>
<td>catalyze discussion, content be engaging, contribution be acknowledged, legitimacy to respond improved, motivation to contribute be increased, user contribution be emphasized</td>
</tr>
<tr>
<td></td>
<td>Accessibility be rigorous</td>
<td>awareness of others’ activities be supported, effort be minimal</td>
</tr>
<tr>
<td>External information be considered</td>
<td>Information be managed effectively</td>
<td>collective knowledge representation be supported, increase content breadth, information sources be broad, relevant information be accessible, value of unique information maximized</td>
</tr>
<tr>
<td></td>
<td>Discussions be facilitated</td>
<td>tracking relevant messages be simple</td>
</tr>
<tr>
<td>Social pressures be reduced</td>
<td>Normative ideals be met</td>
<td>anxiety be reduced, divergent thinking be facilitated, diversity of opinion, neutrality, opinions be diverse, opposing views be expressed, reflection be encouraged, sensitive issues be surfaced</td>
</tr>
<tr>
<td></td>
<td>Discussions be facilitated</td>
<td>avoid evaluation apprehension, flame wars be avoided</td>
</tr>
<tr>
<td></td>
<td>Identity be manageable</td>
<td>build common ground, foster group identity</td>
</tr>
<tr>
<td></td>
<td>Group activity be facilitated</td>
<td>sustained interaction</td>
</tr>
</tbody>
</table>

Table 6.1: Goal associations and inclusions for the FaceBook DSS project
6.4 Impacted and Related Goal Analysis

Now that Jane has included many retrieved goals into her project, she can begin to assess the means through which those goals have been achieved in the literature, as well as what side effects those means have been found to have. She does this by examining a list of recommended design alternatives (the means), and the goals outside her current project scope those features also contribute to. These impacted and related goals have been found to be impacted by those design alternatives but have not been included in the project.

Jane quickly assesses the list of impacted and related goals, shown in Figure 6.1. She notes that there is a green list, an orange list and a red list. The green list displays positive side effects of the proposed design alternatives. She decides to skim through this list quickly. What she’s really interested in are the other two lists. She examines the orange list, the goals that receive mixed contributions from the proposed designs. She is worried about unforeseen negative side effects. One of the impacted and related goals, that has received contradictory contribution evidence, is document has diverse viewpoints; Jane hadn’t included that goal before, but she decides to examine the ‘additional information’ dialogue for that goal. She notes that the design encourage a summary of the (discussion) thread may hurt all viewpoints considered, which in turn will hurt document has diverse viewpoints. Jane decides that this goal is important, selects it, and associates it with Degree of consensus be obtained and Normative ideals be met.

Jane continues her examination of the lists, moving to the red, negatively impacted related goals. She sees that consensus be obtained is listed – how did she miss that one? – and so quickly selects and associates it with Degree of consensus be obtained and Deliverable creation be facilitated. She wonders why it is negatively impacted, and upon further investigation she finds that Dunne (2009) reports that threaded discussion forums’ reply
mechanisms do not encourage the formation of consensus.

Jane looks through the remainder of the negatively impacted related goals list and then moves on to the next analysis phase.

### 6.5 Project Model Configuration

The creation of a project model marks the beginning of the next major analysis phase. Jane enters in a name for her model, and then examines the hierarchical tree list that is presented to her. She notes the aggregate goal classes each may be ‘drilled down’ into; she clicks on Accessibility be rigorous and sees the two goals she included: awareness of other’s activities be
supported and effort be minimal. Both are orange in colour, which Jane knows means they receive contradictory contributions from other elements. She clicks on awareness of others’ […] and is presented with several contributing items, which she had previously not seen. She notices that each item is selected by default, and decides to unselect several (after reading their descriptions and deciding they were irrelevant), following the instructions presented at the top of the screen. Jane notices that as she deselected the nodes, the colour of awareness of others’ activities be supported changed. It is now green, satisficed\(^2\).

Jane continues this process for the other aggregate goals, noting that several items in the list appear deselected. She notices that she had previously deselected these nodes earlier in her analysis of the included awareness of others’ activities be supported. She thus realizes that deselecting a design or goal in one section of the model will propagate that same item’s selection status throughout the model.

Jane continues to configure and explore the model. She decides to explore a goal slice of foster group identity, because she is worried about the tradeoffs between social cues and pressures on information sharing. She therefore clicks on the Q7 icon and opens Open OME to assess the generated graph (shown in Figure 6.2).

