

# INFOCRYSTAL :

## A Visual Tool For Information Retrieval

by

Anselm Spoerri

Submitted to the Department of Civil and Environmental Engineering  
in Partial Fulfillment of the Requirements  
for the Interdepartmental Degree of

DOCTOR OF PHILOSOPHY  
in Information Visualization  
at the

Massachusetts Institute of Technology

February 1995

© 1995 Massachusetts Institute of Technology.  
All rights reserved.

Signature of Author.....

Certified by.....

Richard Marcus, Principal Research Scientist  
MIT Laboratory for Information and Decisions Systems  
Thesis Supervisor

Certified by.....

Steven Lerman, Professor of Civil and Environmental Engineering  
Department of Civil and Environmental Engineering  
Thesis Committee Chairman

Accepted by.....

Joseph M. Sussman,  
Chairman, Departmental Committee on Graduate Studies  
Department of Civil and Environmental Engineering

Barber King



# INFOCRYSTAL :

## A Visual Tool For Information Retrieval

by

Anselm Spoerri

Submitted to the Department of Civil and Environmental Engineering  
on January 20, 1995 in partial fulfillment of the requirements  
for the Interdepartmental Degree of Doctor of Philosophy  
in Information Visualization

### ABSTRACT

The InfoCrystal™ is a novel representation that uses a simple visual metaphor to help users deal with some of the complexities inherent in information retrieval. As a visualization tool, it can display all the possible binary as well as continuous relationships among N concepts. As a visual query language, the InfoCrystal enables users to formulate both Boolean and vector space queries graphically. Hence, it provides a visual framework that unifies the complementary Boolean and Partial Matching approaches and allows users to take advantage of their respective strengths. The InfoCrystal acts like a Boolean Calculator and users can use it to employ the expressive power of the Boolean retrieval approach and its broadening / narrowing techniques in a visual way. Further, users can assign relevance weights to the concepts and formulate weighted queries by interacting with a threshold slider. The InfoCrystal offers the added advantage that users can control in a visual way how to translate weighted queries into Boolean queries. Finally, arbitrarily complex queries can be created by using the InfoCrystals as building blocks and organizing them in a hierarchical structure.

A user study tested a specific aspect of the InfoCrystal interface by comparing it with a standard Boolean retrieval interface. Although this study did not test all the valuable features of the InfoCrystal, it produced the following useful results: 1) It showed that novice users, who received only a short tutorial, could successfully use the novel InfoCrystal interface. 2) The study showed that the InfoCrystal, even at an early stage of development, performed as well as the familiar Boolean interface, although the study was biased in favor of the Boolean mode. 3) The user feedback concerning the InfoCrystal interface was very encouraging and it helped to pinpoint possible improvements.

The InfoCrystal has broad applications because it offers a "visual machinery" to compare and relate any number of ordinary or fuzzy sets of arbitrary data items. It opens up new possibilities for complex data

explorations. The InfoCrystal enables users to integrate and explore information retrieved by different methods or from different sources in a flexible, dynamic and interactive way.

Thesis Supervisor: Richard Marcus  
Title: Principal Research Scientist  
MIT Laboratory for Information and Decisions Systems

Thesis Chairman: Steven Lerman  
Title: Professor of Civil and Environmental Engineering

Thesis Advisor: Paul Resnick  
Title: Assistant Professor of Management

Thesis Advisor: Ronald Mac Neil  
Title: Principal Research Associate  
Visible Language Workshop, MIT Media Lab

InfoCrystal™ is a trademark of the Massachusetts Institute of Technology.  
This thesis describes research done at the Massachusetts Institute of Technology within the Center for Educational Computing Initiatives. Support for the research described in this thesis has been generously provided by the UBILAB of the Union Bank of Switzerland.

# ACKNOWLEDGMENTS

The completion of this thesis has been a long journey and could have not been accomplished with the generous help and support of many people.

I would like to especially thank the following people:

- Richard Marcus, my thesis supervisor, for his guidance and unconditional support that has its own magic. He helped me to ground my research efforts more firmly by giving me a better foundation in the field of information retrieval.
  - Steve Lerman for providing me with an intellectual & academic home at MIT and guiding me in the resolution of many strategic and practical issues.
  - Paul Resnick for asking tough questions and for being a friend over the years.
  - Ron MacNeil for his gentle encouragement.
  - Thomas Sheridan, Jeremy Wolfe and the late Muriel Cooper for being there in the initial stages of this thesis.
  - Whitman Richards, without whom I would not have had the opportunity to study at MIT in the first place. I thank him for his continued belief in me.
  - Rudolf Marty from the Union Bank of Switzerland for providing me so generously with the financial support needed to conduct the research for this thesis. I am especially grateful to him for his willingness to take a risk and to pledge to support me at the very beginning when my research effort had not crystallized yet.
  - Hans-Peter Frei from the UBILAB of the Union Bank of Switzerland for deciding to continue to support my research efforts and for his constructive and valuable feedback on the content of the thesis work.
  - Elka Spoerri for her love and dedicated support over so many years: "Elka, we have made it - the mission is completed. I THANK YOU !"
  - Pamela Robertson-Pearce for having journeyed with me for all this time. I thank her for her love & emotional support, for encouraging me to do what I really cared for and helping me to express my artistic creativity.
-

- Noah Fisher and Joel Fisher for their support.
  - David Charron and Heather Mapstone from the MIT Licensing Office for helping me coin the name *InfoCrystal* for what I had created.
  - Neil Strickland from the Mathematics Dept. at MIT and the staff at the MIT Libraries for their help to locate literature showing how Venn diagrams with N sets could be generated.
  - Lynne Bolduc for to being there for me in the last two years while I was giving birth to the InfoCrystal.
  - The members of the Center for Educational Computing Initiatives (CECI) for giving generously of their time for the user studies and for creating a friendly working environment.
  - Chris King and Peter Yao, who helped with the implementation of the InfoCrystal software as part of the Undergraduate Research Opportunities Program at MIT.
  - All the people who have appeared in my life in the last year to enrich it and who have acted as catalysts in my transformation.
  - Carol Evans for guiding me through the emotional rapids I had to move through and for helping me to face and realize myself.
  - Last, but definitely not least, my father Theodor Spoerri, who unfortunately died much too young for both of us and whom I miss terribly: "Thedi, this Ph.D. and its thesis are a tribute to you and the immense loss I feel. I had entered academia in search of you, only to realize that nobody can take your place and all that is left for me to do is to learn by myself how to express the fire within."
-

---

# TABLE OF CONTENTS

<b>1 Introduction</b> .....	<b>17</b>
1.1 Information Visualization .....	18
1.2 Information Retrieval .....	20
1.3 Goal of the Thesis .....	21
1.4 Thesis Organization .....	22
1.5 Concrete Example .....	22
<b>2 Information Retrieval Models</b> .....	<b>25</b>
2.1 Introduction .....	25
2.2 General Model of Information Retrieval .....	25
2.3 Major Information Retrieval Models .....	29
2.3.1 Boolean Retrieval .....	29
2.3.1.1 Standard Boolean .....	30
2.3.1.2 Narrowing and Broadening Techniques .....	32
2.3.1.3 Smart Boolean .....	34
2.3.1.4 Extended Boolean Models .....	36
2.3.2 Statistical Model .....	38
2.3.2.1 Vector Space Model .....	38
2.3.2.2 Probabilistic Model .....	39
2.3.2.3 Latent Semantic Indexing .....	42
2.3.2.4 Document Clustering.....	43
2.3.3 Linguistic and Knowledge-based Approaches .....	44
2.3.3.1 DR-LINK Retrieval System .....	44
2.4 Conclusion.....	46
<b>3 InfoCrystal</b> .....	<b>49</b>
3.1 Introduction .....	49
3.2 2D versus 3D Visualization .....	49
3.3 Visualizing Relationships .....	50
3.4 Rank Layout Algorithm .....	53
3.5 Example Revisited .....	65

---

3.6	The Design Process of the InfoCrystal .....	67
3.6.1	The First Designs for the InfoCrystal .....	67
3.6.2	InfoCrystal Networks .....	69
3.6.3	Combining the InfoCrystal with Venn Diagrams .....	70
<b>4</b>	<b>Visualizing Boolean Queries .....</b>	<b>75</b>
4.1	Introduction .....	75
4.2	Query Space Visualized by the InfoCrystal .....	77
4.2.1	Ways of Specifying a Boolean Query .....	78
4.2.2	InfoCrystal as a Boolean Calculator .....	78
4.3	Creating Complex and Nested Queries .....	80
4.4	The Outliner Tool .....	86
4.5	Narrowing and Broadening Techniques .....	87
<b>5</b>	<b>Visualizing Weighted Queries .....</b>	<b>93</b>
5.1	Introduction .....	93
5.2	Formulating Weighted Queries using the InfoCrystal .....	94
5.3	The Bull's-Eye Layout .....	98
5.4	The Expressive Limits of Weighted Queries .....	101
5.5	Possible Alternative for Specifying Weighted Queries .....	103
5.6	Discussion .....	104
<b>6</b>	<b>Visualizing Vector Space Queries .....</b>	<b>105</b>
6.1	Introduction .....	105
6.2	Visualizing Any Ranking Function .....	107
6.3	The Continuous Bull's-Eye Mapping .....	112
6.4	Discussion .....	115
<b>7</b>	<b>InfoCrystal Software .....</b>	<b>117</b>
7.1	Introduction .....	117
7.2	InfoCrystal Software in Pictures .....	120
7.3	How to Drop an Input from an InfoCrystal ? .....	134
7.4	How to Add an Input to an InfoCrystal ? .....	134
7.5	How to Update the Selection Pattern in a Modified InfoCrystal ? .....	135
<b>8</b>	<b>Experimental Evaluation .....</b>	<b>137</b>
8.1	Introduction .....	137
8.2	Experimental Design .....	138

---



---

8.3 Experimental Analysis .....	147
8.3.1 Paired-Difference T-Test .....	149
8.3.2 Analysis of Variance .....	149
8.4 Analysis of the Experimental Results.....	153
8.4.1 Results for the Recognition Task.....	153
8.4.1.1 Categorical Paired-Difference Scores .....	154
8.4.1.2 Time Measurements.....	156
8.4.1.3 Discussion .....	158
8.4.2 Results for the Generation Task .....	161
8.4.2.1 Generation Task Biased in Favor of Boolean Query Language .....	161
8.4.2.2 Categorical Paired-Difference Scores .....	163
8.4.2.3 Continuous Paired-Difference Scores .....	166
8.4.2.4 Time Measurements.....	168
8.4.2.5 Discussion .....	171
8.5 Lessons Learned and General Discussion .....	174
8.5.1 Difference Between the Two Query Languages .....	175
8.5.2 Conclusion.....	176
<b>9 User Feedback .....</b>	<b>179</b>
<b>10 Relevant Research .....</b>	<b>185</b>
10.1 Overview Maps .....	185
10.2 Visualizing Hierarchical Structures.....	192
10.3 Familiar Metaphors for Accessing Information.....	195
10.4 Visual Query Languages .....	197
<b>11 Applications .....</b>	<b>203</b>
<b>12 Future Research .....</b>	<b>213</b>
<b>13 Conclusion .....</b>	<b>219</b>
<b>14 Epilogue .....</b>	<b>223</b>
<b>Bibliography .....</b>	<b>227</b>
<b>Appendix: Tutorial.....</b>	<b>233</b>

---



---

# LIST OF FIGURES

<b>Figure 2.1</b> General Model of the Information Retrieval Process .....	27
<b>Figure 2.2</b> Ways to Broaden or Narrow a Boolean Query .....	33
<b>Figure 3.1</b> Transforming a Venn diagram into the InfoCrystal .....	51
<b>Figure 3.2</b> InfoCrystal with THREE inputs, emphasizing qualitative information .....	56
<b>Figure 3.3</b> InfoCrystal with THREE inputs, emphasizing quantitative information .....	57
<b>Figure 3.4</b> InfoCrystal with FOUR inputs .....	58
<b>Figure 3.5</b> InfoCrystal with FIVE inputs .....	59
<b>Figure 3.6</b> InfoCrystal with SIX inputs .....	60
<b>Figure 3.7</b> InfoCrystal with SEVEN inputs .....	61
<b>Figure 3.8</b> InfoCrystal with EIGHT inputs .....	62
<b>Figure 3.9</b> InfoCrystal with NINE inputs .....	63
<b>Figure 3.10</b> InfoCrystal with THIRTEEN inputs .....	64
<b>Figure 3.11</b> Example Revisited .....	65
<b>Figure 3.12</b> The First Designs of the InfoCrystal .....	68
<b>Figure 3.13</b> InfoCrystal Network with three inputs .....	69
<b>Figure 3.14</b> InfoCrystal Network with four inputs .....	69
<b>Figure 3.15</b> InfoCrystal Network with five inputs .....	70
<b>Figure 3.16</b> InfoCrystal with three inputs combined with Venn diagrams .....	72
<b>Figure 3.17</b> InfoCrystal with four inputs combined with Venn diagrams .....	72
<b>Figure 3.18</b> InfoCrystal with five inputs combined with Venn diagrams .....	73
<b>Figure 4.1</b> Visualizing Boolean Relationships in the InfoCrystal .....	76

---

<b>Figure 4.2</b>	Using the InfoCrystal as a Boolean Calculator .....	79
<b>Figure 4.3</b>	Visualizing a Complex and Hierarchical InfoCrystal structure (1) ..	81
<b>Figure 4.4</b>	Visualizing a Complex and Hierarchical InfoCrystal structure (2) ..	82
<b>Figure 4.5</b>	InfoCrystal Tree Structure .....	83
<b>Figure 4.6</b>	The Query Outliner .....	86
<b>Figure 4.7</b>	Ways to Broaden or Narrow a Boolean Query .....	88
<b>Figure 4.8</b>	Visualizing Stemming, Field Level, and Proximity Constraints .....	89
<b>Figure 4.9</b>	Visualizing Coordination, Stemming, Field Level, and Proximity Constraints .....	90
<b>Figure 4.10</b>	Visualizing Boolean Coordination Using an Area-Based Measure .....	91
<b>Figure 5.1</b>	Visualizing Weighted Queries (1) .....	95
<b>Figure 5.2</b>	Visualizing Weighted Queries (2) .....	95
<b>Figure 5.3</b>	Bull's-eye Layout .....	99
<b>Figure 5.4</b>	How to Compute the Center of Mass .....	99
<b>Figure 5.5</b>	How to Compute the Bull's-eye Layout .....	100
<b>Figure 5.6</b>	Expressive Limits of Weighted Queries .....	102
<b>Figure 6.1</b>	How to Compute the Center of Mass (continuous version) .....	106
<b>Figure 6.2</b>	Relationship between the discrete and continuous versions of the InfoCrystal (1) .....	108
<b>Figure 6.3</b>	Relationship between the discrete and continuous versions of the InfoCrystal (2) .....	108
<b>Figure 6.4</b>	InfoCrystal Visualizing Continuous Relationships (1) .....	109
<b>Figure 6.5</b>	Clustering of Documents Satisfying Boolean Constituents (1) .....	109
<b>Figure 6.6</b>	InfoCrystal Visualizing Continuous Relationships (2) .....	110
<b>Figure 6.7</b>	Clustering of Documents Satisfying Boolean Constituents (2) .....	110
<b>Figure 6.8</b>	InfoCrystal Visualizing Continuous Relationships (3) .....	111

---

---

<b>Figure 6.9</b> InfoCrystal Visualizing Continuous Relationships (4) .....	111
<b>Figure 7.1</b> How to get started? .....	121
<b>Figure 7.2</b> State-Sheet of an InfoCrystal.....	121
<b>Figure 7.3</b> The Query Outline Visualized as an InfoCrystal .....	122
<b>Figure 7.4</b> Displaying the InfoCrystal One Level Deep .....	123
<b>Figure 7.5</b> Descending in the Query Structure .....	124
<b>Figure 7.6</b> What-if Analysis (Before) .....	125
<b>Figure 7.7</b> What-if Analysis (After) .....	126
<b>Figure 7.8</b> Reorganizing the Query Structure (Before) .....	127
<b>Figure 7.9</b> Reorganizing the Query Structure (First Move) .....	128
<b>Figure 7.10</b> Reorganizing the Query Structure (Second Move) .....	129
<b>Figure 7.11</b> Reorganizing the Query Structure (Third Move) .....	130
<b>Figure 7.12</b> Reorganizing the Query Structure (Fourth Move) .....	131
<b>Figure 7.13</b> Complex Query Outline .....	132
<b>Figure 7.14</b> Complex Query Outline Visualized as an InfoCrystal .....	133
<b>Figure 8.1</b> Recognition Task Display (InfoCrystal) .....	142
<b>Figure 8.2</b> Recognition Task Display (Boolean) .....	143
<b>Figure 8.3</b> Generation Task Display (InfoCrystal) .....	145
<b>Figure 8.4</b> Generation Task Display (Boolean) .....	146
<b>Figure 8.5</b> Overview of How Data Will Be Analyzed.....	148
<b>Figure 8.6</b> Simplifying a Query using a Hierarchical InfoCrystal Query .....	160
<b>Figure 10.1</b> VIBE (Visualization By Example) .....	186
<b>Figure 10.2</b> Semantic Map .....	188
<b>Figure 10.3</b> Cybermap .....	188
<b>Figure 10.4</b> BEAD .....	189

---

<b>Figure 10.5</b> Scatter/Gather .....	190
<b>Figure 10.6</b> Tree-Map .....	192
<b>Figure 10.7</b> Cone Tree .....	193
<b>Figure 10.8</b> Piles .....	196
<b>Figure 10.9</b> TileBars .....	196
<b>Figure 10.10</b> Venn Diagrams .....	198
<b>Figure 10.11</b> Cougar .....	199
<b>Figure 10.12</b> Cube of Contents .....	199
<b>Figure 10.13</b> Filter/Flow .....	200
<b>Figure 11.1</b> InfoCrystal used as a General-Purpose Coordinator and Generator of Diverse Data Streams .....	204
<b>Figure 12.1</b> Input/Cone Representation .....	214
<b>Figure 12.2</b> Input/Cone Used to Define Inputs to Five-Concept InfoCrystal ...	214
<b>Figure 12.3</b> Using Clustering Techniques to Define the InfoCrystal Inputs .....	215

---

# LIST OF TABLES

<b>Table 2.1</b> Standard Boolean Model .....	31
<b>Table 2.2</b> Smart Boolean Model .....	35
<b>Table 2.3</b> Extended Boolean Model .....	37
<b>Table 2.4</b> Statistical Retrieval Models .....	41
<b>Table 2.5</b> Classification of Major Retrieval Methods Based on Linguistic Features.....	45
<b>Table 2.6</b> Key Problems of Major Retrieval Methods and Possible Solutions ...	46
<b>Table 8.1</b> Queries for the Recognition Task .....	141
<b>Table 8.2</b> Queries for the Generation Task .....	144
<b>Table 8.3</b> Analysis of Variance Table .....	151
<b>Table 8.4</b> Scores for the Recognition Task .....	154
<b>Table 8.5</b> Paired-Difference T-test for Recognition Task (Scores) .....	155
<b>Table 8.6</b> ANOVA Tables for Recognition Task (Scores) .....	155
<b>Table 8.7</b> Time Data for Recognition Task .....	156
<b>Table 8.8</b> Paired-Difference T-test for Recognition Task (Time Measurements) .....	157
<b>Table 8.9</b> ANOVA Tables for Recognition Task (Time Measurements) .....	158
<b>Table 8.10</b> Analysis of Specific Queries for the Recognition Task .....	159
<b>Table 8.11</b> Categorical Scores for the Generation Task .....	164

---

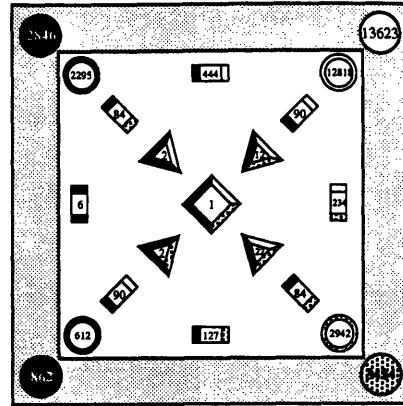
<b>Table 8.12</b> Paired-Difference T-test for Generation Task (Categorical Scores) .....	164
<b>Table 8.13</b> ANOVA Tables for Generation Task (Categorical Scores) .....	165
<b>Table 8.14</b> Continuous Scores for the Generation Task .....	166
<b>Table 8.15</b> Paired-Difference T-test for Generation Task (Continuous Scores).....	167
<b>Table 8.16</b> ANOVA Tables for Generation Task (Continuous Scores) .....	167
<b>Table 8.17</b> Time Data for Generation Task.....	168
<b>Table 8.18</b> Paired-Difference T-test for Generation Task (Time Measurements).....	169
<b>Table 8.19</b> ANOVA Tables for Generation Task (Time Measurements) .....	170
<b>Table 8.20</b> Analysis of Specific Queries for the Generation Task .....	172
<b>Table 8.21</b> Misses and False Alarms .....	173
<b>Table 9.1</b> User Feedback .....	182

---



# CHAPTER 1

## INTRODUCTION



*InfoCrystal*

Information is being created and becoming available in ever growing quantities as the access possibilities to it proliferate. There is currently a great deal of excitement and confusion about the promise of an Electronic Information Superhighway that would enable anybody to access these diverse and large information sources. Many information providers are developing on-line services to provide users with an interface to this emerging rich universe of knowledge stored in the form of multimedia documents, business and financial data, games and entertainment, shopping and consumer information. However, the realization of the promise to make any information available to users almost instantly, commonly referred to as the *information explosion*, is already becoming a mixed blessing without better methods to filter, retrieve and manage this potentially unlimited influx of information. Users face an *information overload* problem and they require tools to explore this vast universe of information in a structured way.