### 6.6 Generated Reports and Models

Jane assesses the goal model derived from the ‘Q7’ file she downloaded. The section of the model discussed below is shown in detail in Figure 6.3 and is centred on the goal foster group identity. Jane looks for alternative solutions for that goal and the other modeled high-level goals. She notes that foster group identity receives contributions from awareness of shared representation and integrate into single document. These items, in turn, are

\(^2\)See Appendix A3 for a technical explanation of how this evaluative labeling is performed
helped by the design feature Collaborative statement writing. She notes that integrate into single document and another node, express arguments formally both help summarization be simple. Jane believes that this latter goal is essential to help build consensus, and decides that the two contributing nodes to this goal, integrate into single document and express arguments formally are mutually exclusive alternatives. She thus changes their contribution links to ‘OR’ and begins to assess the tradeoffs associated with
choosing either approach.

In doing so, Jane notes that if she were to incorporate the design feature Collaborative statement writing into her system, awareness of shared representation would help foster group identity, while integrate into single document would hurt document has diverse viewpoints. The alternative, express arguments formally, does not contribute to either of these goals, but Jane knows that because she is looking at a model slice, there are additional design features that contribute to the goal express arguments formally outside of the model. Nevertheless, she decides that summarization and group identity are both very important, and thus selects the design feature Collaborative statement writing, which contributes to both.

Following her analysis of the model slice, Jane reconfigures her project model to account for the above insights; she deselects express arguments formally. This process is repeated for several important model slices. Jane is then sufficiently satisfied with her analysis and design feature selections and thus she proceeds to generate a summary report of her analyses.

The report is in two sections, the first being a summary of suggested design features and the second being a list of publications to consider.

The summary of design features (shown in Figure 6.4) lists several candidate features that Jane has indicated to be relevant to the FaceBook DSS project. Beside each candidate is a list of the goals it contributes towards. Each goal is marked according to the contributions it receives from that design feature and its parents. With this list, Jane can look over candidate features and decide on a feasible system prototype configuration or select from a variety of off-the-shelf platforms, using this list as a heuristic aid.

The list of publications (shown in Figure 6.5) is sorted by the number of references the project model contains to concepts mentioned in each publication. The more references a publication has,
### Project Model Reports - "Jane"

<table>
<thead>
<tr>
<th>Related Design Alternatives</th>
<th>Description and Related Goals</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>anonymity</strong></td>
<td>user posting has no real-world identity associated with it (Lourenço, R.P. &amp; Costa, J.P., 2007)</td>
</tr>
<tr>
<td><strong>collaborative filtering</strong></td>
<td>filtering information based on other user’s evaluation of it. (Lev-on, A. &amp; Hardin, R., 2008)</td>
</tr>
<tr>
<td><strong>Collaborative statement writing</strong></td>
<td>Users work together to draft a commonly accepted document (Grönlund, K., Strandberg And, K. &amp; Himmelroos, S., 2009)</td>
</tr>
<tr>
<td><strong>Emphasize uniqueness of contribution</strong></td>
<td>&quot;Emphasize the unique value of each individual’s contribution&quot; (Antin, J., 2007)</td>
</tr>
<tr>
<td><strong>encourage quoting previous posts</strong></td>
<td>authority figure encourages users to reply to one another (Chen, D &amp; Wang, Y., 2009)</td>
</tr>
<tr>
<td><strong>implicit rating</strong></td>
<td>Assign points to messages based on their referal quantities (Chen, D.T. &amp; Hung, D., 2002)</td>
</tr>
</tbody>
</table>

#### Relevant Publications

<table>
<thead>
<tr>
<th>References</th>
<th>Publication</th>
<th>Selected Goals mentioned</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>Chen, D.T. &amp; Hung, D. (2002) Personalised knowledge representations: the missing half of online discussions. <em>British Journal of Educational Technology.</em></td>
<td>tracking relevant messages be simple, referring to others be simple, summarization be simple, determining relevant discussion to join be simple, collective knowledge representation be supported</td>
</tr>
<tr>
<td>4</td>
<td>Lev-on, A. &amp; Hardin, R. (2008) Internet-Based Collaborations and Their Political Significance. <em>Journal of Information Technology &amp; Politics.</em></td>
<td>flame wars be avoided, opinions be diverse, opposing views be expressed, relevant information be accessible</td>
</tr>
</tbody>
</table>

---

**Figure 6.4:** Excerpt of summary of relevant design features for the Facebook DSS project.

**Figure 6.5:** Excerpt of summary of relevant publications for the Facebook DSS project.

the higher it is ranked on the list. Jane, having access to scholarly databases, perceives that Chen and Hung (2002) is highly-ranked, and proceeds to simply click on the bibliographic citation in order to obtain a copy of Chen and Hung’s article from its publisher’s website.
6.7 Discussion

This process helped Jane to develop a prototype plan for her own ‘Facebook DSS’ project. Specifically, based on her analysis she designs her prototype to include the following features:

1. Collaborative statement writing
2. Anonymity
3. Threaded discussion
4. Mandate search for related ideas
5. Implicit rating
6. Collaborative filtering

Essentially, Jane envisions a social DSS that is centered around the development of a collaboratively written assessment of a problem. This is intended to help consensus be formed, as well as to encourage group identity (in comparison to the formal argument support she had considered as an alternative). Each user would be given a certain number of edits they can make to the document; thus they need to be careful about their activity (a less constrained, brainstorming process may occur in the threaded discussion areas, discussed below).

In order to leverage outside information, Jane proposes that users of the system be mandated to supply several links to relevant information, which will help to ensure a diversity of views (and help offset the homogenizing effects of collaborative statements). These links would be subject to a collaborative filter, through which users would rank and view the most highly-ranked content. Jane is somewhat unsure about this feature’s effect on the diversity of views, but wishes to have a strong mechanism for identifying good content. Thus, she tentatively includes the feature in her prototype design.

Each link to outside information has an attached threaded discussion forum in which participants may discuss the linked content’s relevance to the problem at hand. Based on the number of replies a posted comment elicits, a ranking is assigned to that posting user, who may
use those points to make additional edits in the collaborative written document. These comments may be made anonymously, or with their official Facebook identities. In either case, the system would track the actual posting account, and thus be capable of awarding implicit points to anonymous postings.

Jane is confident that while this prototype design will help move the project towards its objectives, user tests and further design iterations and reconsideration of some of the design decisions will need to be undertaken. Nevertheless, by engaging with the content of the design knowledge library, Jane has developed a clearer vision for her project team to explore.
Discussion of interview findings

This chapter begins with an analysis of the practitioner interviews conducted for this thesis. Participants were interviewed in order to assess whether the GO-DKL framework can support current online deliberation system design practices. First, we present an analysis of these practitioners’ general design practices; this includes their work environments, methodologies, and their use of design knowledge. Second, the practitioners’ interpretation, critique and suggestions regarding the GO-DKL framework and browser demonstration are outlined. We then conclude this chapter with a synthesis of those interview results, coupled with other lessons learned from the development of this framework.

7.1 Online deliberation system designer practices

The six practitioners interviewed can be loosely divided into two classes, the first being system developers. These individuals work on building their own online deliberation systems. The second class of practitioner can be termed as analysts; they select from a range of off-the-shelf products based on a system’s perceived alignment with an organization’s objectives. Two analysts
Table 7.1: Interview participants

<table>
<thead>
<tr>
<th>Pseudonym</th>
<th>Project</th>
<th>Class</th>
<th>Channel</th>
</tr>
</thead>
<tbody>
<tr>
<td>Anna</td>
<td>Immigrant Discussion Forum</td>
<td>Analyst</td>
<td>Face-to-face</td>
</tr>
<tr>
<td>David</td>
<td>Citizens’ Pros/Cons on proposals</td>
<td>Developer</td>
<td>Skype</td>
</tr>
<tr>
<td>Ellen</td>
<td>Municipal Transportation Dialogues</td>
<td>Analyst</td>
<td>Skype</td>
</tr>
<tr>
<td>Patrick</td>
<td>Group Decision Support System</td>
<td>Developer</td>
<td>Skype</td>
</tr>
<tr>
<td>James</td>
<td>Consensus-building platform</td>
<td>Developer</td>
<td>Skype</td>
</tr>
<tr>
<td>Bryan</td>
<td>Citizen-created issue campaigns</td>
<td>Developer</td>
<td>Skype</td>
</tr>
</tbody>
</table>

and four developers were interviewed. The participants generally worked individually or in small groups, and often worked with client organizations for product testing or non-profit endeavours. See Table 7.1 for a concise introduction to these practitioners.

7.1.1 Design Methods

A goal of the interviews was to obtain insight into how the framework could fit into practitioners’ existing work practices. To accomplish this, participants were asked about their design process, and whether they followed any formal methodologies. Five out of six participants can be said to practice iterative design; they either explicitly mentioned that “it works well for me to do iterative design” (Anna), they “iterate” based on user test results (Bryan), “get some feedback and then you know, iterate quickly” (Patrick), hinted at it through statements such as “refining, refining, refining” (James) or mentioned a reliance on “paper prototyping studies” (David). Pilot studies and/or user testing is a large part of these practitioners’ process; “we did a user study on election day” (David).
7.1.2 Design Knowledge

In order to determine the potential fit of this thesis’ framework with current practices and needs of online deliberation design practitioners, participants were asked to describe how they currently utilize design knowledge in the form of past experience, documents and publications to help them in their practice.