Information visualization techniques can provide better methods for accessing and understanding large information spaces. This thesis develops a novel spatial representation, called the *InfoCrystal*, that can visualize abstract information spaces, such as document spaces, that do not have explicit spatial properties that simplify the visualization problem. The development of such representations contributes both to the emerging field of information visualization and to the established field of information retrieval. The *InfoCrystal* embodies new visual representation techniques that can help to solve problems encountered in information retrieval. More generally, the *InfoCrystal* has broad applications because it offers a "visual machinery" to compare and relate any number of arbitrary data sets.

Highly trained users, who perform complex data explorations, will likely be the first adopters of the tools developed in this thesis. As these tools will become more popular, they may be integrated into an interface with a broad appeal that enables users to "surf" the information explosion and "cruise" on the Information Superhighway.

## **1.1 Information Visualization**

Researchers at Xerox PARC believe that visual interfaces that recode the information in progressively more abstract and simpler representations will play a central role in the effective management of large information spaces [Card et al. 1991]. Recent work in scientific visualization shows how large sets of data can be visualized in such a way that human perception can detect patterns revealing the underlying structure in the data more readily than by a direct analysis of the numbers [Rosenblum 1994]. When applied to retrieving information, information visualization seeks to reveal structural relationships between documents and their context that would be more difficult to detect by individual retrieval requests [Card et al. 1991].

Humans have a highly developed and versatile ability to extract information from visual stimuli. The field of Computational Vision is trying to determine how the human visual system processes information and what constraints it exploits to arrive at a three-dimensional perception given the two-dimensional nature of its input [Marr 1982]. A major constraint, which the human visual system uses, is that the visible physical world consists mostly of smooth surfaces whose visual properties change smoothly across them, except at object boundaries, and that objects change their position in a continuous fashion. Hence, for visualization to succeed, transformations have to be found, whereby the visual activity on the computer screen reflects a virtual reality that shares many of the laws and principles governing the physical world for which our human perceptual system has been "optimized". In particular, a transformation must lead to visual codes whose features vary smoothly across some portion of the image and lead to visual discontinuities that are meaningful with respect to the data. Ideally, the variables used to create visual codes should not lead to spurious and meaningless perceptual boundaries.

---

Many abstract concepts seem to be mentally represented by structures originally dedicated to the representation of space and the movement of objects within it [Pinker 1990]. It has long been known that an object's spatial location has a different perceptual status than its color, lightness, texture, or shape, and that people extract information more easily from spatial representations. Spatial data provide a structure for storing and retrieving information and facilitate recall. Hence, visualization should exploit spatial properties of data or provide suitable spatial metaphors to be effective.

Most of the visualization problems that are currently being investigated involve continuous, multi-variate fields over space and time [Rosenblum 1994]. Hence, the transformation problem is simplified, because the data has an explicit spatial structure that can be exploited. This thesis, however, addresses the difficult problem of how to visualize information that is abstract and does not have explicit spatial properties that can be exploited. In particular, it addresses how to access large information spaces, where users usually find it hard to visualize how the contents relate to their interests. This thesis deals with the challenging question of how to visually encode an abstract information space so as to exploit the ability of the human visual system to rapidly recognize spatial patterns and to minimize the cognitive load. In particular, it is the goal to create a representation that provides a *spatial overview* of the data elements and *simultaneously* provides *visual cues about the content* of the data elements. These opposing requirements are difficult to satisfy, especially when the content of the data elements needs to be described along many dimensions, as is the case, for example, with documents that are described by multiple keywords or concepts. This thesis attempts to resolve these opposing requirements by exploiting the grouping principles used by the human visual system to make relationships between different, but related data elements visible and immediate. Further, it creates a visual representation that not only has *descriptive* power, because it enables users to see large amounts of information in a compact way, but that also has *expressive* power that enables users, for example, to interact with the data to issue commands.

---

## 1.2 Information Retrieval

The domain of information retrieval poses three challenges. First, the currently dominant Boolean or Exact Matching approach needs to become more user-friendly. General users find it difficult to use the Boolean operators and apply parentheses to formulate effective Boolean queries [Borgman 1989, Belkin and Croft 1993]. Further, few have mastered how to fully exploit the expressive power of Boolean query language [Marcus 1991]. Second, the Partial Matching approaches, which are initially easier to use, present users with a sequential list of the "best" documents. This can create a "tunnel vision" effect, because the ranked list obscures what the role the query terms played in the ranking of the retrieved documents. Users could use this type of feedback to help them decide how to proceed in their search. Third, recent retrieval experiments have shown that the competing Exact and Partial matching approaches are complementary because the sets of relevant documents retrieved by them do not overlap to a great extent [Belkin et al. 1993]. Hence, there is a growing consensus that a combination of these two approaches is needed to enhance the retrieval effectiveness [Belkin et al. 1993]. However, the complementary Exact and Partial Matching approaches need to be combined in a framework that enables users to make effective use of their respective strengths.

The problems mentioned above and of the lack of visual feedback cause users to feel confused while searching for information, which in turn undermines their confidence and effectiveness. There is a growing awareness that besides the need to develop more versatile retrieval methods, a great deal of leverage can be obtained by developing better visual tools that support users in the search process and that provide them with a more comprehensive overview of an information space [Fox et al. 1993, Kahle et al. 1993].

Metaphorically speaking, it is as if users, using current retrieval methods, have to begin their exploration of a large information space in darkness. On the one hand, they can use a flashlight with a very narrow, but powerful beam of light (i.e., formulating a very specific and complex query: high precision, but low recall) which gives them only a very limited view of the information space. In order to piece together a more comprehensive picture, users need to cast the flashlight in different directions in an orchestrated

---

fashion (i.e., formulating multiple queries guided by a well-developed strategy requiring sufficient expertise). On the other hand, users can use a light source that casts a wide but very dim beam of light (i.e., formulating a simple and broad query: high recall, but low precision) which provides them only with a very murky and undifferentiated view. Instead of being in darkness, they are now surrounded by thick fog, where too much information is presented in a very unstructured way, and it is not clear how the retrieved data really relates to their interests. It is our goal to provide users with a lighting environment that enables them to use multiple light sources at the same time to illuminate the information space, where the emerging structure is clearly perceivable and can be easily interpreted. Further, the proposed tool should allow users to create complex and powerful lighting strategies that reveal areas in the information space that are of great interest to them or provide them with insight into how to proceed in the search process.

### 1.3 Goal of the Thesis

This thesis demonstrates how information visualization offers ways to accomplish the needed improvements in information retrieval. In particular, this thesis addresses the problem of how to enhance the ability of users to access information by developing better ways for visualizing information and formulating queries graphically. Further, it develops a visual framework that unifies the Exact and the Partial Matching approaches and enables users to take advantage of their respective strengths. As the amount of available information keeps growing at an ever increasing rate, it will become critical to provide users with *high-level visual retrieval tools* that enable them to explore, manipulate, and relate large information spaces to their interests in an interactive way. We use the term "high-level" because these tools are designed to give users a flexible visual framework for both how to retrieve and how to explore information.

To address the problems outlined above, this thesis develops the InfoCrystal, which is an example of such a high-level retrieval tool and it has the following functionality: 1) Users can *explore* an information space along several dimensions simultaneously without having to abandon their sense of overview. 2) Users can *manipulate* the information by *creating useful*

---

*abstractions*. 3) Similar to a spreadsheet, users can ask "*what-if*" questions and observe the effects without having to change the framework of a query. 4) Users receive *support* in the search process because they receive *dynamic visual feedback* on how to proceed. 5) Users can formulate queries *graphically*, and they have *flexibility* in terms of the particular methods used to retrieve the information.

## 1.4 Thesis Organization

This thesis is organized as follows: 1) We will consider a concrete retrieval example to set the stage. 2) We will review the major text retrieval paradigms such as the Exact Matching and the Partial Matching approaches. 3) We will introduce the InfoCrystal and proceed to demonstrate how it can be used to visualize and formulate Boolean, weighted and vector space queries. We will also describe a query outlining tool that enables users to create and manage complex queries. 4) We will give a brief overview of the current InfoCrystal software environment. 5) We will report on a set of two evaluation experiments that we conducted to test specific aspects of the InfoCrystal interface by comparing with a standard Boolean interface. In an appendix we will describe in detail the tutorial that introduced the subjects to the InfoCrystal interface. Further, we will present the feedback received from the experimental subjects. 6) We will review and compare relevant previous research with the InfoCrystal. 7) We will describe several brief application scenarios of the InfoCrystal. 8) We will outline the research to be conducted in the future. 9) We will provide a summary of the key accomplishments of this thesis. Finally, we will also reflect on the major challenges and opportunities facing the field of information visualization.

## 1.5 Concrete Example

It is best to consider a concrete example to describe some the problems a user currently faces when searching for information. For example, if we are interested in documents that talk about "visual query languages for retrieving information and that consider human factors issues" then the first problem we are faced with is the *vocabulary problem*. Which particular concepts should we use to represent our information need ? The following concepts could capture our interest: (*Graphical OR Visual*), *Information*

---

*Retrieval, Query language, Human Factors.* Most of the existing on-line retrieval systems use Boolean or Exact Matching operators to combine the identified concepts to form a query. Hence, we are faced next with the *coordination problem*. Which operators should we use and how should we use them to coordinate the concepts ? On the one hand, the most exclusive query would join the concepts by using the AND operator. Such a query, performed on the INSPEC Database for the years 1991-92, retrieved only one document containing all four concepts. On the other hand, the most inclusive query would join the concepts by using the OR operator; it retrieved 19,691 documents. Hence, either too few documents or too many documents are presented. How should we broaden the exclusive query or narrow the inclusive query to retrieve more relevant documents? We will revisit this example after we have introduced the InfoCrystal and we will show how it could help users to modify the query successfully.

---





## CHAPTER 2

# INFORMATION RETRIEVAL MODELS

### 2.1 Introduction

The purpose of this chapter is two-fold: First, we want to set the stage for the problems in information retrieval that we try to address in this thesis. Second, we want to give the reader a quick overview of the major textual retrieval methods, because the InfoCrystal can help to visualize the output from any of them. We begin by providing a general model of the information retrieval process. We then briefly describe the major retrieval methods and characterize them in terms of their strengths and shortcomings.

### 2.2 General Model of Information Retrieval

The goal of **information retrieval** (IR) is to provide users with those documents that will satisfy their information need. We use the word "document" as a general term that could also include non-textual information, such as multimedia objects. Figure 4.1 provides a general overview of the information retrieval process, which has been adapted from Lancaster and Warner (1993). Users have to formulate their information need in a form that can be understood by the retrieval mechanism. There are several steps involved in this translation process that we will briefly discuss below. Likewise, the contents of large document collections need to be described in a form that allows the retrieval mechanism to identify the potentially relevant documents quickly. In both cases, information may be lost in the transformation process leading to a computer-usable representation. Hence, the matching process is inherently imperfect.

Information seeking is a form of problem solving [Marcus 1994, Marchionini 1992]. It proceeds according to the interaction among eight subprocesses: problem recognition and acceptance, problem definition, search system selection, query formulation, query execution, examination of results

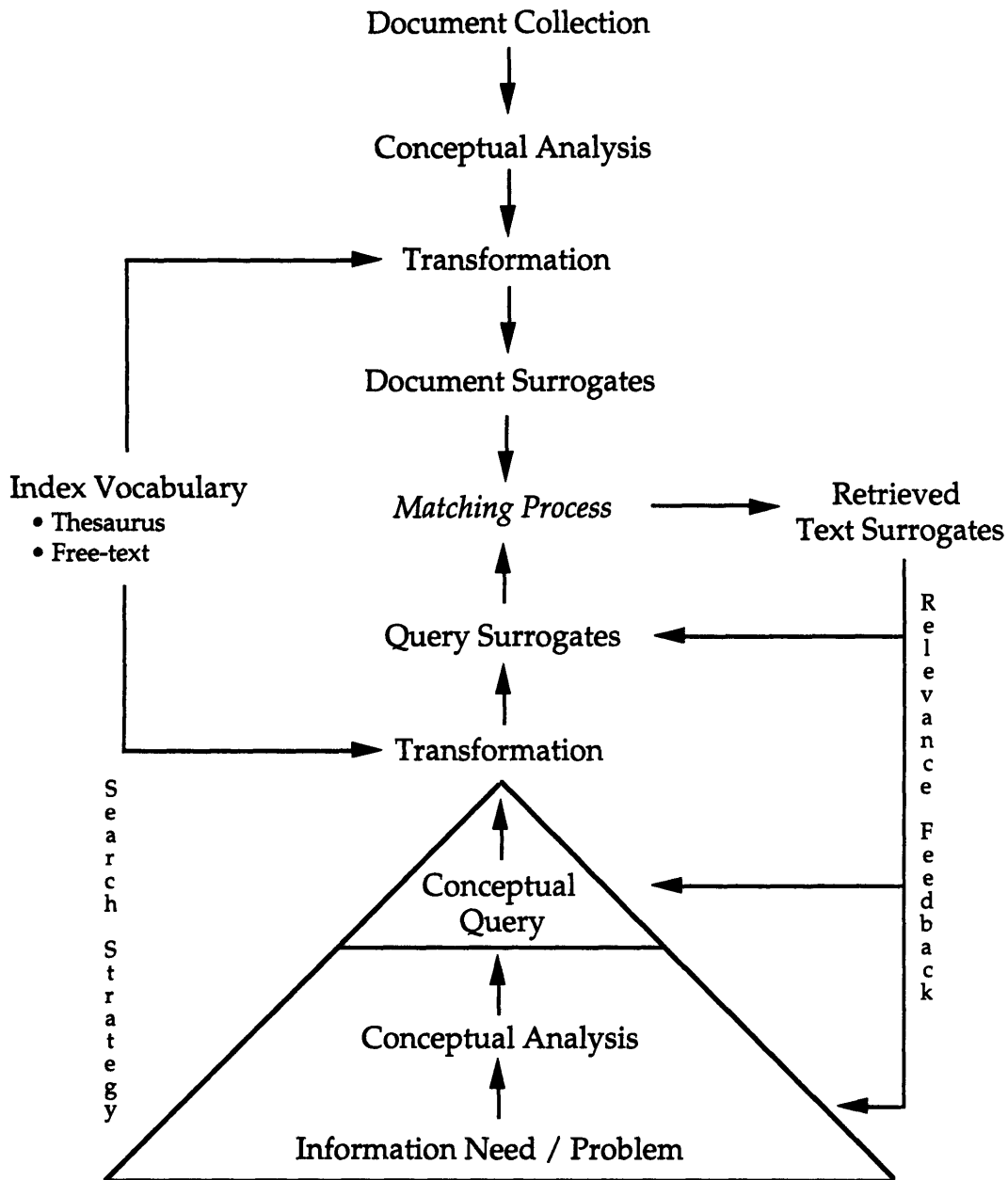
---

(including relevance feedback), information extraction, and reflection/iteration/termination. To be able to perform effective searches, users have to develop the following expertise: knowledge about various sources of information, skills in defining search problems and applying search strategies, and competence in using electronic search tools.

Marchionini (1992) contends that some sort of spreadsheet is needed that supports users in the problem definition as well as other information seeking tasks. The InfoCrystal is such a spreadsheet because it assists users in the formulation of their information needs and the exploration of the retrieved documents, using the a visual interface that supports a "what-if" functionality. He further predicts that advances in computing power and speed, together with improved information retrieval procedures, will continue to blur the distinctions between problem articulation and examination of results. The InfoCrystal is both a visual query language and a tool for visualizing retrieval results.

The information need can be understood as forming a pyramid, where only its peak is made visible by users in the form of a conceptual query (see Figure 2.1). The conceptual query captures the key concepts and the relationships among them. It is the result of a conceptual analysis that operates on the information need, which may be well or vaguely defined in the user's mind. This analysis can be challenging, because users are faced with the general "vocabulary problem" as they are trying to translate their information need into a conceptual query. This problem refers to the fact that a single word can have more than one meaning, and, conversely, the same concept can be described by surprisingly many different words. Furnas, Landauer, Gomez and Dumais (1983) have shown that two people use the same main word to describe an object only 10 to 20% of the time. Further, the concepts used to represent the documents can be different from the concepts used by the user. The conceptual query can take the form of a natural language statement, a list of concepts that can have degrees of importance assigned to them, or it can be statement that coordinates the concepts using Boolean operators. Finally, the conceptual query has to be translated into a query surrogate that can be understood by the retrieval system.

---



**Figure 2.1:** represents a general model of the information retrieval process, where both the user's information need and the document collection have to be translated into the form of surrogates to enable the matching process to be performed. This figure has been adapted from Lancaster and Warner (1993).