Several themes emerged from this line of questioning, a prominent one being a heavy reliance on tacit experience, or “intuitive insights” (Patrick). David provided a concise description of this practice: “It (his experience) is all over this soup. A soup of past experience” (David). This was interpreted to signify that David’s ‘soup’, his tacit experience, while informal, can be leveraged to guide his design process with relative efficacy. Patrick clarified his stance on this point, by stating that he can know what works and what doesn’t, “manually in (my) head [...] but I guess going forward we’d have to get a bit more systematic about these things.” Ellen clearly outlines a division between documentation and tacit experience: “most of the time, I draw on my own experience” rather than relying on documentation. These perspectives recall the findings of Ormerod et al. (1999), who found that designers usually work from memory.

While participants did not generally rely on documentation of past design experiences, the work environment often dictated the amount of documentation generated. Anna stated “I don’t like having” documentation, but “depending on the client”, she needs to have a record of her design decisions. This is reflected in a quote from Ellen:

Because I’m a sole consultant now, I carry it around in my head. [...] When I worked for Engagement Inc.¹, we had an ongoing knowledge development project: We’d have a debrief with each team that was assigned to each task, talk about what worked and didn’t, and circulate around to the wider staff. Plus the best practices were encoded and made available in a handbook. (Ellen)

¹Pseudonym
Another important theme found in the transcripts was the sources of design inspiration. Participants often mentioned drawing inspiration from design features of existing popular websites, and repurposing those designs in pursuit of their unique objectives. For example, David mentioned a module in FaceBook that enabled the quick cycling through ‘thumbnail’ photographs of “people you might know”, which inspired him to repurpose this concept by replacing photographs with short pro/con statements that could be similarly cycled through. Furthermore, David cited Wiki software as an inspiration, as did James: “I had this idea that I was gonna merge two applications: The forum software called php bb, and the wiki.” Bryan mentioned drawing inspiration from what worked on various e-government websites, such as “10 Downing Street”. James explicitly summarizes this perspective: “the things that inform me are not documentation about projects, but the projects themselves.” Existing tools were by far the most commonly cited source of inspiration for practitioners.

Being a part of a community of practice, or at the minimum, discussing with fellow practitioners and expert users was often mentioned. James clearly exemplifies this perspective: “It’s funny, I find it very difficult to find relevant information through Google sometimes. It’s mostly through other people.” Echoing this appreciation of the knowledge of others, Patrick states that “we’ve talked to a lot of people who’ve done this stuff for a living, we’ve talked about what their gripes are about existing tools, what they wish for, better tools.” Obtaining opinions regarding the utility of various tools and approaches from the larger design community is important to these practitioners.

### 7.1.3 Design Issues

Many of the issues faced by practitioners in their design work also appear as concepts in the knowledge base. For instance, David mentioned a goal of his project was to help participants to “identify common ground”. This echoes the knowledge base relationship build common ground
Discussion of interview findings

is helped by shared background be emphasized, which in turn is helped by locate users on map (from Gurzick and Lutters, 2009). Another example was mentioned by Patrick: “Another big discussion is anonymity and using real names”. The knowledge base has quite a few relationships involving anonymity, such as anonymity hurts message be taken seriously, which helps opinion be considered (from Rose and Sæbø, 2010). These issues were discussed by participants prior to the demonstration of the framework prototype, and thus they were derived from their own experience.

Participants were also asked to complete a short survey question that involved the ranking of generalized design issue categories (see Figure 7.1). Some of the categories were derived from the first complete iteration of the framework’s aggregate goals (as described in chapter 5, though the phrasing was modified for clarity), while others were obtained from a recent blog posting by a prominent online deliberation group, the National Academy of Public Administration’s Tools for online idea generation (NAPA, 2011). This mix was intended to measure how relevant the aggregate goals developed in the research are to practitioners. Bold terms in Figure 7.1 below are aggregate goals developed inductively as mentioned in Chapter 3. The top four, and eight out of the top ten most relevant issue categories to participants’ design practice are based on the framework’s aggregate goals. This result serves to validate the inductive categorization of goals from the literature as being relevant to design practice.

An interesting pattern was present in the accounts of various practitioners as they discussed design issues they had faced in their practice. They reported an evolution of their understanding of the domain, and how designs might interact with mutably-defined goals. Anna notes that “it’s like we’re making discoveries. We found out that being anonymous online is being destructive. But that wasn’t really known when it was first started.” Similarly, David states that “when I first got into this (field), I was all about consensus. [...] I feel we’re [now] at the point where we can’t even talk about consensus on the larger issues.” Interestingly, problems appear to become more
complex as a designer learns more about the domain: “I had actually quite a clear vision as to what I was trying to achieve at the beginning of the project, and it’s become a lot more murky now” (James). The interviewed designers appear to approach a problem with a clear vision of what needs to be done, but as they begin to build and test their approach, note the various complexities that emerge and problematize their early design decisions.

### 7.2 Practitioner perspectives of the framework

This section begins with a summary of interview participants’ statements regarding the relevance of the knowledge base’s contents and the GO-DKL analysis procedure to their practice. It then presents their critiques and suggestions to improve the framework.
7.2.1 Relevance of knowledge base contents

During the demonstration of the framework prototype tool, participants found many items in the knowledge base to be highly relevant to their current projects. As they scrolled through the lists of library goals retrieved for their project, exclamations included “we actually talked about that [issue] yesterday” (Anna), “that’s the reason” we have that process (David), “yeah, that’s [several retrieved concepts] right on, totally that would work” (Ellen), a related project team “did everything that’s on this page” (Patrick), and “all these problems [I’ve] actually had to face” (James). Additionally, Bryan notes that he wasn’t “aware there was such academic literature on the topic; it’d be interesting to read more.” It should be noted, however, that many other concepts were not deemed relevant to their particular project. This is not unexpected, as the diverse array of content in the knowledge base would not reasonably be applicable in every situation. Furthermore, these statements attest to the relevance of the codified literature, and provide some validation to the inclusion and exclusion criteria that governed the selection of publications (see chapter 3 for more details).

7.2.2 Relevance of analysis procedure

The most commonly expressed statement regarding the relevance of the analysis procedure essentially claims that the process helps designers learn of issues they hadn’t previously considered. For example, Anna stated that “it would help me go....did I touch all those in my work with my client? Did I miss anything?” David claimed that the lists of retrieved library goals function like “heuristics” that would be useful “almost midway through” a project, and echoed the latter part of Anna’s statement: “[With this procedure] I can be like ‘am I missing anything? Have I not considered anything that I should have considered?’” Bryan noted that the framework would “give us new ideas.” Somewhat differently, James’ said he would use the framework at the start of
his design process, as it would help identify “potential problems that you’re gonna come across in your project.” This latter assessment still speaks towards the identification of new issues, but is more focused on problems rather than potential solutions.

James was quite interested in the impacted and related goals feature, as well as the colour coding of nodes based on the value of their received contributions. As mentioned in preceding sections, a green label would indicate that the item received net positive contributions from other items. However, James stated that “I wasn’t really looking at the green bits, I was looking at the red and yellows. I know I had problems with those.” He felt as though seeing the potential problems was more important than the potential benefits.

Another notable interpretation of the analysis procedure focuses on the simple act of explicitly stating and categorizing one’s project objectives, which is thought to afford a clearer understanding of the project as a whole. Anna notes that the process helps her to think abstractly about what is it that I’m doing”, while David notes that the process is similar to business consultants asking stakeholders to describe their work processes: “It might help you formalize as you have a better sense of what your project is. [...] Just by writing it out, you can get a better understand of what you’re working on.” This perspective may serve as some validation towards the analytical affordances of the procedure itself.

In terms of relevance to a wider community of practitioners, Patrick recalled a story of a group discussion at a conference where participants wished for “a [software] wizard where you could just input your goals, what are you trying to achieve, and your other parameters, and it will give you the tool recommendation.” However, he continues “you can’t really say (which tools are appropriate), because it’s in my view it’s still more important that you’ve got your process right, and only then does the tool have an impact”. However, he concludes by remarking about this framework: “But this is a level deeper, if you tie it [one’s project] to go via [...] goals, [to] design choices, and then you could maybe one level below [...] start mapping tools to those [design
features].” This brings us to the suggested improvements to the framework that participants discussed.

7.2.3 Critiques and suggestions

As they worked through the process, several participants were unsure of the exact purpose of the framework and what it would provide for them. Therefore, they suggested that user’s position in the process be made clearer, and that the potential output of the process be made clearer from the beginning. Ellen criticized the time it takes for the designer to reach a recommendation, noting that “most people [...] [are] gonna want to get to your list of recommendations as fast as you can.” (Ellen)

While this may be interpreted as both a critique and a testament to the relevance of the knowledge base contents, Ellen and Anna both requested some form of report as an output of the process. At the time of the interviews, the project model was the only output. This was expanded with the introduction of two other reporting mechanisms: the ranked bibliography and the list of design feature candidates (as discussed in Chapter 5).