Similarly, the meanings of documents need to be represented in the form of text surrogates that can be processed by computer. A typical surrogate can consist of a set of index terms or descriptors. The text surrogate can consist of multiple fields, such as the title, abstract, descriptor fields to capture the meaning of a document at different levels of resolution or focusing on different characteristic aspects of a document. Once the specified query has

been executed by IR system, a user is presented with the retrieved document surrogates. Either the user is satisfied by the retrieved information or he will evaluate the retrieved documents and modify the query to initiate a further search. The process of query modification based on user evaluation of the retrieved documents is known as relevance feedback [Lancaster and Warner 1993]. Information retrieval is an inherently interactive process, and the users can change direction by modifying the query surrogate, the conceptual query or their understanding of their information need.

It is worth noting here the results, which have been obtained in studies investigating the information-seeking process, that describe information retrieval in terms of the cognitive and affective symptoms commonly experienced by a library user. The findings by Kuhlthau et al. (1990) indicate that thoughts about the information need become clearer and more focused as users move through the search process. Similarly, uncertainty, confusion, and frustration are nearly universal experiences in the early stages of the search process, and they decrease as the search process progresses and feelings of being confident, satisfied, sure and relieved increase. The studies also indicate that cognitive attributes may affect the search process. User's expectations of the information system and the search process may influence the way they approach searching and therefore affect the intellectual access to information.

Analytical search strategies require the formulation of specific, well-structured queries and a systematic, iterative search for information, whereas browsing involves the generation of broad query terms and a scanning of much larger sets of information in a relatively unstructured fashion. Campagnoni et al. (1989) have found in information retrieval studies in hypertext systems that the predominant search strategy is "browsing" rather than "analytical search". Many users, especially novices, are unwilling or unable to precisely formulate their search objectives, and browsing places less cognitive load on them. Furthermore, their research showed that search strategy is only one dimension of effective information retrieval; individual differences in visual skill appear to play an equally important role.

These two studies argue for information displays that provide a spatial overview of the data elements and that simultaneously provide rich visual cues about the content of the individual data elements. Such a representation

---

is less likely to increase the anxiety that is a natural part of the early stages of the search process and it caters for a browsing interaction style, which is appropriate especially in the beginning, when many users are unable to precisely formulate their search objectives.

## 2.3 Major Information Retrieval Models

The following major models have been developed to retrieve information: the **Boolean** model, the **Statistical** model, which includes the vector space and the probabilistic retrieval model, and the **Linguistic and Knowledge-based** models. The first model is often referred to as the "exact match" model; the latter ones as the "best match" models [Belkin and Croft 1992]. The material presented here is based on the textbooks by Lancaster and Warner (1992) as well as Frakes and Baeza-Yates (1992), the review article by Belkin and Croft (1992), and discussions with Richard Marcus, my thesis advisor and mentor in the field of information retrieval.

Queries generally are less than perfect in two respects: First, they retrieve some irrelevant documents. Second, they do not retrieve all the relevant documents. The following two measures are usually used to evaluate the effectiveness of a retrieval method. The first one, called the *precision rate*, is equal to the proportion of the retrieved documents that are actually relevant. The second one, called the *recall rate*, is equal to the proportion of all relevant documents that are actually retrieved. If searchers want to raise precision, then they have to narrow their queries. If searchers want to raise recall, then they broaden their query. In general, there is an inverse relationship between precision and recall. Users need help to become knowledgeable in how to manage the precision and recall trade-off for their particular information need [Marcus 1991].

### 2.3.1 Boolean Retrieval

A query in a modern Boolean-based system can be characterized along the following four dimensions: First, it uses the Boolean operators AND, OR, and NOT to coordinate the identified concepts to form a query. Second, users can impose proximity requirements between terms, whereby two terms have to appear next to each other or in the same sentence, paragraph or section.

---

Proximity constraints enable users to form phrase-like queries, which can be more reliable carriers of meaning than single terms out of context. Third, users can require that a concept appear in particular fields, such as the author, title, controlled index, descriptors, abstract or full-text field. Fourth, users can perform a stemming or truncation operation on a word. By reducing a word to its morphological stem and using it as a prefix, users can retrieve many words that are related to the original term [Marcus 1991]. Users can formulate queries with different precision and recall characteristics by making the appropriate choices along these four dimensions.

### **2.3.1.1 Standard Boolean**

In Table 2.1 we summarize the defining characteristics of the standard Boolean approach and list its key advantages and disadvantages. It has the following strengths: 1) It is easy to implement and it is computationally efficient [Frakes and Baeza-Yates 1992]. Hence, it is the standard model for the current large-scale, operational retrieval systems and many of the major on-line information services use it. 2) It enables users to express structural and conceptual constraints to describe important linguistic features [Marcus 1991]. Users find that synonym specifications (reflected by OR-clauses) and phrases (represented by proximity relations) are useful in the formulation of queries [Cooper 1988, Marcus 1991]. 3) The Boolean approach possesses a great expressive power and clarity. Boolean retrieval is very effective if a query requires an exhaustive and unambiguous selection. 4) The Boolean method offers a multitude of techniques to broaden or narrow a query. 5) The Boolean approach can be especially effective in the later stages of the search process, because of the clarity and exactness with which relationships between concepts can be represented.

The standard Boolean approach has the following shortcomings: 1) Users find it difficult to construct effective Boolean queries for several reasons [Cooper 1988, Fox and Koll 1988, Belkin and Croft 1992]. Users are using the natural language terms AND, OR or NOT that have a different meaning when used in a query. Thus, users will make errors when they form a Boolean query, because they resort to their knowledge of English. For example, in ordinary conversation a noun phrase of the form "A and B"

---

	<b>Standard Boolean</b>
<b>Goal</b>	<ul style="list-style-type: none"> <li>• Capture conceptual structure and contextual information</li> </ul>
<b>Methods</b>	<ul style="list-style-type: none"> <li>• Coordination: AND, OR, NOT</li> <li>• Proximity</li> <li>• Fields</li> <li>• Stemming / Truncation</li> </ul>
<b>(+)</b>	<ul style="list-style-type: none"> <li>• Easy to implement</li> <li>• Computationally efficient =&gt; all the major on-line databases use it</li> <li>• Expressiveness and Clarity Synonym specifications (OR-clauses) and phrases (AND-clauses).</li> </ul>
<b>(-)</b>	<ul style="list-style-type: none"> <li>• Difficult to construct Boolean queries.</li> <li>• All or nothing AND too severe, and OR does not differentiate enough.</li> <li>• Difficult to control output: Null output &lt;-&gt; Overload.</li> <li>• No ranking</li> <li>• No weighting of index or query terms</li> <li>• No uncertainty measure</li> </ul>

**Table 2.1:** summarizes the defining characteristics of the standard Boolean approach and list the its key advantages and disadvantages.

usually refers to more entities than would "A" alone, whereas when used in the context of information retrieval it refers to fewer documents than would be retrieved by "A" alone. Hence, one of the common mistakes made by users is to substitute the AND logical operator for the OR logical operator when translating an English sentence to a Boolean query. Furthermore, to form complex queries, users must be familiar with the rules of precedence and the use of parentheses. Novice users have difficulty using parentheses, especially nested parentheses. Finally, users are overwhelmed by the multitude of ways a query can be structured or modified, because of the combinatorial explosion of feasible queries as the number of concepts increases. In particular, users have difficulty identifying and applying the different strategies that are available for narrowing or broadening a Boolean query [Marcus 1991, Lancaster and Warner 1993]. 2) Only documents that satisfy a query exactly are

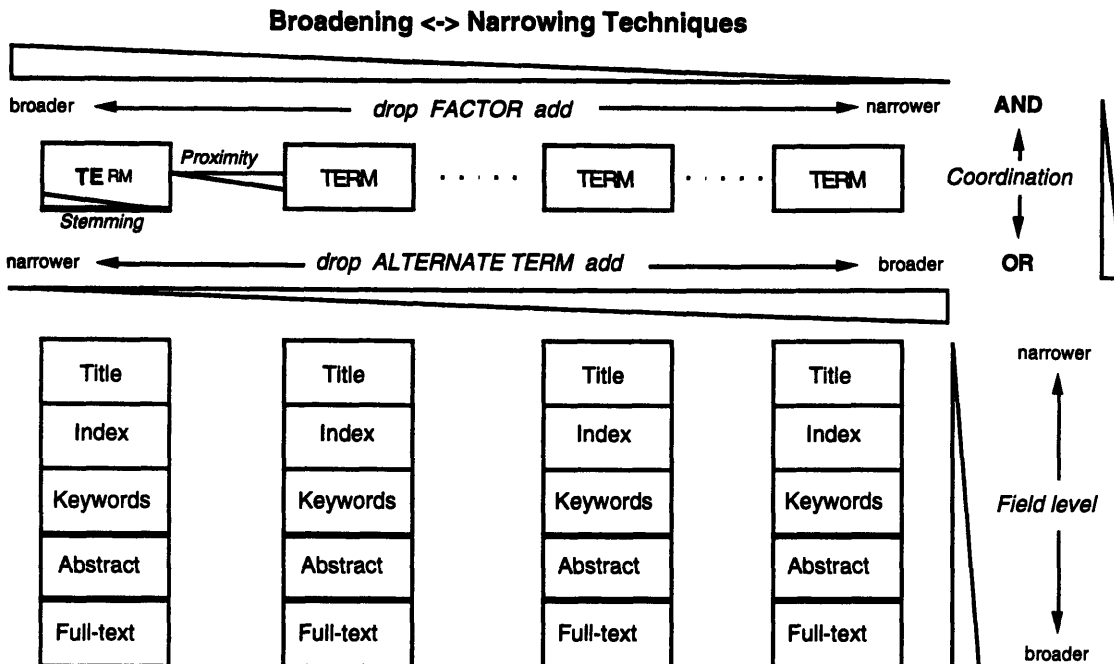
retrieved. On the one hand, the AND operator is too severe because it does not distinguish between the case when none of the concepts are satisfied and the case where all except one are satisfied. Hence, no or very few documents are retrieved when more than three and four criteria are combined with the Boolean operator AND (referred to as the Null Output problem). On the other hand, the OR operator does not reflect how many concepts have been satisfied. Hence, often too many documents are retrieved (the Output Overload problem). 3) It is difficult to control the number of retrieved documents. Users are often faced with the null-output or the information overload problem and they are at loss of how to modify the query to retrieve the reasonable number documents. 4) The traditional Boolean approach does not provide a relevance ranking of the retrieved documents, although modern Boolean approaches can make use of the degree of coordination, field level and degree of stemming present to rank them [Marcus 1991]. 5) It does not represent the degree of uncertainty or error due the vocabulary problem [Belkin and Croft 1992].

### **2.3.1.2 Narrowing and Broadening Techniques**

As mentioned earlier, a Boolean query can be described in terms of the following four operations: degree and type of coordination, proximity constraints, field specifications and degree of stemming as expressed in terms of word/string specifications. If users want to (re)formulate a Boolean query then they need to make informed choices along these four dimensions to create a query that is sufficiently broad or narrow depending on their information needs. Most narrowing techniques lower recall as well as raise precision, and most broadening techniques lower precision as well as raise recall. Any query can be reformulated to achieve the desired precision or recall characteristics, but generally it is difficult to achieve both. Each of the four kinds of operations in the query formulation has particular operators, some of which tend to have a narrowing or broadening effect. For each operator with a narrowing effect, there is one or more inverse operators with a broadening effect [Marcus 1991]. Hence, users require help to gain an understanding of how changes along these four dimensions will affect the broadness or narrowness of a query.

---





**Figure 2.2:** captures how coordination, proximity, field level and stemming affect the broadness or narrowness of a Boolean query. By moving in the direction in which the wedges are expanding the query is broadened.

Figure 2.2 shows how the four dimensions affect the broadness or narrowness of a query: 1) *Coordination*: the different Boolean operators AND, OR and NOT have the following effects when used to add a further concept to a query: a) the AND operator narrows a query; b) the OR broadens it; c) the effect of the NOT depends on whether it is combined with an AND or OR operator. Typically, in searching textual databases, the NOT is connected to the AND, in which case it has a narrowing effect like the AND operator. 2) *Proximity*: The closer together two terms have to appear in a document, the more narrow and precise the query. The most stringent proximity constraint requires the two terms to be adjacent. 3) *Field level*: current document records have fields associated with them, such as the "Title", "Index", "Abstract" or "Full-text" field: a) the more fields that are searched, the broader the query; b) the individual fields have varying degrees of precision associated with them, where the "title" field is the most specific and the "full-text" field is the most general. 4) *Stemming*: The shorter the prefix that is used in truncation-based searching, the broader the query. By reducing a term to its morphological stem and using it as a prefix, users can retrieve many terms that are conceptually related to the original term [Marcus 1991].

Using Figure 2.2, we can easily read off how to broaden query. We just need to move in the direction in which the wedges are expanding: we use the OR operator (rather than the AND), impose no proximity constraints, search over all fields and apply a great deal of stemming. Similarly, we can formulate a very narrow query by moving in the direction in which the wedges are contracting: we use the AND operator (rather than the OR), impose proximity constraints, restrict the search to the title field and perform exact rather than truncated word matches. In Chapter 4 we will show how Figure 2.2 indicates how the broadness or narrowness of a Boolean query could be visualized.

### 2.3.1.3 Smart Boolean

There have been attempts to help users overcome some of the disadvantages of the traditional Boolean discussed above. We will now describe such a method, called *Smart Boolean*, developed by Marcus [1991, 1994] that tries to help users construct and modify a Boolean query as well as make better choices along the four dimensions that characterize a Boolean query. We are not attempting to provide an in-depth description of the Smart Boolean method, but to use it as a good example that illustrates some of the possible ways to make Boolean retrieval more user-friendly and effective. Table 2.2 provides a summary of the key features of the Smart Boolean approach.

Users start by specifying a natural language statement that is automatically translated into a Boolean Topic representation that consists of a list of factors or concepts, which are automatically coordinated using the AND operator. If the user at the initial stage can or wants to include synonyms, then they are coordinated using the OR operator. Hence, the Boolean Topic representation connects the different factors using the AND operator, where the factors can consist of single terms or several synonyms connected by the OR operator. One of the goals of the Smart Boolean approach is to make use of the structural knowledge contained in the text surrogates, where the different fields represent contexts of useful information. Further, the Smart Boolean approach wants to use the fact that related concepts can share a common stem. For example, the concepts "computers" and "computing" have the common stem comput\*. The initial strategy of the Smart Boolean approach is to start out with the broadest possible query within the constraints of how the

---

	<b>Smart Boolean</b>										
<b>Goal</b>	<ul style="list-style-type: none"> <li>• Structure search (re-)formulation process.</li> <li>• Use structural and contextual knowledge-bases and clarity of Boolean expressions.</li> </ul>										
<b>Methods</b>	<ul style="list-style-type: none"> <li>• Natural language statement is automatically translated into Boolean Topic Representation</li> <li>• Boolean Topic Representation:           <table style="margin-left: 20px; border: none;"> <tr> <td>ANDs of ORs of concepts</td> <td>Keyword/stem, all fields</td> </tr> <tr> <td>• Conceptual info. -&gt;</td> <td>Coordination and Add/Drop Factor</td> </tr> <tr> <td>• Contextual info. -&gt;</td> <td>Proximity</td> </tr> <tr> <td>• Structural info. -&gt;</td> <td>Field levels</td> </tr> <tr> <td>• Synonym or word relationships -&gt;</td> <td>Stemming/Truncation overlap</td> </tr> </table> <p>=&gt; all this information can be used to rank documents</p> </li> <li>• Techniques to Broaden and Narrow query</li> </ul>	ANDs of ORs of concepts	Keyword/stem, all fields	• Conceptual info. ->	Coordination and Add/Drop Factor	• Contextual info. ->	Proximity	• Structural info. ->	Field levels	• Synonym or word relationships ->	Stemming/Truncation overlap
ANDs of ORs of concepts	Keyword/stem, all fields										
• Conceptual info. ->	Coordination and Add/Drop Factor										
• Contextual info. ->	Proximity										
• Structural info. ->	Field levels										
• Synonym or word relationships ->	Stemming/Truncation overlap										
<b>(+)</b>	<ul style="list-style-type: none"> <li>• No need for Boolean operators =&gt; Convert operator-free statement into ANDs of ORs</li> <li>• Assist user in query (re)formulation: by asking users targeted questions to automatically modify the query.</li> <li>• "Why irrelevant?" -&gt; activates narrowing methods.</li> <li>• "Broaden by Dropping Factors" to estimate recall.</li> </ul>										
<b>(-)</b>	<ul style="list-style-type: none"> <li>• How to visualize ?           <ul style="list-style-type: none"> <li>• Conceptual query representation (BTR)</li> <li>• Query modification techniques and their effects</li> <li>• Structured relevance feedback</li> </ul> </li> </ul>										

**Table 2.2:** summarizes the defining characteristics of the Smart Boolean approach and list the its key advantages and disadvantages.

factors and their synonyms have been coordinated. Hence, it modifies the Boolean Topic representation into the query surrogate by using only the stems of the concepts and searches for them over all the fields. Once the query surrogate has been performed, users are guided in the process of evaluating the retrieved document surrogates. They choose from a list of reasons to indicate why they consider certain documents as relevant. Similarly, they can indicate why other documents are not relevant by interacting with a list of possible reasons. This user feedback is used by the Smart Boolean system to automatically modify the Boolean Topic



representation or the query surrogate, whatever is more appropriate. The Smart Boolean approach offers a rich set of strategies for modifying a query based on the received relevance feedback or the expressed need to narrow or broaden the query. The Smart Boolean retrieval paradigm has been implemented in the form of a system called CONIT, which is one of the earliest expert retrieval systems that was able to demonstrate that ordinary users, assisted by such a system, could perform equally well as experienced search intermediaries [Marcus 1983]. However, users have to navigate through a series of menus listing different choices, where it might be hard for them to appreciate the implications of some of these choices. A key limitation of the previous versions of the CONIT system has been that lacked a visual interface. The most recent version has a graphical interface and it uses the tiling metaphor suggested by Anick et al. (1991), and discussed in section 10.4, to visualize Boolean coordination [Marcus 1994]. This visualization approach suffers from the limitation that it enables users to visualize specific queries, whereas we will propose a visual interface that represents all whole range of related Boolean queries in a single display, making changes in Boolean coordination more user-friendly. Further, the different strategies of modifying a query in CONIT require a better visualization metaphor to enable users to make use these search heuristics. In Chapter 4 we show how some of these modification techniques can be visualized.

#### **2.3.1.4 Extended Boolean Models**

Several methods have been developed to extend the Boolean model to address the following issues: 1) The Boolean operators are too strict and ways need to be found to soften them. 2) The standard Boolean approach has no provision for ranking. The Smart Boolean approach and the methods described in this section provide users with relevance ranking [Fox and Koll 1988, Marcus 1991]. 3) The Boolean model does not support the assignment of weights to the query or document terms. We will briefly discuss the *P-norm* and the *Fuzzy Logic* approaches that extend the Boolean model to address the above issues.