Another common suggestion was to permit design practitioners to contribute to the knowledge base. This would include new goals, design features, relationships and categorizations. The scholarly contents of the knowledge base would therefore be stored alongside experiential knowledge.

There were many suggestions and critiques regarding the phrasing of the knowledge base elements. The passive voice of the goal phrasing was unappealing to Anna and Ellen. The terminology of the various goal types was confusing to Ellen. Ellen suggested that being faithful to the wording of the literature was less important than a clear and consistent vocabulary, an attribute that Patrick and Bryan also desired. Additionally, many element names (goals and design
features) were ambiguous to the participants. However, the additional information with a
description often helped to mitigate this confusion.

The level of trust participants placed in the knowledge base were revealed in several interesting
insights. Patrick generally noted that users had to “trusting (sic) the resource there [...] but over
time, people could have feedback and deal better with that, made collaboratively.” The latter
phrase hints at Patrick’s suggestion of users being able to contribute to the knowledge base and
assign confidence measures to various items (such as rating or recommendation mechanisms).
David provided a very insightful interpretation of the trust issue:

I don’t have concerns of trustworthiness of this, ‘cause I wouldn’t be looking at this
for trust. it might be different from what your [the researcher’s] goal is. I wouldn’t be
looking at this as an authoritative guide, like “this will help me make sense of what is
out there, and how people have framed their research and, what’s a kind of a way of
navigating some of their claims’, not necessarily that their claims are true. Even if
they have empirical data to back it up, like it’s something for me to look into”. So,
I’m not actually concerned with trustworthiness.

This statement appears complementary to the participants’ general perception of the knowledge
base and analysis procedure as more of a ‘checklist’ that needs to be qualitatively examined and
contextualized, rather than as an objective record of universally applicable knowledge. This
furthermore aligns with David’s earlier statement that he was “leery of having a static knowledge
representation” and his view that “even if you’re not able to achieve like empirical facts, I think
that what you’re doing is moving towards knowledge.” Prior to this statement, David was
discussing his project as more of a ‘data point’ than a record of objective knowledge. Indeed, the
knowledge base is a collection of relationships extracted from the literature, that were highly
situated. As such, these ‘data points’ should be seen as starting points to a more contextualized
design approach, rather than as reusable ‘chunks’ of knowledge.
7.3 Summary of interview findings

Overall, participants found the knowledge base contents (including the aggregate goals) to be relevant to their own design practice. It serves as a valuable reference checklist or heuristic for designers to assess potential benefits or problems throughout their design process. Furthermore, participants felt that the act of defining and associating one’s initial project goals with library items was found to help one clearly understand one’s own project. However, designers did not explicitly trust the relationships presented in the knowledge base, which lends support to the notion that the framework acts more as a ‘springboard’ for situated design decisions.

Due to time constraints, a robust evaluation of the specific components of the analytic processes (specifically, the tree-list and goal graph representations of the project model) was not achieved. Initial impressions of these representational devices range from useful to time-consuming. Due to the variety of representations and reporting mechanisms supported by the framework, designers may arguably have some flexibility in accommodating their own design processes.
Conclusions

The preceding chapters of this thesis have outlined the motivations, related work, methodology, iterative development, usage guide, application example, and the evaluation of the GO-DKL framework. This final chapter concludes the thesis with a discussion of the framework’s perceived strengths, contributions to the literature, limitations, as well as a discussion of possibilities for future work.

8.1 Strengths and contributions

Based on the assessment of interview participants working in the application domain, this research has developed a working model for structuring online deliberation design knowledge in a manner that supports its analysis. Furthermore, the developed analysis procedure centered on the goal-directed association and retrieval of knowledge base goals relevant to one’s own design project has been found to reasonably support designer activity. The GO-DKL framework thus supports a functional process for organizing means-ends knowledge.
8 Conclusions

The framework itself, composed of the methods for structuring and analysing online deliberation design knowledge, can potentially contribute to that academic field. A publicly accessible, relatively complete and reliable population of the knowledge base from relevant publications may help scholars and practitioners in this field to better synthesize past findings and address new problems and challenges, building off one another’s work.

The tree-list method for representing goal-oriented models is a unique contribution to the goal modeling field\(^1\). This representation, if studied more rigorously, might help model analysts to usably examine and assess complex domain models. Indeed, it may facilitate a focused analysis of certain relationships within a goal graph, in an attempt to mitigate the daunting task of assessing complex goal graphs with many intersecting relationships (Pastor et al., 2011). The automated generation of model slices might also afford a closer look at multiple facets of the same data set which might yield additional insights.