The **P-norm** method developed by Fox (1983) allows query and document terms to have weights, which have been computed by using term frequency statistics with the proper normalization procedures. These normalized

---

	Extended Boolean Models
Goal	<ul style="list-style-type: none"> <li>• Less strict Boolean operators</li> <li>• Ranked output</li> </ul>
Methods	<ul style="list-style-type: none"> <li>• <b>P-norm</b>      OR       AND </li> <li>Uses a distance-based measure to approximate Boolean operators.</li> <li><math>p=1</math> : vector space, <math>p=\infty</math> : strict Boolean.</li> <li>• <math>SIM(query(OR),document) = \sqrt[p]{\sum q^p d^p / \sum q^p}</math></li> <li>• <math>SIM(query(AND),document) = 1 - \sqrt[p]{\sum q^p (1-d)^p / \sum q^p}</math></li> <li>• <math>SIM(query(NOT),document) = 1 - SIM(query, document)</math></li> <li>(-) Computationally expensive.</li> <li>(-) Not all axioms of Boolean algebra satisfied.</li> </ul> <hr/> <ul style="list-style-type: none"> <li>• <b>Fuzzy logic</b></li> <li>[OR -&gt; max], [AND -&gt; min] and [NOT -&gt; 1 - max]</li> <li>(-) Lack of sensitivity of min and max:  <math>\min(0.2, 0.8) = \min(0.2, 0.3)</math>.</li> </ul>

**Table 2.3:** summarizes the defining characteristics of the Extended Boolean approach and list the its key advantages and disadvantages.

weights can be used to rank the documents in the order of decreasing distance from the point  $(0, 0, \dots, 0)$  for an OR query, and in order of increasing distance from the point  $(1, 1, \dots, 1)$  for an AND query. Further, the Boolean operators have a coefficient  $P$  associated with them to indicate the degree of strictness of the operator (from 1 for least strict to infinity for most strict, i.e., the Boolean case). The P-norm uses a distance-based measure and the coefficient  $P$  determines the degree of exponentiation to be used. The exponentiation is an expensive computation, especially for  $P$ -values greater than one.

In **Fuzzy Set theory**, an element has a varying degree of membership to a set instead of the traditional binary membership choice. The weight of an index term for a given document reflects the degree to which this term describes the content of a document. Hence, this weight reflects the degree of membership of the document in the fuzzy set associated with the term in question. The degree of membership for union and intersection of two fuzzy sets is equal to the maximum and minimum, respectively, of the degrees of membership of the elements of the two sets. In the "Mixed Min and Max" model developed by Fox and Sharat (1986) the Boolean operators are softened by considering the query-document similarity to be a linear combination of the min and max weights of the documents.

### **2.3.2 Statistical Model**

The *vector space* and *probabilistic* models are the two major examples of the statistical retrieval approach. Both models use statistical information in the form of term frequencies to determine the relevance of documents with respect to a query. Although they differ in the way they use the term frequencies, both produce as their output a list of documents ranked by their estimated relevance. The statistical retrieval models address some of the problems of Boolean retrieval methods, but they have disadvantages of their own. Table 2.4 provides summary of the key features of the vector space and probabilistic approaches. We will also describe *Latent Semantic Indexing* and *clustering* approaches that are based on statistical retrieval approaches, but their objective is to respond to what the user's query did not say, could not say, but somehow made manifest [Furnas et al. 1983, Cutting et al. 1991].

#### **2.3.2.1 Vector Space Model**

The **vector space model** represents the documents and queries as vectors in a multidimensional space, whose dimensions are the terms used to build an index to represent the documents [Salton 1983]. The creation of an index involves lexical scanning to identify the significant terms, where morphological analysis reduces different word forms to common "stems", and the occurrence of those stems is computed. Query and document surrogates are compared by comparing their vectors, using, for example, the cosine similarity measure. In this model, the terms of a query surrogate can

---

be weighted to take into account their importance, and they are computed by using the statistical distributions of the terms in the collection and in the documents [Salton 1983]. The vector space model can assign a high ranking score to a document that contains only a few of the query terms if these terms occur infrequently in the collection but frequently in the document. The vector space model makes the following assumptions: 1) The more similar a document vector is to a query vector, the more likely it is that the document is relevant to that query. 2) The words used to define the dimensions of the space are orthogonal or independent. While it is a reasonable first approximation, the assumption that words are pairwise independent is not realistic.

### **2.3.2.2 Probabilistic Model**

The **probabilistic retrieval model** is based on the Probability Ranking Principle, which states that an information retrieval system is supposed to rank the documents based on their probability of relevance to the query, given all the evidence available [Belkin and Croft 1992]. The principle takes into account that there is uncertainty in the representation of the information need and the documents. There can be a variety of sources of evidence that are used by the probabilistic retrieval methods, and the most common one is the statistical distribution of the terms in both the relevant and non-relevant documents.

We will now describe the state-of-art system developed by Turtle and Croft (1991) that uses Bayesian inference networks to rank documents by using multiple sources of evidence to compute the conditional probability  $P(\text{Info need} | \text{document})$  that an information need is satisfied by a given document. An inference network consists of a directed acyclic dependency graph, where edges represent conditional dependency or causal relations between propositions represented by the nodes. The inference network consists of a document network, a concept representation network that represents indexing vocabulary, and a query network representing the information need. The concept representation network is the interface between documents and queries. To compute the rank of a document, the inference network is instantiated and the resulting probabilities are propagated through the network to derive a probability associated with the node representing the information need. These probabilities are used to rank documents.

---

The statistical approaches have the following strengths: 1) They provide users with a relevance ranking of the retrieved documents. Hence, they enable users to control the output by setting a relevance threshold or by specifying a certain number of documents to display. 2) Queries can be easier to formulate because users do not have to learn a query language and can use natural language. 3) The uncertainty inherent in the choice of query concepts can be represented. However, the statistical approaches have the following shortcomings: 1) They have a limited expressive power. For example, the NOT operation can not be represented because only positive weights are used. It can be proven that only  $2^{N \cdot N}$  of the  $2^{2^N}$  possible Boolean queries can be generated by the statistical approaches that use weighted linear sums to rank the documents. This result follows from the analysis of Linear Threshold Networks or Boolean Perceptrons [Anthony and Biggs 1992]. For example, the very common and important Boolean query ((A and B) or (C and D)) can not be represented by a vector space query (see section 5.4 for a proof). Hence, the statistical approaches do not have the expressive power of the Boolean approach. 3) The statistical approach lacks the structure to express important linguistic features such as phrases. Proximity constraints are also difficult to express, a feature that is of great use for experienced searchers. 4) The computation of the relevance scores can be computationally expensive. 5) A ranked linear list provides users with a limited view of the information space and it does not directly suggest how to modify a query if the need arises [Spoerri 1993, Hearst 1994]. 6) The queries have to contain a large number of words to improve the retrieval performance. As is the case for the Boolean approach, users are faced with the problem of having to choose the appropriate words that are also used in the relevant documents.

Table 2.4 summarizes the advantages and disadvantages that are specific to the vector space and probabilistic model, respectively. This table also shows the formulas that are commonly used to compute the term weights. The two central quantities used are the inverse term frequency in a collection (*idf*), and the frequencies of a term *i* in a document *j* (*freq(i,j)*). In the probabilistic model, the weight computation also considers how often a term appears in the relevant and irrelevant documents, but this presupposes that the relevant documents are known or that these frequencies can be reliably estimated.

---



<i>Statistical</i>	<b>Vector Space</b>	<b>Probabilistic</b>
<b>Motivation</b>	Simplify query formulation Ability to control output	Address uncertainty in query representations
<b>Goal</b>	Rank the output based on Similarity	Probability of Relevance
<b>Methods</b>	Cosine measure	Use of different models
<b>Source</b>	<b>Query Term Statistics</b> <u>Vector-Space:</u> <ul style="list-style-type: none"> <li>• <math>\text{similarity}(Q,D) = \sum (w_{iq} \times w_{ij}) / \text{"normalizer"}</math>  where <math>w_{iq} = (0.5 + 0.5 \text{freq}_{iq} / \text{maxfreq}_q) \times \text{idf}(i)</math>  <math>w_{ij} = \text{freq}_{ij} \times \text{idf}(i)</math></li> <li>• inverse term freq. in collection <math>\text{idf}(i) = \log_2 (N-n(i)) / n(i)</math>.</li> </ul> <u>Probabilistic:</u> <ul style="list-style-type: none"> <li>• term weight = <math>\log [(r_t / R-r_t) / ((n_t - r_t) / ((N-n_t) - (R-r_t)))]</math>  ="hits / misses) / (false alarms/correct misses)"</li> <li>• <math>\text{similarity}_{jk} = \sum (C + \text{idf}(i)) \times \text{tf}(i,j)</math>  where <math>\text{tf}(i,j) = K + (1-K) (\text{freq}(i,j) / \text{maxfreq}(j))</math>.</li> </ul>	
<b>Issues</b>	<ul style="list-style-type: none"> <li>• How to express NOT ?</li> <li>• Proximity searches ?</li> <li>• Limited expressive power</li> <li>• Computationally intensive</li> <li>• Assumes that terms are independent.</li> <li>• Lack of structure to represent important linguistic features</li> <li>• How to better visualize the retrieved set ?</li> </ul>	<ul style="list-style-type: none"> <li>• Estimation of needed probabilities</li> <li>• Prior knowledge needed.</li> <li>• Independence assumption</li> <li>• Boolean relations lost.</li> <li>• Which model is best ?</li> </ul>

**Table 2.4:** summarizes the defining characteristics of the statistical retrieval approach, which includes the vector space and the probabilistic model and we list the their key advantages and disadvantages.

If users provide the retrieval system with relevance feedback, then this information is used by the statistical approaches to recompute the weights as follows: the weights of the query terms in the relevant documents are increased, whereas the weights of the query terms that do not appear in the relevant documents are decreased [Salton and Buckley 1990]. There are multiple ways of computing and updating the weights, where each has its advantages and disadvantages. We do not discuss these formulas in more

detail, because research on relevance feedback has shown that significant effectiveness improvements can be gained by using quite simple feedback techniques [Salton and Buckley 1990]. Furthermore, what is important to this thesis is that the statistical retrieval approach generates a ranked list, however how this ranking has been computed in detail is immaterial for the purpose of this thesis.

### **2.3.2.3 Latent Semantic Indexing**

Several statistical and AI techniques have been used in association with domain semantics to extend the vector space model to help overcome some of the retrieval problems described above, such as the "dependence problem" or the "vocabulary problem". One such method is **Latent Semantic Indexing (LSI)**. In LSI the associations among terms and documents are calculated and exploited in the retrieval process. The assumption is that there is some "latent" structure in the pattern of word usage across documents and that statistical techniques can be used to estimate this latent structure. An advantage of this approach is that queries can retrieve documents even if they have no words in common. The LSI technique captures deeper associative structure than simple term-to-term correlations and is completely automatic. The only difference between LSI and vector space methods is that LSI represents terms and documents in a reduced dimensional space of the derived indexing dimensions. As with the vector space method, differential term weighting and relevance feedback can improve LSI performance substantially.

Foltz and Dumais (1992) compared four retrieval methods that are based on the vector-space model. The four methods were the result of crossing two factors, the first factor being whether the retrieval method used Latent Semantic Indexing or keyword matching, and the second factor being whether the profile was based on words or phrases provided by the user (Word profile), or documents that the user had previously rated as relevant (Document profile). The LSI match-document profile method proved to be the most successful of the four methods. This method combines the advantages of both LSI and the document profile. The document profile provides a simple, but effective, representation of the user's interests. Indicating just a few documents that are of interest is as effective as

---

generating a long list of words and phrases that describe one's interest. Document profiles have an added advantage over word profiles: users can just indicate documents they find relevant without having to generate a description of their interests.

#### **2.3.2.4 Document Clustering**

**Document Clustering** is another approach that has been extensively investigated as a method for improving information retrieval and to support users in the search process by enabling them to use browsing as a way of accessing information [Cutting et al. 1991]. Clustering algorithms can be divided in two categories: hierarchical and partitioning algorithms [Willet 1988]. Hierarchical algorithms either repeatedly combine the data elements to form increasingly larger clusters or they divide all of them into increasingly smaller clusters. The algorithms differ in terms of the similarity measures they use, but every measure considers all the elements in a pair of clusters to produce a value. Hierarchical algorithms are computationally expensive, but they provide an analysis of the data at different levels of granularity. Partitioning algorithms simply divide the data elements into one flat set of disjoint clusters by selecting an initial partition of the data and then iteratively moving elements to their nearest cluster centroid and eventually converging to a partition. These algorithms are less accurate, but they are usually faster than hierarchical clustering algorithms.

Document clustering is used to group texts with related vector representations, and term clustering is used to group related words and phrases. Representatives of the document clusters are used for comparison to the query, rather than the original text representations [Willet 1988]. Term clusters are typically used to expand the query representation. The general assumption of document clustering is that mutually similar documents will tend to be relevant to the same queries, and, hence, that automatic determination of groups of such documents can improve recall by effectively broadening a search request. Document clustering seeks to reduce the burden for a user in formulating a query by automating the process of inferring relevancy. Further, clustering intends to assist those who cannot always formulate a comprehensive query or who are not well versed in how to formulate a query [Cutting et al. 1991]. We will describe in Chapter 10 several

---

retrieval methods that use different clustering algorithms to generate overview maps of an information space.

### **2.3.3 Linguistic and Knowledge-based Approaches**

In the simplest form of automatic text retrieval, users enter a string of keywords that are used to search the inverted indexes of the document keywords. This approach retrieves documents based solely on the presence or absence of exact single word strings as specified by the logical representation of the query. Clearly this approach will miss many relevant documents because it does not capture the complete or deep meaning of the user's query. The Smart Boolean approach and the statistical retrieval approaches, each in their specific way, try to address this problem (see Table 2.5). Linguistic and knowledge-based approaches have also been developed to address this problem by performing a morphological, syntactic and semantic analysis to retrieve documents more effectively [Lancaster and Warner 1993]. In a morphological analysis, roots and affixes are analyzed to determine the part of speech (noun, verb, adjective etc.) of the words. Next complete phrases have to be parsed using some form of syntactic analysis. Finally, the linguistic methods have to resolve word ambiguities and/or generate relevant synonyms or quasi-synonyms based on the semantic relationships between words. The development of a sophisticated linguistic retrieval system is difficult and it requires complex knowledge bases of semantic information and retrieval heuristics. Hence these systems often require techniques that are commonly referred to as artificial intelligence or expert systems techniques.

#### **2.3.3.1 DR-LINK Retrieval System**

We will now describe in some detail the DR-LINK system developed by Liddy et al., because it represents an exemplary linguistic retrieval system. DR-LINK is based on the principle that retrieval should take place at the conceptual level and not at the word level. Liddy et al. attempt to retrieve documents on the basis of what people mean in their query and not just what they say in their query. DR-LINK system employs sophisticated, linguistic text processing techniques to capture the conceptual information in documents. Liddy et al. have developed a modular system that represents and matches text at the lexical, syntactic, semantic, and the discourse levels of language. Some of the

---

modules that have been incorporated are: The Text Structurer is based on discourse linguistic theory that suggests that texts of a particular type have a predictable structure which serves as an indication where certain information can be found. The Subject Field Coder uses an established semantic coding scheme from a machine-readable dictionary to tag each word with its disambiguated subject code (e.g., computer science, economics) and to then produce a fixed-length, subject-based vector representation of the document and the query. The Proper Noun Interpreter uses a variety of processing heuristics and knowledge bases to produce: a canonical representation of each proper noun; a classification of each proper noun into thirty-seven categories; and an expansion of group nouns into their constituent proper noun members. The Complex Nominal Phraser provides means for precise matching of complex semantic constructs when expressed as either adjacent nouns or a non-predicating adjective and noun pair. Finally, The Natural Language Query Constructor takes as input a natural language query and produces a formal query that reflects the appropriate logical combination of text structure, proper noun, and complex nominal requirements of the user's information need. This module interprets a query into pattern-action rules that translate each sentence into a first-order logic assertion, reflecting the Boolean-like requirements of queries.

Linguistic Level	Boolean Retrieval	Statistical	Linguistic and Knowledge-based
Lexical	Stop word list	Stop word list	Lexicon
Morphological	Truncation symbol	Stemming	Morphological analysis
Syntactic	Proximity operators	Statistical phrases	Grammatical phrases
Semantic	Thesaurus	Clusters of co-occurring words	Network of words/phrases in semantic relationships

**Table 2.5:** characterizes the major retrieval methods in terms of how deal with lexical, morphological, syntactic and semantic issues.

To summarize, the DR-LINK retrieval system represents content at the conceptual level rather than at the word level to reflect the multiple levels of human language comprehension. The text representation combines the lexical, syntactic, semantic, and discourse levels of understanding to predict the relevance of a document. DR-LINK accepts natural language statements, which it translates into a precise Boolean representation of the user's relevance requirements. It also produces a summary-level, semantic vector representations of queries and documents to provide a ranking of the documents.

## 2.4 Conclusion

There is a growing discrepancy between the retrieval approach used by existing commercial retrieval systems and the approaches investigated and promoted by a large segment of the information retrieval research community. The former is based on the Boolean or Exact Matching retrieval model, whereas the latter ones subscribe to statistical and linguistic approaches, also referred to as the Partial Matching approaches. First, the major criticism leveled against the Boolean approach is that its queries are difficult to formulate. Second, the Boolean approach makes it possible to represent structural and contextual information that would be very difficult to represent using the statistical approaches. Third, the Partial Matching approaches provide users with a ranked output, but these ranked lists obscure

Key Problems	Possible Solutions
Selection of Search Vocabulary	<ul style="list-style-type: none"> <li>• Thesaurus</li> <li>• Latent Semantic Indexing</li> </ul>
Search strategy (re)formulation	<ul style="list-style-type: none"> <li>• Smart Boolean</li> <li>• Statistical &amp; Linguistic Approaches</li> <li>• Thesaurus</li> <li>• Graphical Interfaces</li> </ul>
Information Overload	<ul style="list-style-type: none"> <li>• Ranking</li> <li>• Clustering</li> <li>• Visualization</li> </ul>

**Table 2.6:** lists some of the key problems in the field of information retrieval and possible solutions.

valuable information. Fourth, recent retrieval experiments have shown that the Exact and Partial matching approaches are complementary and should therefore be combined [Belkin et al. 1993].

In Table 2.6 we summarize some of the key problems in the field of information retrieval and possible solutions to them. We will attempt to show in this thesis: 1) how visualization can offer ways to address these problems; 2) how to formulate and modify a query; 3) how to deal with large sets of retrieved documents, commonly referred to as the information overload problem. In particular, this thesis overcomes one of the major "bottlenecks" of the Boolean approach by showing how Boolean coordination and its diverse narrowing and broadening techniques can be visualized, thereby making it more user-friendly without limiting its expressive power. Further, this thesis shows how both the Exact and Partial Matching approaches can be visualized in the same visual framework to enable users to make effective use of their respective strengths.

---





# CHAPTER 3

## INFOCRYSTAL

### 3.1 Introduction

How can we visualize how the contents of a large and abstract information space are related to multiple interests specified by the user ? We begin to answer this question by first addressing the question of how to visualize all the possible relationships among  $N$  concepts. Towards that end we will develop the discrete version of the InfoCrystal. The goal of this chapter is to demonstrate how the InfoCrystal can be used as a visualization tool that shows how the contents of an information space are related to a set of specified concepts. In particular, we will revisit the example presented in the introduction to show this. Second, we will demonstrate in the next chapter how the InfoCrystal can be used to formulate *Boolean queries* graphically. We will also show how the InfoCrystals can be used as building blocks and integrated in a hierarchical structure to formulate arbitrarily complex queries. Third, we will show in a subsequent chapter how users can assign *relevance weights* to the concepts and set a *threshold* to select relationships of interest. This enables users to formulate weighted Boolean queries. Fourth, we will describe the *rank layout* and the *bull's-eye layout* principle that visualize an InfoCrystal so that the relationship with the highest rank or the one with the largest relevance score will lie in its center, respectively. Finally, in a subsequent chapter we will show how the InfoCrystal can be generalized to visualize Partial Matching retrieval methods. Hence, we will demonstrate that the InfoCrystal can be used both as a visualization tool and visual query language.