8.2 Limitations

One of the major limitations of this study is the fact that I coded documents without support from other coders. This limits the reliability of both the coding scheme, process, and extraction of concepts from the literature. The iterative development of the coding scheme was an attempt to mitigate this issue, though the limitation remains. A related limitation is that publications often discussed design features, and goals at different levels of abstraction. Thus, the definitions of meta concepts such as ‘design feature’ and ‘goal’ had to be flexible in order to accommodate this variability. This recalls the interview participants’ concerns regarding the unclear vocabulary of many knowledge base items. Again, had other researchers helped to code documents, a more consistent application and definition of the concepts might have been achieved.

\(^1\)A short technical paper explaining the GO-DKL browser has been published in the forthcoming *i* tools fair workshop (Hilts and Yu, 2011).
8.3 Future Work

An exciting avenue for future work is the expansion of the framework to support an online community of designers. These designers would be able to contribute their own claims and relationships, and comment on the knowledge base contents, recalling Eilouti’s (2009) analysis-synthesis design knowledge cycle. This would help validate the knowledge base contents, as designers could perceive how their peers evaluate certain design features, goals, or relationships. The above-mentioned reliance on intuitive relationships could be minimized, should the community begin to replace them with observed relationships from concrete cases. In sum, these community-based activities appear likely to help develop a broad and in-depth repository.
Another direction is to examine the utility of a \textit{bottom-up} approach to the framework. In this thesis, a goal-directed, \textit{top-down} process was defined. Such an approach – starting from existing design features, and working up towards impacted goals – might help accommodate a more diverse range of design practices.

A related venture would be to include publicly-accessible, pre-built project models that represent well-known systems. For example, a designer might examine the relationships between Twitter’s design features and the various goals of that system. The designer may then build off that ‘Twitter template’ and repurpose the model to suit his or her own scenario.

Finally, rigorous user testing of the analysis process, especially focusing on the tree-list representation of the project model, will likely help to strengthen the overall utility of the framework and its tool support. A systematic means of controlling the vocabulary of the knowledge base items would likely reduce the time required to make sense of items displayed in the tree-list. Furthermore, it would be valuable for a model slice exported from the tree-list, that was reconfigured in Open OME, to be able to be imported back into the design knowledge library, so that a designer may continue to explore the updated tree-list of his or her project.

These endeavours should help the framework develop into a more robust and comprehensive design support tool that effectively helps a community of practitioners build on one another’s ideas.
Appendices

A1 The Knowledge Base

GO-DKL version 3 features a knowledge base containing extracted claims, relationships, and concepts from 29 publications. The following is a list of the publications therein; please see their respective entries in the bibliography for further information.

1. Afshar et al. (2009)
5. Chen (2009)
7. Cohen et al. (2000)
10. Grönlund et al. (2009)
13. Introne (2009)
15. Lampe and Johnston (2005)
A2 MySQL records to tree-list representation

The below outlines the basic means by which retrieved rows from a database containing the following fields are converted into the tree-list representation described in chapter 5.

A query retrieves a project model, consisting of rows of the following fields:
AggregateGoal, IncludedGoal, Relationship1, IncludedGoal2, Relationship2, DesignFeature.

The records are sorted by each field, starting with the leftmost – AggregateGoal, until the rightmost – DesignFeature.

In this manner, an algorithm can easily loop through the returned records and construct a hierarchical list. First, create root list items for each distinct\(^2\) AggregateGoal, then create distinct

\(^2\)Distinct among its branch siblings. EG: one Library Goal may appear as the child of multiple Aggregate Goals.
Child nodes for each LibraryGoal and so on.

In order to be converted into a tree-list as implemented in GO-DKL browser version 3, the output of the above algorithm needs to be an unordered HTML list. EG:

```html
<ul>
  <li>Aggregate Goal 1
      <ul>
        <li>Library Goal 1
            <ul>
              <li>Library Goal 3</li>
            </ul>
        </li>
        <li>Library Goal 4
            <ul>
              <li>Design Feature 1</li>
              <li>Design Feature 2</li>
            </ul>
        </li>
      </ul>
  </li>
  <li>Aggregate Goal 2
      <ul>
        <li>Library Goal 1
            <ul>
              <li>Library Goal 6</li>
            </ul>
        </li>
        <li>Library Goal 3
            <ul>
              <li>Design Feature 5</li>
            </ul>
        </li>
      </ul>
  </li>
</ul>
```

This list is then assigned a method from the Treeview jQuery plugin, available at http://jquery.bassistance.de/treeview/. EG:

```javascript
$("#unordered-list").treeview();
```

An open source JavaScript library
In GO-DKL browser version 3, while the unordered list structure is as above, additional data is added to the list to provide the feature set described. For example, HTML form elements are present within child list items. These forms carry a unique ID\(^4\) that is a string consisting of the parent element id and the child element id (eg: a Library Goal’s id). The default value of these form elements is the contribution type between parent and child element. For example, if Goal1 contributes a ‘help’ value to Goal2, the form element within Goal1’s list item would bear the id Goal1.Goal2, and have the default value of helps. These form elements feature a dropdown list of many contribution values featured in the \(i^*\) syntax. Therefore, by selecting a different value for the form element, the user may change the contribution value between Goal1 and Goal2.