### 3.2 2D versus 3D Visualization

Before addressing the question of how to visualize relationships, we want to briefly motivate our deliberate decision to use "only" a two-dimensional display to solve the problem statements of this thesis. Three-dimensional

---

displays are visually very appealing and they have the power to dazzle users. This is certainly one of the reasons that there is currently a great rush to enhance information displays with 3D computer graphics, especially as the cost for the needed computer power and speed continues to decrease. In the case of scientific visualization, where the data commonly has its origin in a three-dimensional physical space, this choice makes a great deal of sense. However, in the case of abstract information spaces the use of 3-D requires a more careful justification.

Three-dimensional displays are ideally suited for representing information spaces that satisfy the same constraints that govern the physical world for which our visual system has been optimized. As stated previously, the visible physical world consists mostly of smooth surfaces, whose visual properties change smoothly across them, except at object boundaries. The human visual system uses two-dimensional projections to reconstruct the three-dimensional world. It follows that there will be information that is not visible from a given point of view. Hence, three-dimensional displays require users to shift their point of view to see the information that is currently occluded, causing other information to become occluded. The human visual system uses the way things come into or go out of view at object boundaries to make inferences about the visual world [Spoerri 1991].

Many abstract information spaces do not satisfy the smoothness constraint. They present a special challenge for information visualization because they will cause visual discontinuities that are spurious, especially when we have to shift our point of view in a three-dimensional display. Hence, we choose for now to use a two-dimensional display to limit the creation of misleading visual discontinuities. Further, we want to investigate how much information can be "squeezed out" of a two-dimensional display. Once this has been firmly established, we want to investigate how we can add the third dimension to support the visualization in an appropriate way.

### **3.3 Visualizing Relationships**

How can all the possible combinations or relationships among several search criteria be visualized in a two-dimensional display? A common approach is to use Venn diagrams to visualize set relationships by intersecting geometric shapes that represent each set. There is a common misconception that it is

---

not possible to generate such Venn diagrams that can represent all the possible relationships for any number of sets. There exist constructive proofs that show how we can use convex, but not circular shapes to generate Venn diagrams that represent all the possible relationships between  $N$  concepts, but the visual areas corresponding to the different relationships become increasingly small and difficult to identify as the number of concepts increases [Humphries (1987), Anderson 1988)]. Hence, it is difficult to represent all the possible relationships among more than three concepts in a visually compact and simple way.

We will now demonstrate how we can move beyond the Venn diagram approach so that all the possible relationships among  $N$  variables can be represented in an elegant way. Figure 3.1 shows how a Venn diagram of three intersecting circles can be transformed into an iconic display. We start out by exploding the Venn diagram into its disjoint subsets. Next, we represent the subsets by icons whose shapes reflect the number of criteria satisfied by their contents, also called the *rank* of a subset. Finally, we surround the subset icons by a border area that contains icons, also called *criterion icons*, that represent the original sets.

The goal is to arrive at a representation that lets users use their visual reasoning skills to establish how the interior icons are related to the criterion icons. The following visual coding principles are used in a redundant way:

- **Shape Coding:** is used to indicate the number of criteria that the contents associated with an interior icon satisfy (i.e., one -> circle, two -> rectangle, three -> triangle, four -> square, and so on).

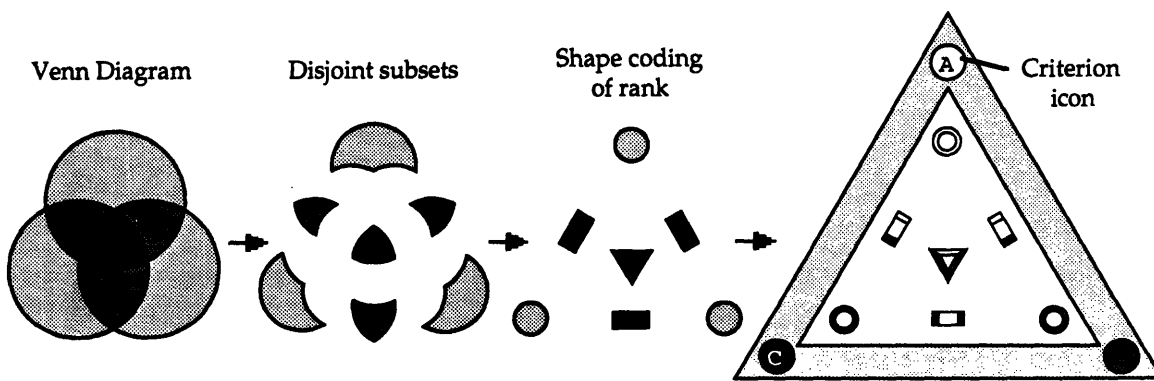


Figure 3.1: shows how we can transform a Venn diagram into an iconic display, called the *InfoCrystal*.

- **Proximity or Location Coding:** The closer an icon is located to a criterion icon, the more likely it is that the icon's contents are related to it.
  - **Rank Coding:** Icons with the same shape are grouped in "invisible" concentric circles, where the rank of an icon is equal to the number of criteria satisfied and the rank increases as we move towards the center of an InfoCrystal.
  - **Color or Texture Coding:** is used to indicate which particular criteria are satisfied by the icon's contents.
  - **Orientation Coding:** The icons are positioned so that their sides face the criteria that they satisfy.
  - **Size Coding:** is used to visualize quantitative information, i.e., the number of elements represented by an icon.
- There is also the possibility to use brightness and saturation to represent quantitative information. We have not considered these two perceptual dimensions in the current implementation of the InfoCrystal, but we plan to do so in the future.

Figures 3.2 to 3.10 show InfoCrystals that consist of three, four, five, six, seven, eight, nine and thirteen search criteria<sup>1</sup>, respectively. The reader should keep in mind that we are limited in this text document to use black and white textures to indicate the different criteria, whereas the use of color does greatly facilitate the ready interpretation of the InfoCrystal with more than four criteria. As these figures show, the number of possible combinations or relationships among  $N$  different criteria grows exponentially and it is equal to  $2^N$ . We visualize  $2^N - 1$  of these possible relationships and we choose not to visualize the relationship that specifies documents that satisfy none of the criteria. One of the objectives of the InfoCrystal is to enable users to explore an information space along several dimensions simultaneously; or to use another metaphor, we want users to be able to juggle multiple concepts without becoming too overwhelmed by the resulting complexity.

The user can choose to visualize the interior icons so as to emphasize the *qualitative* or the *quantitative* information associated with them. If users are interested in how the interior icons are related to the criterion icon, then they

---

<sup>1</sup> We will use the following terminology interchangeably to refer to the different inputs of an InfoCrystal: search criteria, input concept and user interest.

---

can display them in a variety of different styles, as shown in Figures 3.2 to 3.10: 1) as polygons with colored or textured borders, where they can make use of the location, shape, color/texture and orientation coding cues to infer the icon's precise relationship to the criterion icons; 2) as simple polygon outlines, where users receive location, shape and orientation coding cues; 3) as small circular place holders, where they only receive location coding cues. If, however, users want to visualize quantitative information, i.e., the number of documents associated with the interior icons, then the icons can be represented as simple numbers or as circular pie-chart icons whose sizes reflect the numerical information (see Figures 3.3, 3.11, 7.3 to 7.12, 7.14, 12.2). The pie-chart icons are similarly oriented as the polygon icons and the colors or textures of their slices indicate which criteria are satisfied.

### 3.4 Rank Layout Algorithm

We have developed a layout algorithm that enables us to generate InfoCrystals with  $N$  inputs. The objective of this algorithm is to create a layout of the interior icons, where none of their locations coincide. We call it the *rank layout* principle, because it strictly enforces the rank coding principle: the number of criteria satisfied by an interior icon increases as we move towards the center of the InfoCrystal; and users can expect to find the icon with the highest rank in the very center.

The computation of the rank layout involves the following steps, although there are exceptions that we will address below: First, we specify  $N$  circular bands of equal width within which the icons with the same rank have to be placed. Second, we compute a *center of gravity* for each icon as follows: we define a two-dimensional vector pointing from the center of the InfoCrystal to each criterion icon that is satisfied by an interior icon. We take these vectors to compute their center of gravity, which is equal to the averaged sum of all the vectors. Third, we compute for all icons with the same rank the distance of their center of gravity from the InfoCrystal's center. Next we determine how many distinct distance values there are for the icons with the same rank and we subdivide their corresponding circular band accordingly to define a series of circles lying within in this band. Each icon is assigned a circle on which it needs to be placed. Fourth, we define a straight line that passes through the center of gravity of an icon and the InfoCrystal's

---

center. We will place the icon where this line intersects the circle on which the interior icon has to lie. Finally, we orient the icon in such a way to minimize the angle between the normal to the side that corresponds to a particular criterion that is satisfied and the vector that points from the icon's location to that criterion.

There are exceptions to this general algorithm that occur predominantly when we have an even number of inputs to an InfoCrystal. First, degenerate cases occur when the center of gravity of an interior icon coincides with the center of the InfoCrystal. Hence, we can not compute the distance and specify the straight line. For each interior icon polygon, we can define a figure whose corners correspond to the criterion that are satisfied by the icon in question. The degenerate case occurs when this figure is symmetrical. We solve this problem by differentiating between the cases where the number of criteria satisfied, i.e., the rank, is either odd or even. If the rank is odd, then we place a duplicate in each direction that points towards a criterion icon that the interior icon satisfies. If the rank is even then we compute the major axis of symmetry and we place a duplicate where this axis intersects the circle on which the icon has to lie. In addition, if the rank is a multiple of four, then we place duplicates where both the major and minor axes intersect the circle (see Figure 3.7).

Second, in the case of an interior icon of rank two that involves non-adjacent criterion icons, the algorithm outlined above would place the interior icon closer to a criterion icon not related to it than to the criterion icons that it is actually related to (see Figure 3.5). Instead, we choose to duplicate these icons of rank two, so that they are as close as possible to their related criterion icons as well as at the correct distance from the center. Their locations are computed by intersecting the circle on which these duplicate icons of rank two have to lie with the straight line that connects the two non-adjacent criterion icons.

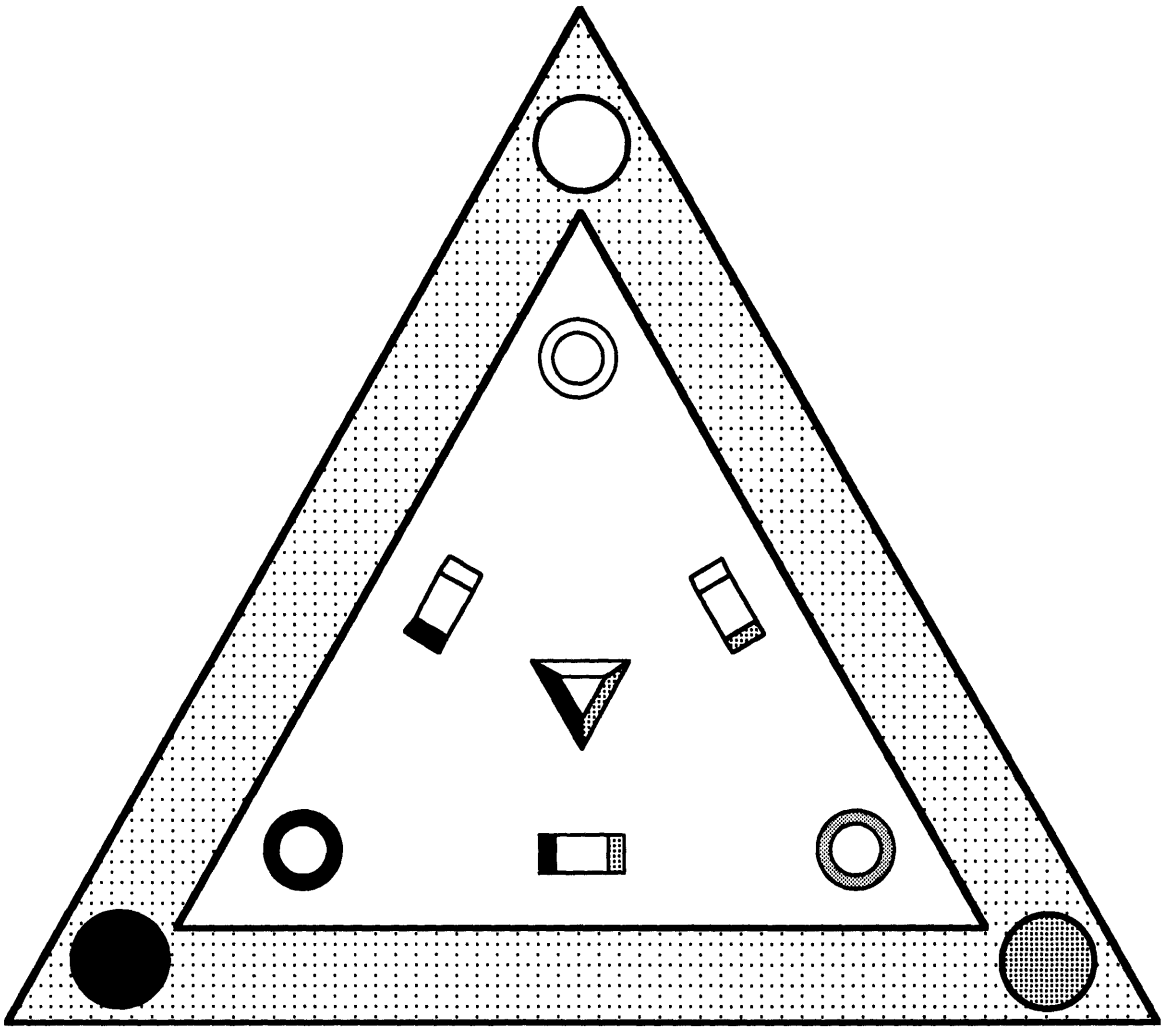
Third, we distinguish between the following cases if we have an even number of criteria. We begin by testing if the figure defined by the criterion icons that are satisfied, called positive criterion icons, possesses an even or odd axis of symmetry. An even symmetry implies that the axis passes through a midpoint between two consecutive positive criterion icons. An odd axis of symmetry implies that the axis passes through one of the criterion

---

icons. We place the interior icon where the axis of symmetry intersects the circle on which the interior icon has to lie and we choose the intersection point that is closer to the center of gravity. If no axis of even or odd symmetry exists, then we parse the ordered list of positive criteria into segments of consecutive numbers. We then calculate the gap between these segments to identify the one that has the largest gaps on either side, which we call the most isolated segment. We calculate the center of gravity for the most isolated segment and test if it coincides with the center of gravity of the positive criterion icons. If they do not coincide, then we can use these two points to define line and we choose the intersection point with the circle on which the interior icon has to lie that is closer to the center of gravity.

We do not claim to have found an absolute solution that will never cause the icons to be mapped to identical locations as the number of input criteria increases. We have focused our energies to devise a layout algorithm that will place the interior icons in different locations, except for very few exceptions (see Figure 3.8), when we have not more than ten concepts that we want to juggle at the same time.

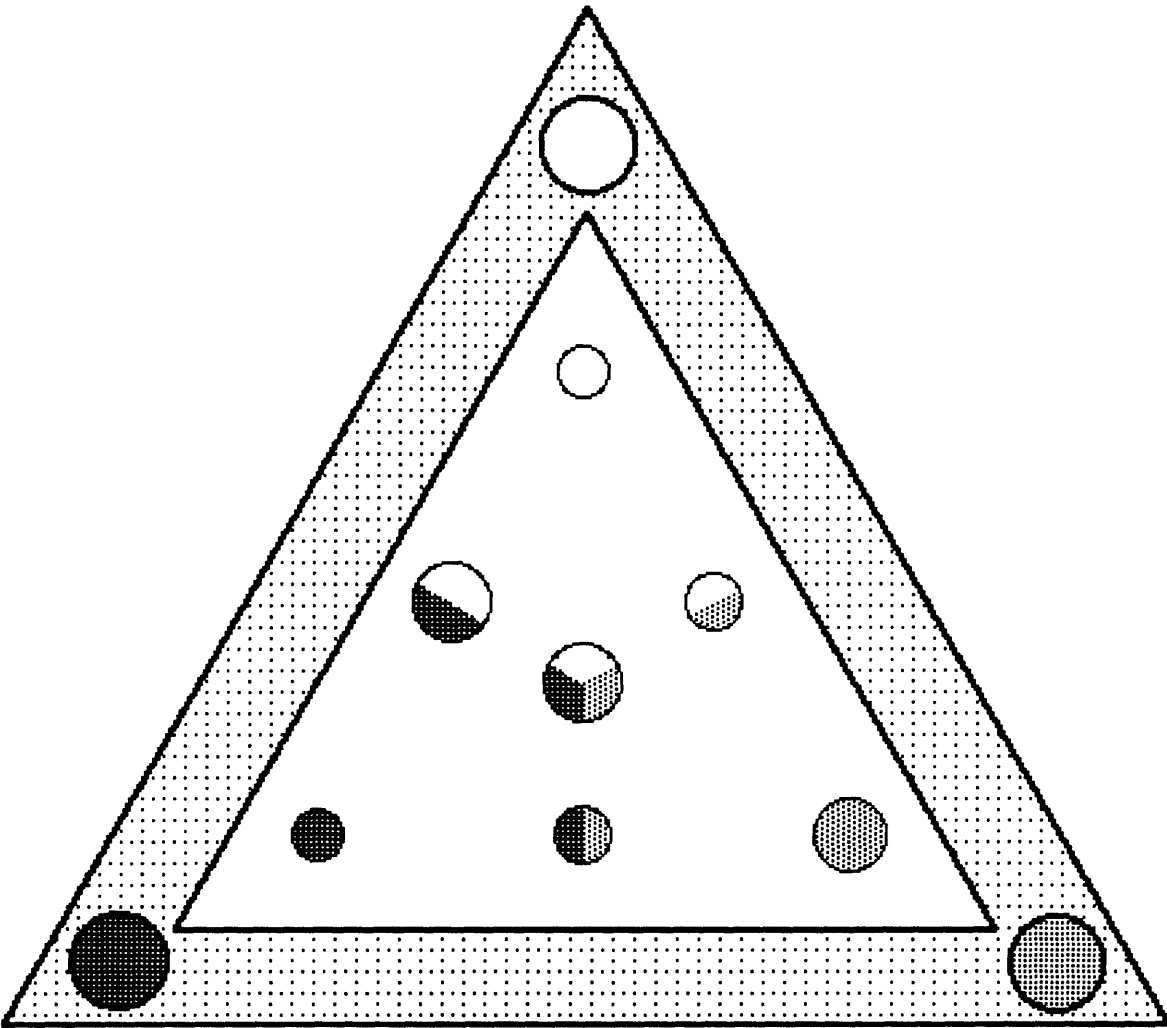
---



**Figure 3.2:** shows the InfoCrystal that visualizes the possible relationships among **three** search criteria. This figure can serve to illustrate a visual strategy that users can use to read a crystal: they can think of a border or criterion icon as a colored light source, and only the icons that are related to that criterion have a side facing it and hence are able to reflect back its colored light.

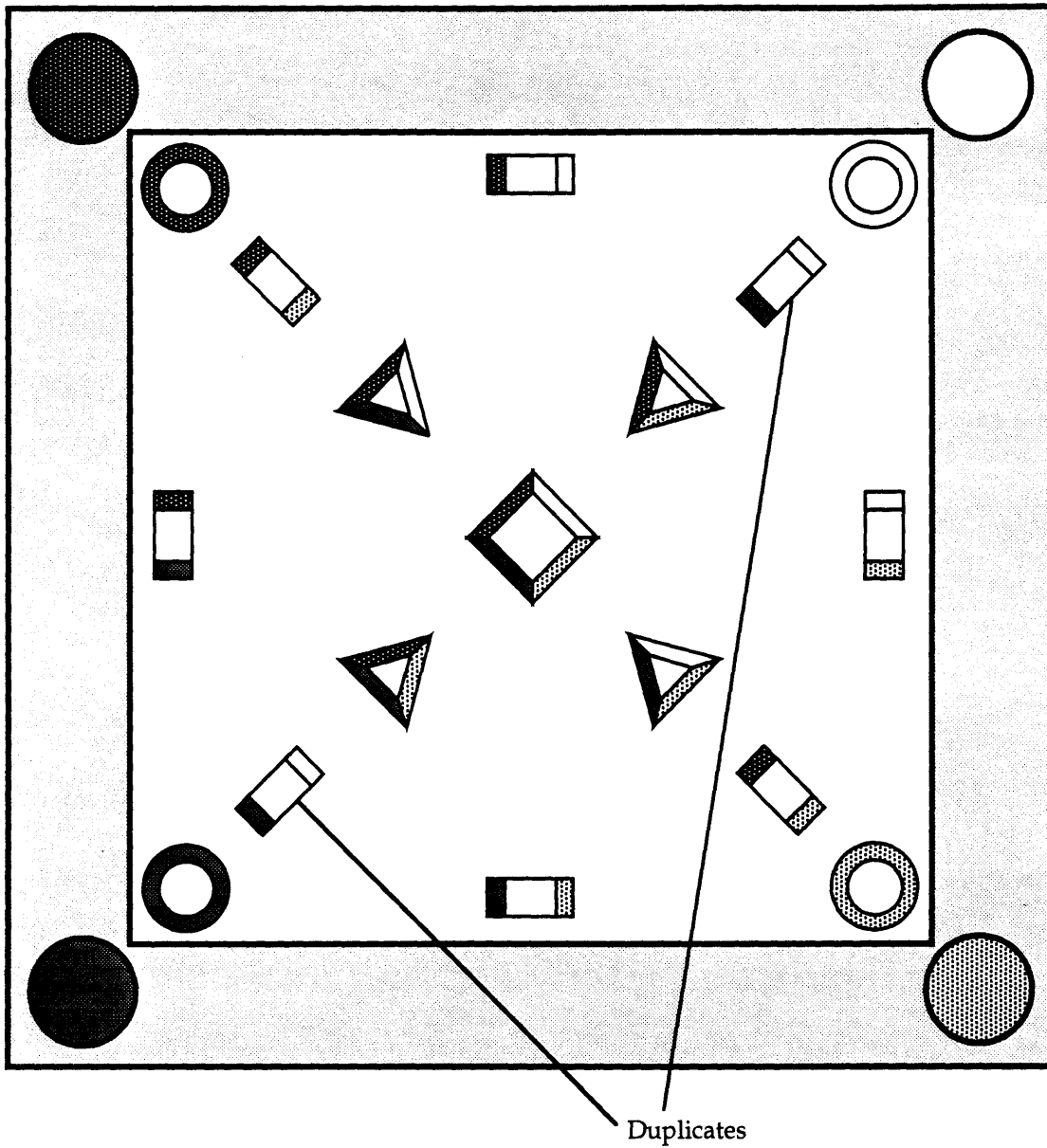
---



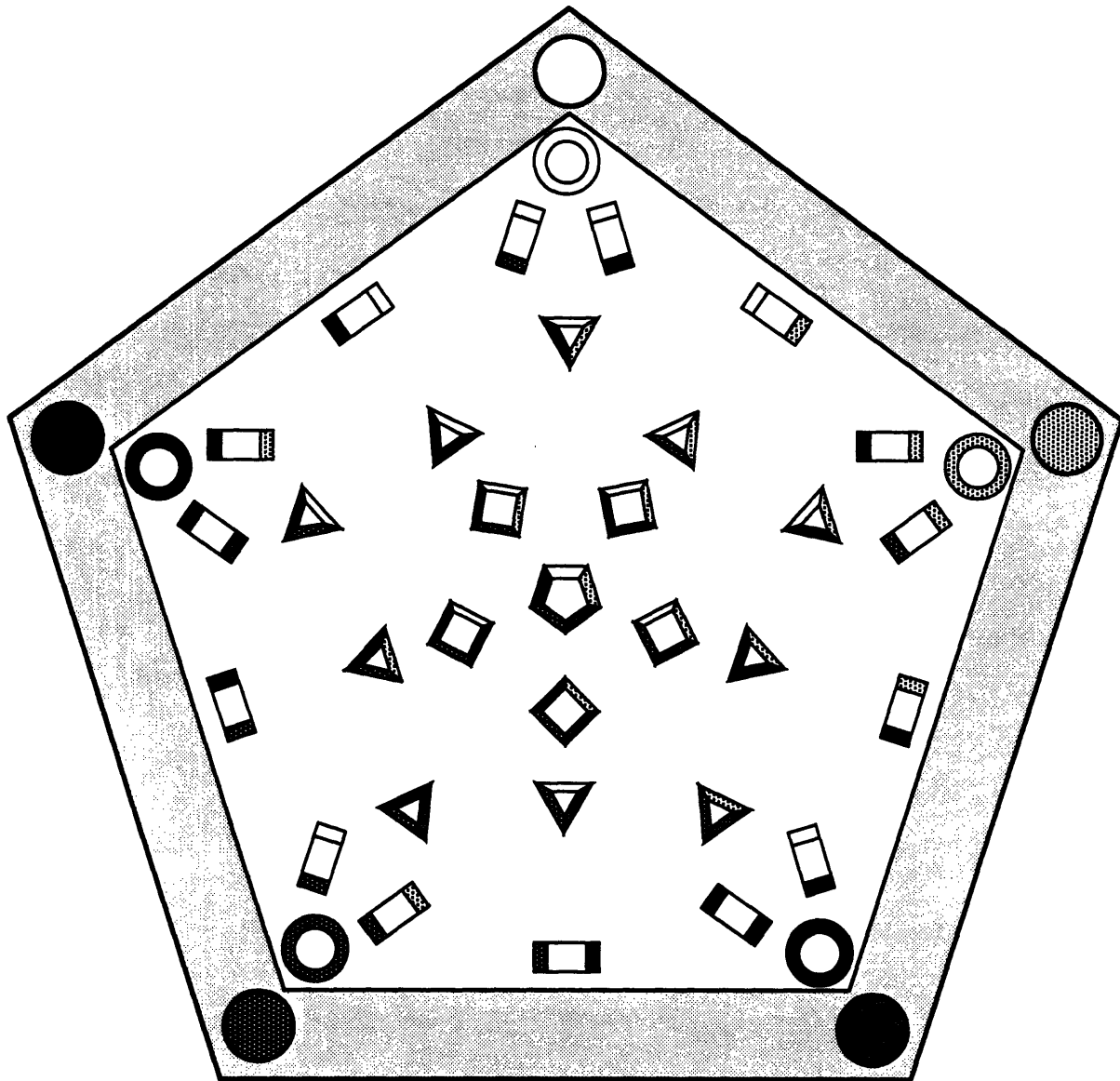


**Figure 3.3:** shows an InfoCrystal that visualizes the quantitative information associated with the interior icons, using the pie-chart style that employs size coding to reflect the quantitative information and the texture or color of the pie slices indicate which criteria are satisfied.

---

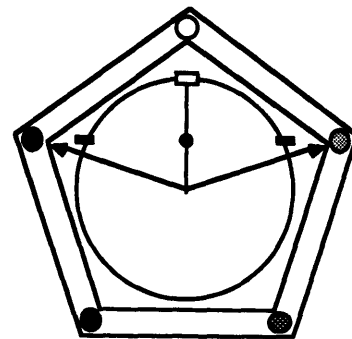


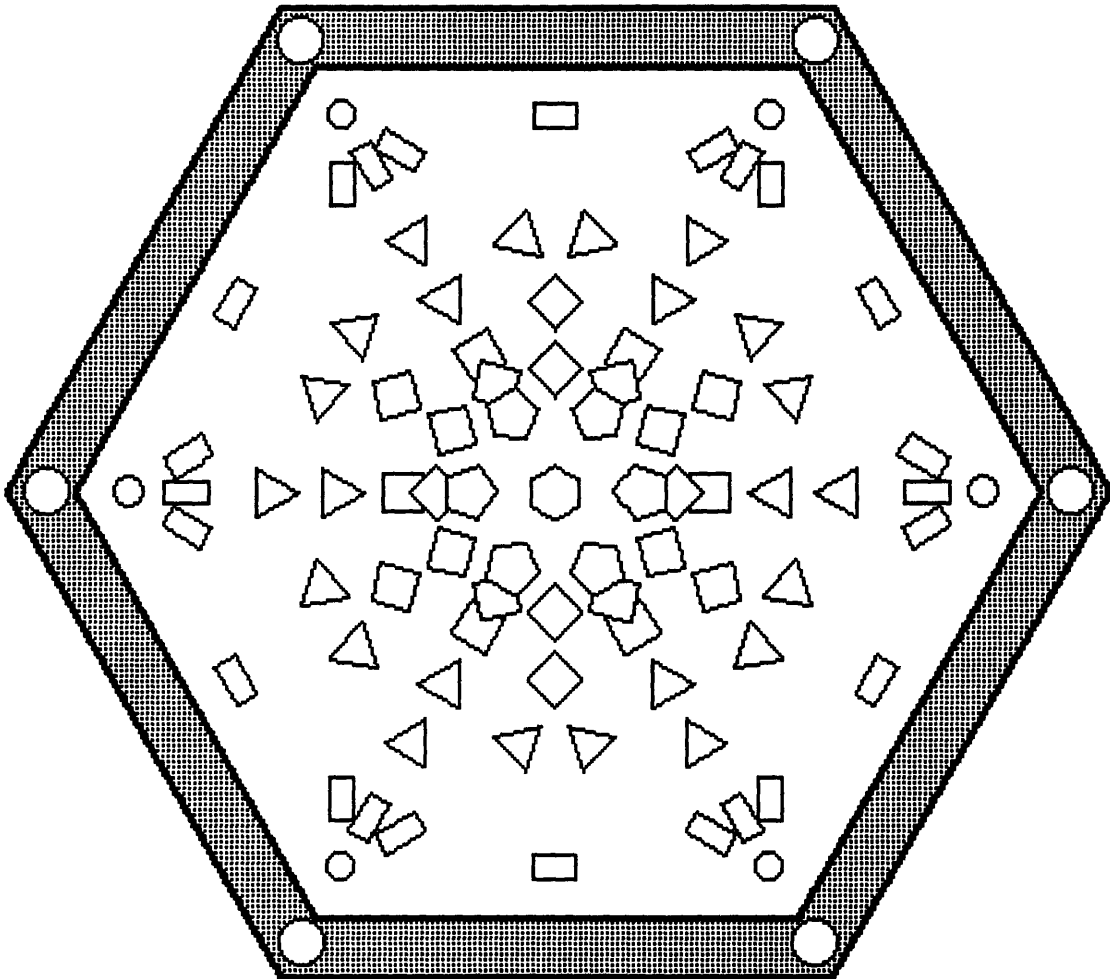
**Figure 3.4:** displays an InfoCrystal that involves **four** criteria. This crystal is the first one where we choose to duplicate certain icons, because there are icons whose center of gravity coincides with the center of the InfoCrystal. These degenerate cases occur for criterion icons that lie diagonally opposite each other. We resolve these degenerate cases by placing the interior icons so that they are close to their related criterion icons as well as at the correct distance from the center.



**Figure 3.5:** shows the InfoCrystal that visualizes all the relationships among five criteria. It is worth stressing again that the use of color would greatly facilitate the rapid interpretation of an InfoCrystal that "juggles" more than four criteria simultaneously.

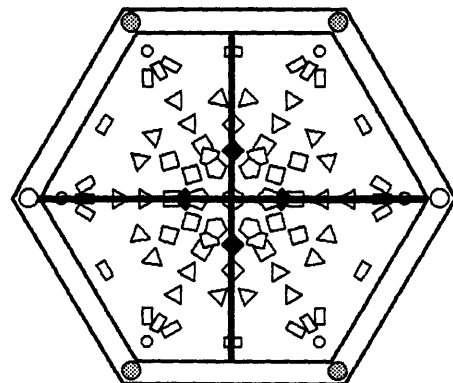
Although none of the centers of gravity of the icons coincide with the center of the InfoCrystal, we elect to duplicate the icons of rank two that involve non-adjacent criterion icons, because we want them to be as close as possible to their related criterion icons as well as at the correct distance from the center. The adjacent figure shows the two vectors that are used for a particular icon of rank two to compute the center of gravity, shown as a solid circle. If we were to place this particular icon based on where the line defined by its center of gravity intersects the circle on which the icon has to lie, then it would end up being placed much closer to a criterion icon that is not related to it than to the ones it is actually related to (shown as a solid white rectangle in adjacent figure). Instead we place this icon of rank two where the line connecting its two criterion icons intersects the circle (shown as solid black rectangles in adjacent figure).

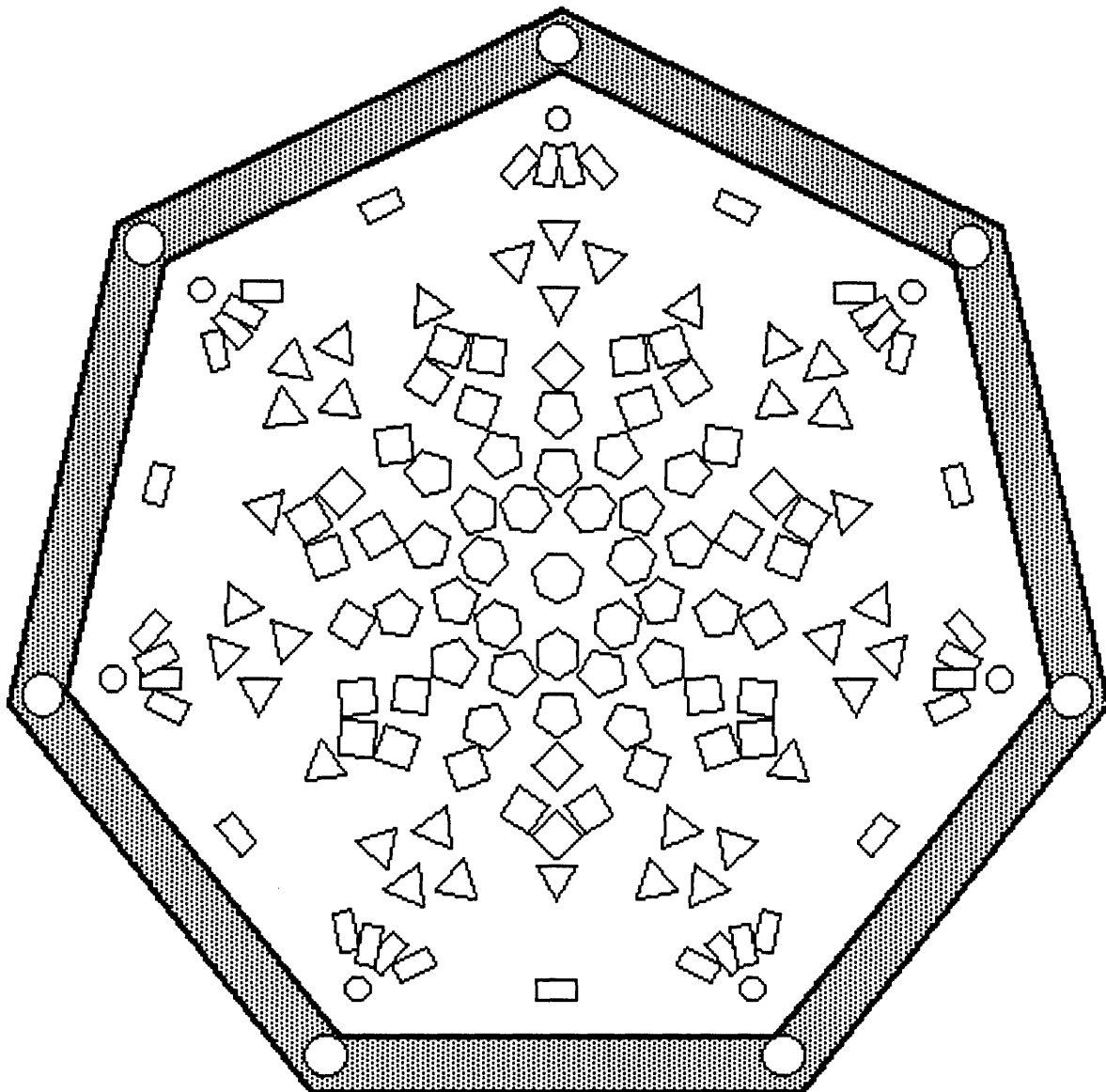




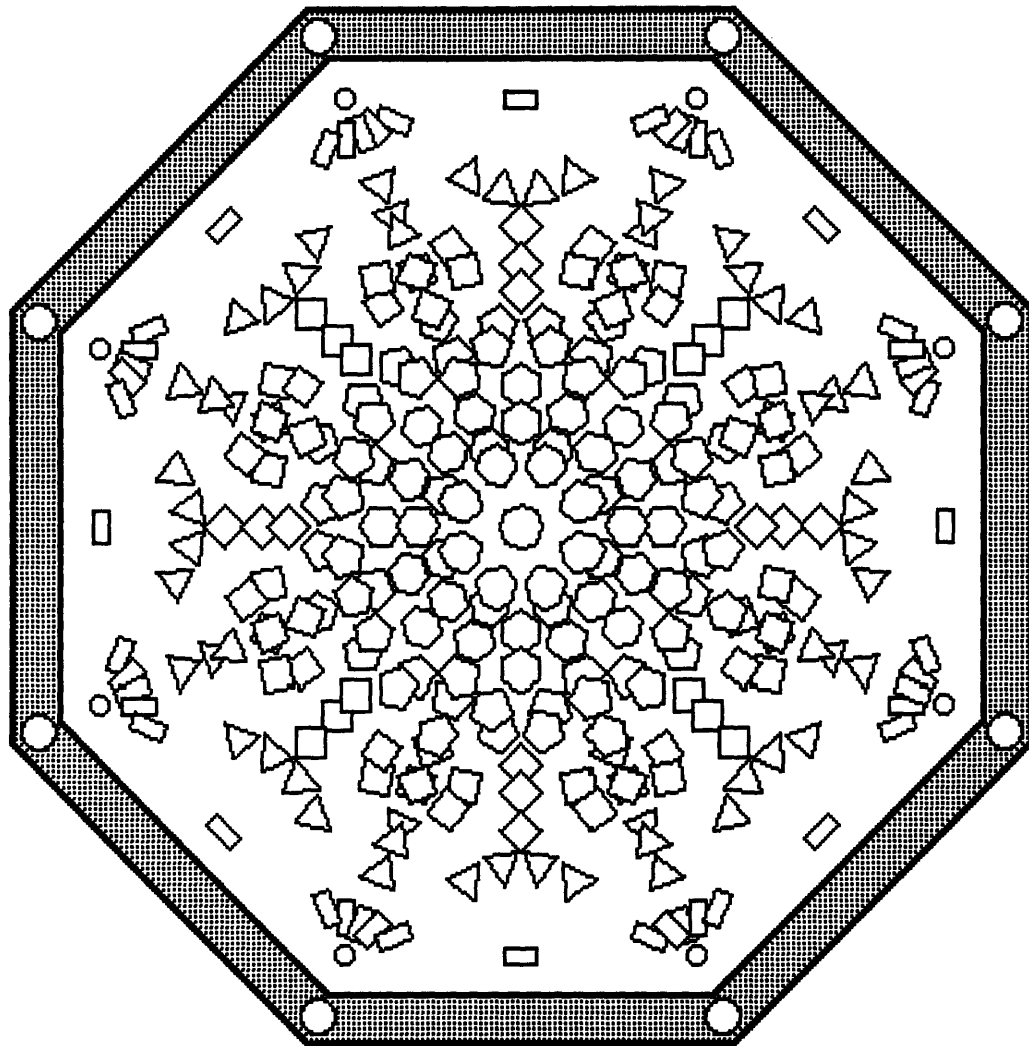
**Figure 3.6:** displays the InfoCrystal that visualizes the 63 different relationships among six search criteria, where at least one of the criteria is satisfied. In this crystal we display the icons only in the outline style, where users still can use location, shape and orientation coding to infer how the icons are related to the criterion icons.