### A3 Tree-list evaluation label propagation

The below evaluation label propagation algorithm was designed to parse the same type of hierarchical unordered HTML list as in the above section. It too, is dependent on the jQuery JavaScript library. The process is a simplification and automation of the qualitative rules described in Horkoff and Yu (2009) and explained in Chapter 2.

At a conceptual level, the process consists of looping through every leaf node\(^5\) in the list. At each leaf node, the `treePropagate` function is called. This function compiles the contribution values of all sibling leaf nodes and passes that compiled value to the parent element. `treePropagate` is then recursively called, applying to the parent elements’ parent node, and so on. After a branch is complete, the initial loop continues to the next branch leaf node sibling set, where `treePropagate` is called again.

Contribution values are assigned an integer based on the strength of the contribution. For

---

\(^4\) Across the entire list  
\(^5\) Lowest level list item, a design feature
example, helps is worth 1 point, while breaks is -2. The sum of all sibling contribution values is divided by the number of contribution values, resulting in a satisfaction ratio. If this ratio is above 30, the parent element is assigned a green label. If it is below -30, the parent is assigned a red label. In between those two, a yellow label is assigned. The contribution value of each leaf node is the currently selected value of the form element present in that node.

The above evaluation process is called once when the list is loaded, and is repeated when any element in the list changes in value. In this manner, the evaluation labels of all list elements are responsive to user interaction.

The key point of interaction is the alteration of contribution values between a child and parent. A user may change the HTML form element's value by selecting a new one from the form element's list. This change is applied to every element in the list with the same unique id string. To accomplish this, the system searches through the DOM (document object model) to find the matching elements. In this manner, a relationship between design feature and library goal (for example) present in multiple branches of the list will all maintain the same value.

The same process is undertaken when a user selects or deselects a list item; matching items throughout the list will correspondingly be selected or deselected.

**A4 MySQL records to Q7 file**

The process by which a model slice is converted into a Q7 file capable of being opened and converted to a goal model in Open OME begins similar to the tree-list construction process. The same general query retrieves the same kind of records. To reiterate, those would be rows of the

---

6 However, recursive calls of `treePropagate` take into account the label assigned to child nodes (propagated from descendants); the value of the child's form element is multiplied by (-)1, depending on the child's label. For example, a denied child that would normally pass a negative value to its parent would pass a positive value.

7 See Yu et al. (2007) for more detail on the Q7 language and syntax.
following fields: AggregateGoal, IncludedGoal, Relationship1, IncludedGoal2, Relationship2, DesignFeature. These records are sorted by each field, starting with the leftmost – AggregateGoal, until the rightmost – DesignFeature.

A program loops through the records, creating arrays for retrieved Aggregate Goals, Library Goals, Design Features, and relationships.

Following this, the unique Aggregate Goals array is looped through, adding entries to an array called AggregateGoals_LibraryGoals. An item in that array is created for each relationship between Aggregate Goals and Library Goals. For example, the string stored in an array entry for more users contribute to more topics helps participation would be structured as follows:

```

May more users contribute to more topics => + participation
```

In this example, the text may is used to tell Open OME’s Q7 parser that the next part of the string preceding the operator is a soft goal. The operator => + signifies that the preceding section of the string helps the latter section.

This process is performed for each unique array set (besides relationships). The string for design features contributing to goals begins with Do (instead of May) to delineate it as a design feature.

Following the loops through the arrays which build the unique relationship arrays, each relationship array is exploded into a string, separated by newlines. These strings\(^8\) are then concatenated into one large string that is outputted as a Q7 file.

\(^8\)Another string is created that begins with the text ‘‘Placeholder { ;’’ and ends with }. In between those elements is the concatenated string of all relationship arrays. Essentially, this process is a workaround that generated a placeholder actor that contains all relationships in the model. It was developed to circumvent Open OME’s parsing of the Q7 file soft goals as both soft goals and distinct actors.


Clift, S. (2011). #pewi Majority see Internet increasing extreme views over giving ordinary citizens a voice - PewInternet.org.