Also for this InfoCrystal we choose to duplicate icons of rank two that involve non-adjacent criterion icons so that they are as close as possible to their related criterion icons as well as at the correct distance from the center. Furthermore, we have to duplicate all the interior icons whose center of gravity coincides with the InfoCrystal's center, which occurs because we have an even number of inputs and we have three axes of symmetry. In particular, we have three such icons of rank four for which we place four duplicates where both the major and minor axes intersect the circle. The adjacent figure shows the axes and the four duplicates for the relationship involving the four darkly shaded criterion icons.



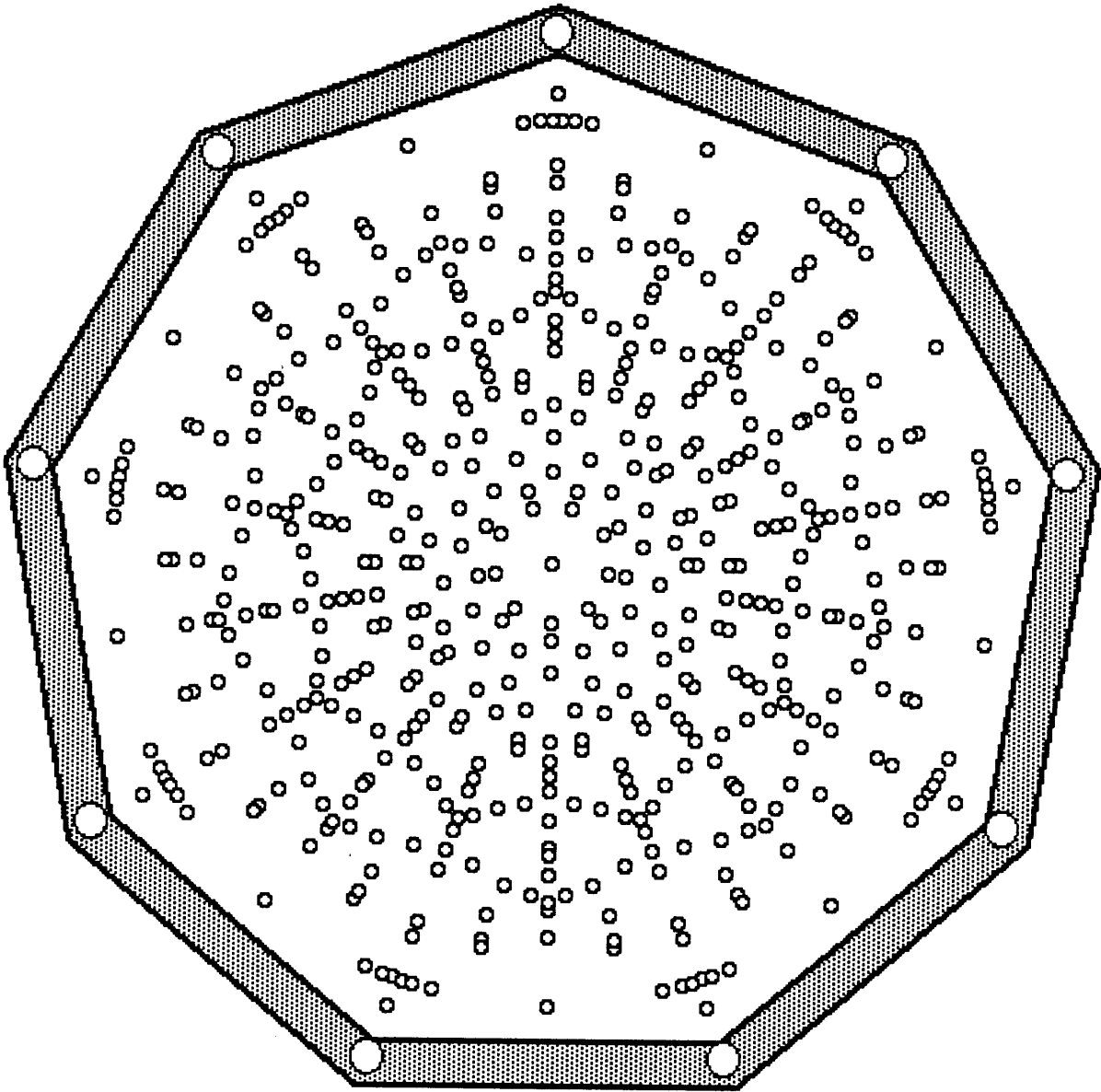


**Figure 3.7:** displays the InfoCrystal that visualizes the 127 different relationships among **seven** search criteria, where at least one of the criteria is satisfied. In this crystal we display the icons only in the outline style, where users still can use location, shape and orientation coding to infer how the icons are related to the criterion icons. Also for this InfoCrystal we choose to duplicate icons of rank two that involve non-adjacent criterion icons so that they are as close as possible to their related criterion icons as well as at the correct distance from the center.

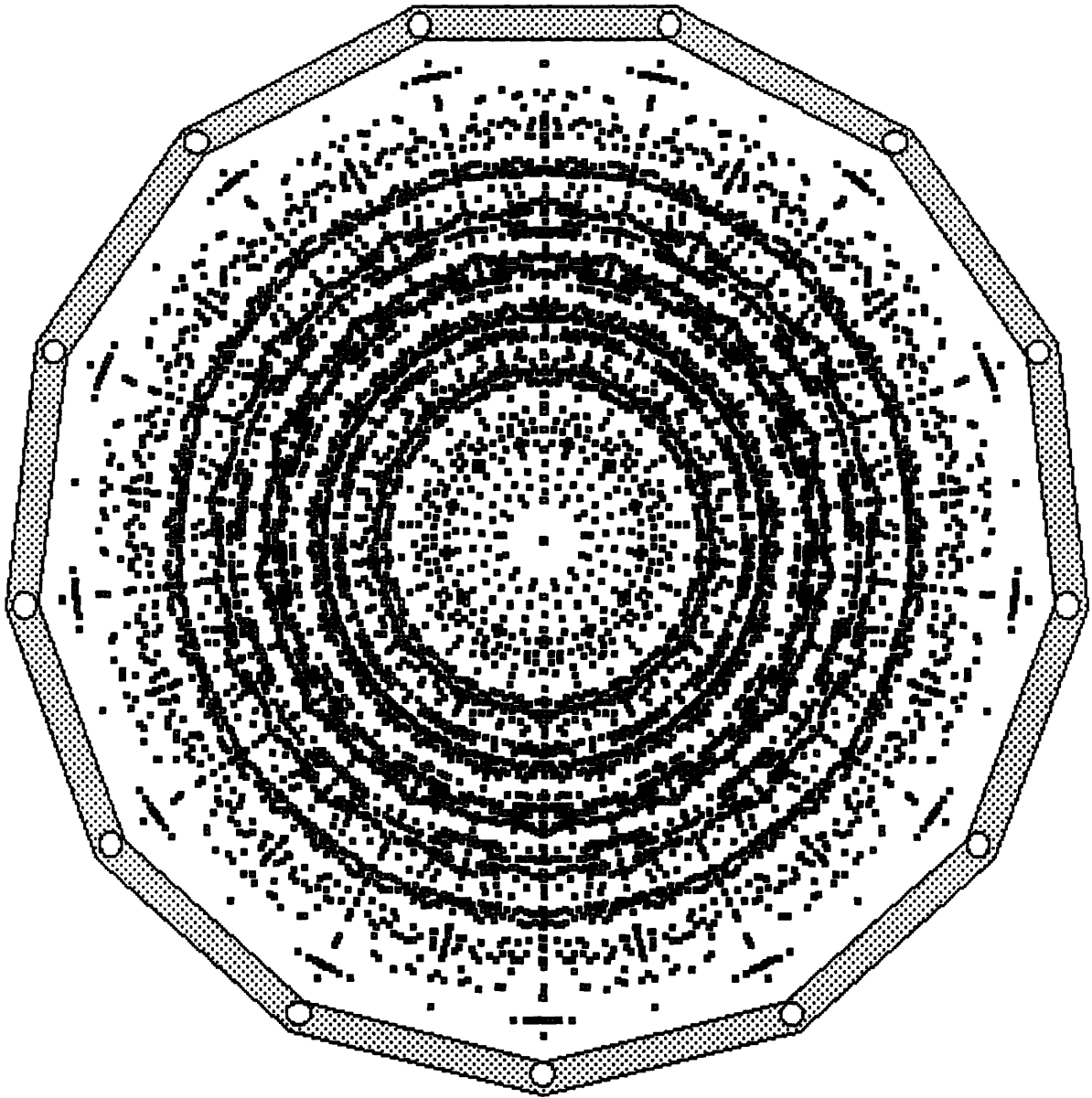


**Figure 3.8:** displays the InfoCrystal that visualizes the 255 different relationships among eight search criteria, where at least one of the criteria is satisfied. In this crystal we render the icons only in the outline style, where users still can use location, shape and orientation coding to infer how the icons are related to the criterion icons.

---



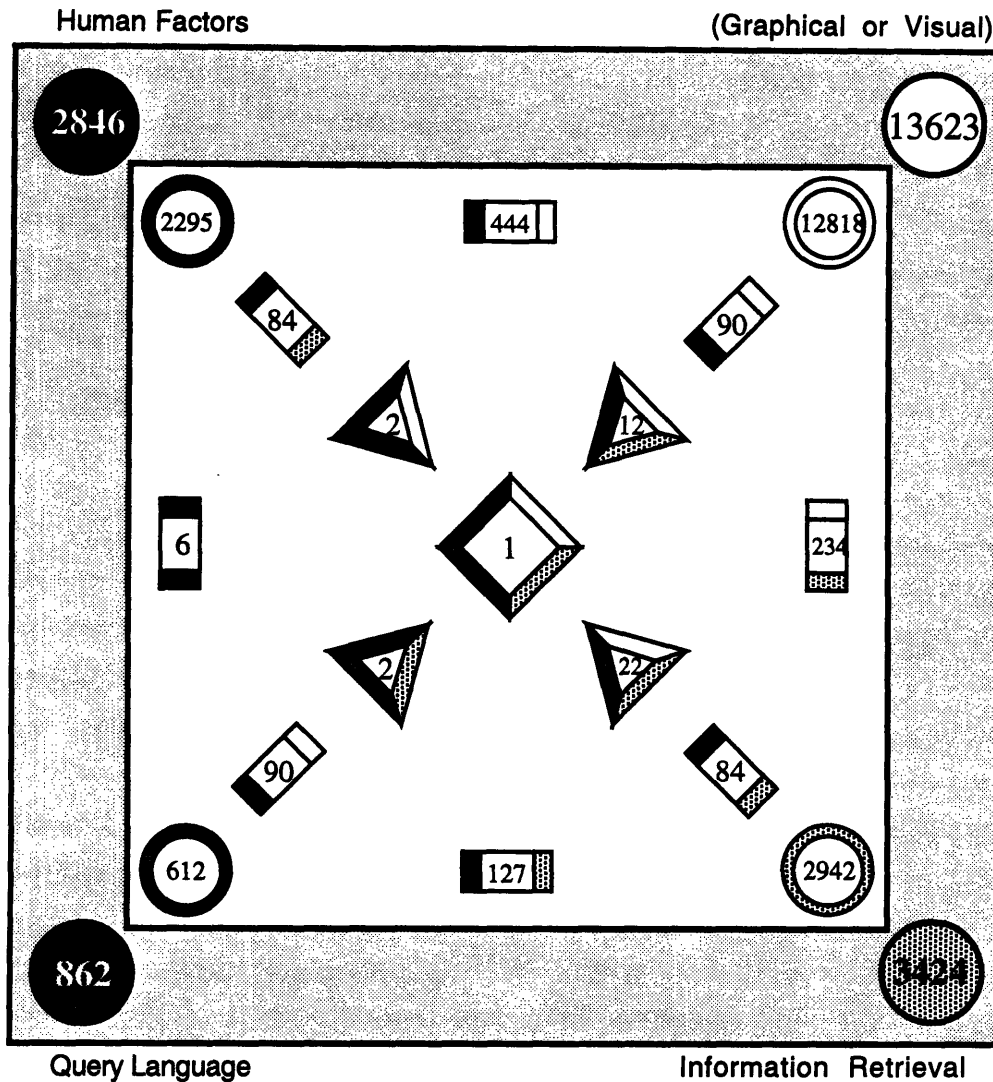
**Figure 3.9:** shows the InfoCrystal that displays all the 511 different relationships among nine search criteria, where at least one of the criteria is satisfied. In this crystal we visualize the icons only in the point style, where users can only use location coding to infer how the icons are related to the criterion icons.



**Figure 3.10:** shows the InfoCrystal that displays all the 8191 different relationships among thirteen search criteria, where at least one of the criteria is satisfied. In this crystal we visualize the icons only in the point style, where users can only use location coding to infer how the icons are related to the criterion icons. The purpose of this figure is to demonstrate how the developed rank layout algorithm is able to generate an InfoCrystal of that complexity. Although the number of concepts results in a complexity of relationships that is staggering and overwhelming, we can imagine, for example, marketing applications where we could use brightness and/or saturation coding to give users a rough sense of how the contents of a database distribute across the space of relationships of so many criteria.

---





**Figure 3.11:** The number associated with an icon indicates how many of the retrieved documents satisfy the relationships represented by it. A total of 19,691 documents was retrieved from the INSPEC Database (1991-92) that satisfy any of the four search criteria, but there is only one document that satisfies all the four criteria ! One of the advantages of the InfoCrystal is that it visualizes how the contents of a database distribute across the different possible relationships and thereby not locking users into just one way of viewing the data.

### 3.5 Example Revisited

We will now revisit the example presented in the section 1.5 to show how the InfoCrystal enables users to see in a single display how the database contents are related to the interests specified by the users. This type of visual feedback could help them to formulate a query that does not retrieve either too few or too many documents. Figure 3.11 displays how the contents of the INSPEC Database (1991-92) relate to the four displayed interests. The center

icon of the InfoCrystal represents the documents that satisfy all the four criteria. In our example there is just one document. We can easily broaden our focus of interest by examining the icons that surround the center icon and satisfy three of the four concepts. For example, there are 22 documents that are related to the (*Graphical OR Visual*), *Information Retrieval*, and *Query Language* concept but not to the *Human Factors* concept. If we want to move further away from our initial interest then we could explore the 6 documents that have been indexed under the *Query Language* and *Human Factors* concept but not under the (*Graphical OR Visual*) or *Information Retrieval* concept.

As the above discussion indicates, the InfoCrystal enables users to easily broaden or narrow their focus of interest. Users can represent their current interests by selecting the interior icons that capture it. The selected interior icons can be thought of as defining a "figure" and the not selected icons as representing the "ground". The InfoCrystal allows users to easily alter this figure-ground relationship. Hence, they are not locked into just one way of viewing the data, but they can explore an information space in a flexible and fluid way. The organization of the InfoCrystal ensures that users can easily infer how the retrieved documents relate to their interests.

We have discussed in chapter 2 how a modern Boolean query can be described along the following four dimensions: coordination, proximity, stemming, and field level. The number of retrieved documents can be changed by making the appropriate choices along these four dimensions. We have noted above that there is only one document that satisfies all four criteria in Figure 3.11. This could be changed by applying a stemming operation to the terms used to search the INSPEC database. Further, we could relax the proximity requirements for the search concepts that involve multiple words and we could search over all fields to obtain more documents that satisfy all four criteria. In chapter 4 we will show how changes along these four dimensions can be specified in a visual way.

---

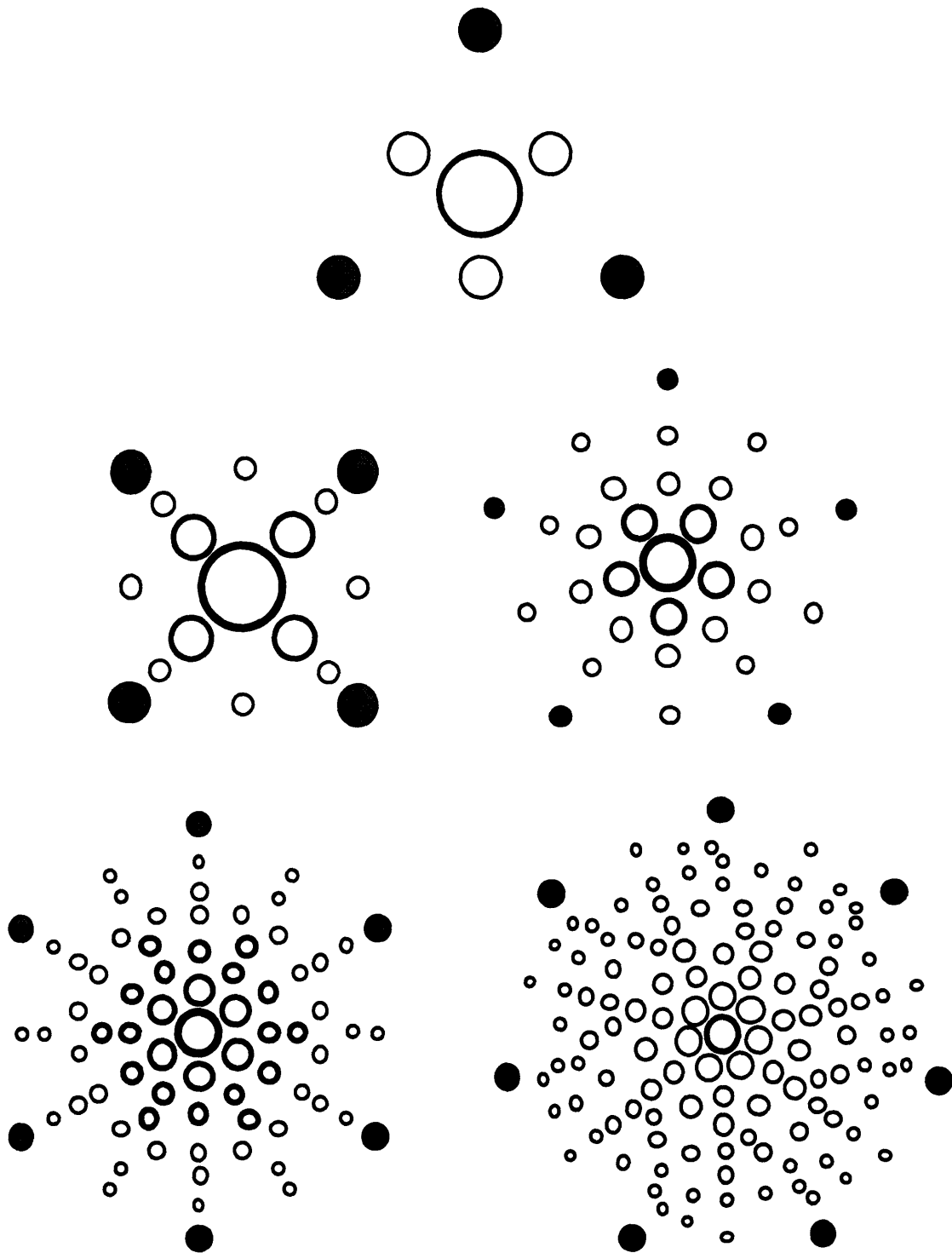
## 3.6 The Design Process of the InfoCrystal

The transformation process depicted in Figure 3.1, where we show how the familiar circular Venn diagrams can be translated into the InfoCrystal, is visually very compelling and memorable. However, it is also misleading because it does not reflect the way we developed the InfoCrystal. If we had chosen the path of attack suggested by Figure 3.1, where we explode circular Venn diagrams, then we might have not been able to create a presentation that can visualize all the relationships among more than three entities. Figure 3.1 serves as a visually compelling bridge between the familiar circular Venn diagrams and the novel InfoCrystal representation. We will now present earlier sketches, beginning with the very first designs and ending with most recent ones before the final version of the InfoCrystal, to give a flavor of the design process of the InfoCrystal. This design history is instructive because it highlights some of the used visual coding principles and their effectiveness. Further, it shows alternative ways of visualizing the relationships among several entities.

### 3.6.1 The First Designs for the InfoCrystal

Rooted in our background in computational vision, we used the location and proximity grouping principle to guide us in the initial designs. We started out by placing a set of search interests on the computer screen. We then imagined that these interests would act like magnets that attract the relevant information, thereby leading to a compact overview of how the contents of a library were related to our specified interests. We wanted a representation that enabled us to focus on specific relationships without forcing us to abandon our sense of overview. Most of all, we want to be able to simultaneously "juggle" as many interests as possible: three balls at the same time - you must be joking! - four, five, ... as many as we can barely keep up in the visual space. Figure 3.12 shows these initial designs.

---



**Figure 3.12:** shows the first sketches of the InfoCrystal with up to seven inputs, where we used the location and proximity grouping principle as the key design and organizing principle.

---

---

### 3.6.2 InfoCrystal Networks

The next figures show how we built on the initial designs by introducing connecting lines as a further coding principle to visualize all the possible relationships among multiple entities. The filled circles perform a dual function: 1) they represent the reference entities; 2) they represent the relationships of rank one. The connecting lines make explicit how a circle is related to the reference entities. The rank of a circle is encoded by its size and to some degree by the number of lines emitting from it.

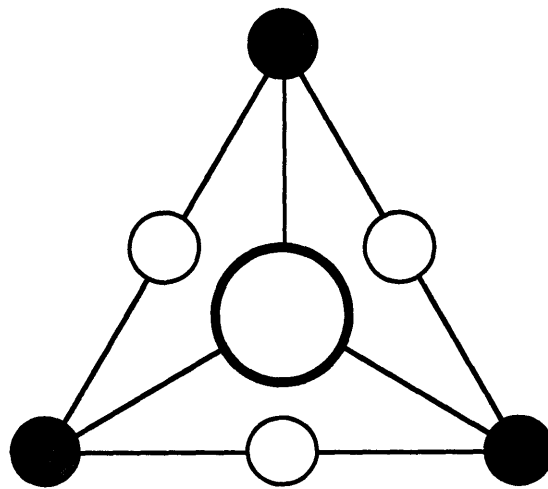


Figure 3.13: shows how the InfoCrystal with three inputs can be visualized as a network.

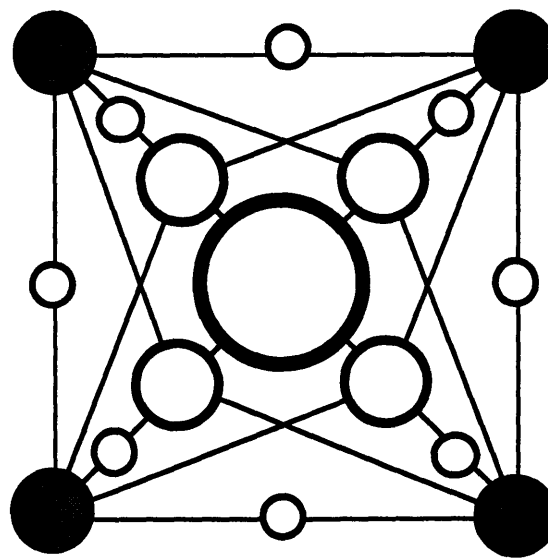


Figure 3.14: shows how the InfoCrystal with four inputs can be visualized as a network.

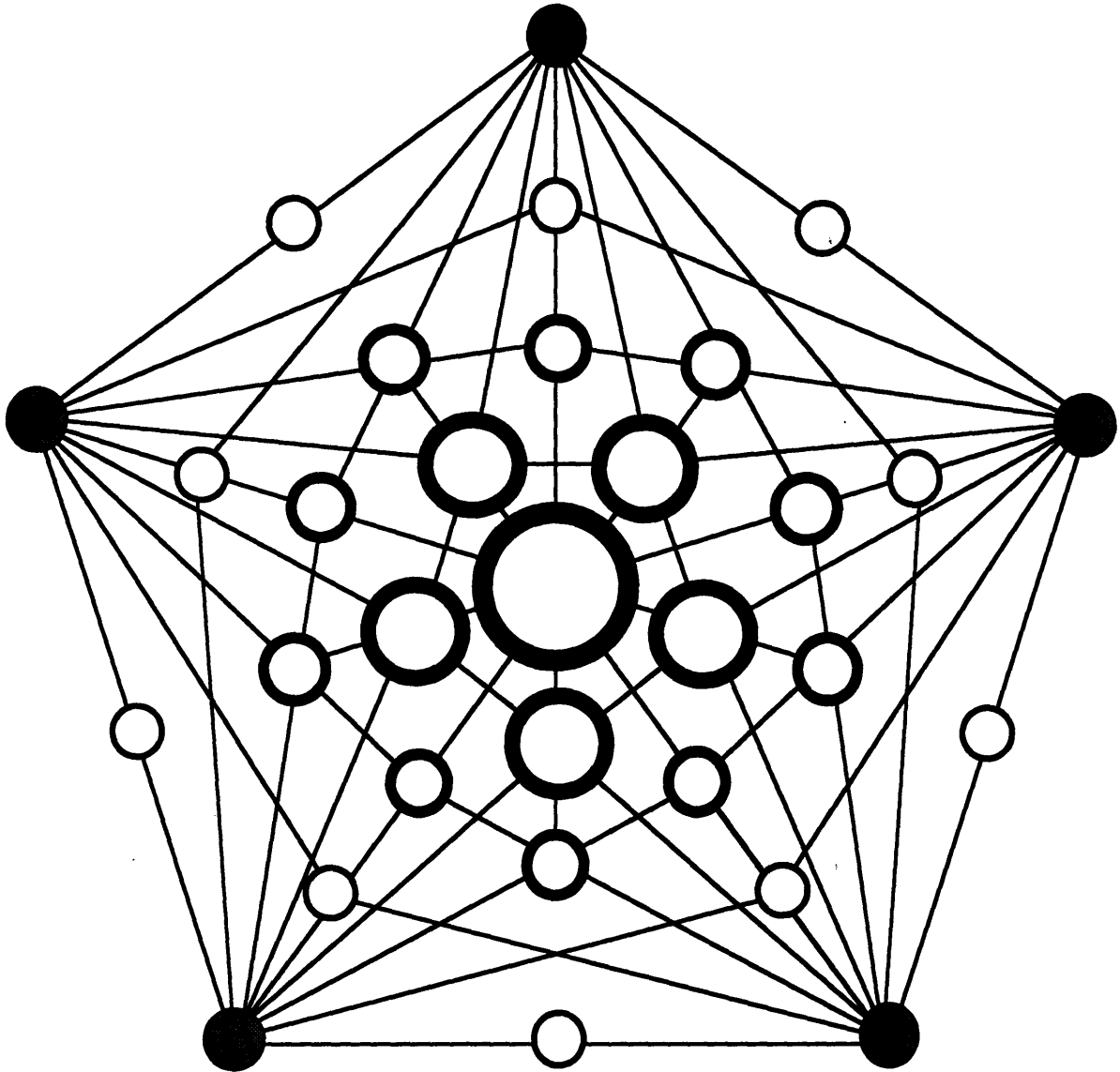


Figure 3.15: shows how the InfoCrystal with five inputs can be visualized as a network.

---

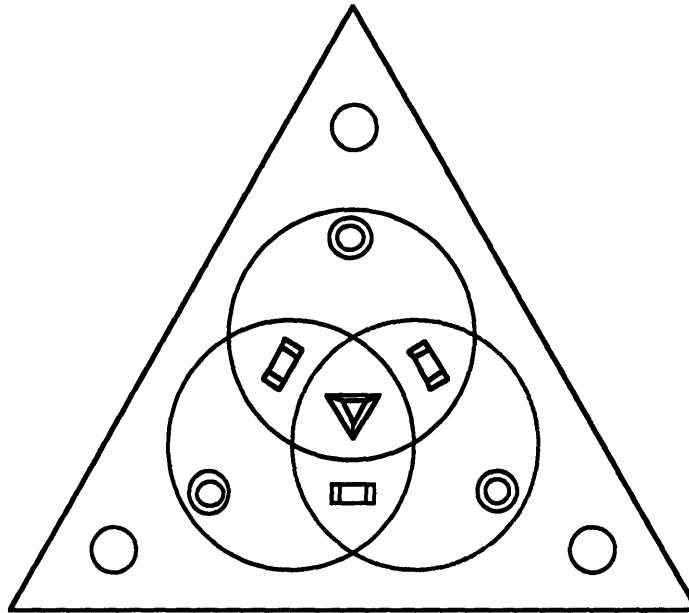
### 3.6.3 Combining the InfoCrystal with Venn Diagrams

Once the locations of the icons, which represent the different relationships, had been figured out, we started to add other visual grouping principles, such as shape, color and orientation, to facilitate the visual interpretation of the designs. The visual appearance of the interior icons in turn had such a strong "button" appearance that it led us to interpret the InfoCrystal as a keyboard. This opened up the door of using the InfoCrystal not only as visualization tool, but also as a visual query language, as we will show in the chapter 4.

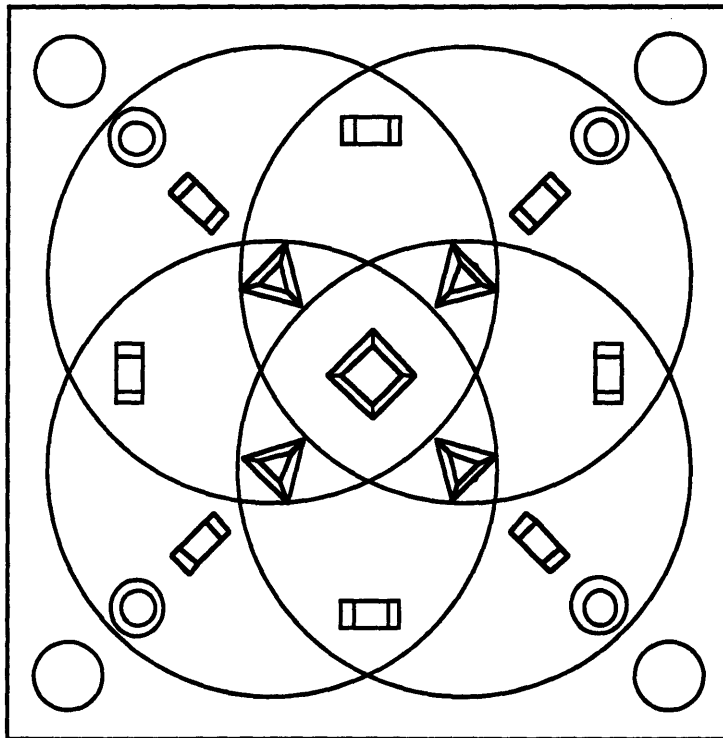
---

Next, we considered using circular Venn diagrams as an additional visual organizing principle. We were wondering if there could be a way of visually enclosing all the interior icons that are related to a particular criterion icon. We started to draw bounding lines, and because we have preference for symmetrical and classical design we used circles. This reminded us of the Venn diagrams that we had encountered during our schooling. However, if we had used the Venn diagrams as our starting point, then we might have "hit a wall", once we would have tried to visualize the relationships among four entities. It is impossible to represent all the relationships among four sets if we want to use circles to represent the sets (see Figure 3.17). For example, Jock Mackinlay at Xerox PARC used circular Venn diagrams as his starting point and he ran into this exact problem of how to devise an arrangement for the cases that involve more than three intersecting sets (personal communication). There are also the interfaces by Michard (1982) and Hearst (1994) that use the Venn diagrams as their key visual metaphor, and they have not been able to move beyond three intersecting sets. We will now present designs that show how we could combine the rank layout of the InfoCrystal with Venn diagrams for the case of at most five intersecting sets (see Figures 3.16 to 3.18). However, we abandoned the circular Venn diagrams as an additional visual organizing principle because first of all it increased the visual clutter. Furthermore, the border segment of an interior icon that is closest to a circle does not have the same color as the circle (see Figures 3.16 to 3.18). Hence, the visual coding cues that users receives from a circle and the border segment of an interior icon in its vicinity are in conflict. This realization was the outcome of informal user studies, where we asked the subjects to color the borders of the interior based on the color of the reference icons and colored circles. The subjects initially used local visual cues to decide how to color a particular border area, but the circle segment closest to the interior icon would not represent the correct color. Hence, the subjects did have to adopt a more global perspective to determine the correct color, which they were able to do.

---



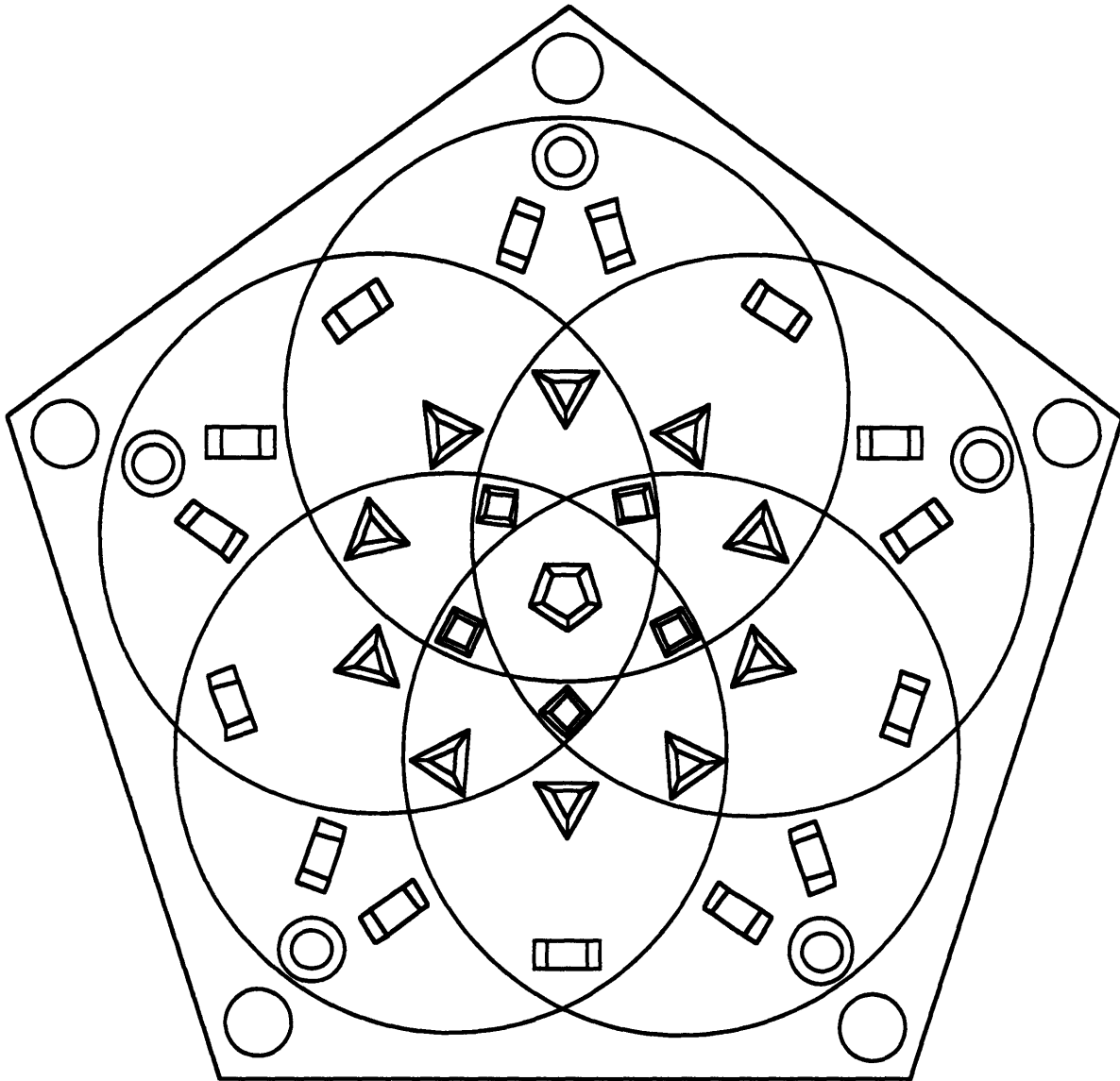
**Figure 3.16:** shows how the InfoCrystal with three inputs can be combined with the familiar circular Venn diagrams.



**Figure 3.17:** shows how the InfoCrystal with four inputs can be combined with the familiar circular Venn diagrams. The four intersecting circles do not create any areas that correspond exclusively to the relationships of rank two that involve non-adjacent criteria. Hence, we have to place these rectangular icons in the areas that "belong" to the icons of rank one.

---





**Figure 3.18:** shows how the InfoCrystal with five inputs can be combined with the familiar circular Venn diagrams. However, the intersecting circles do not contain areas that correspond exclusively to the relationships of rank two that involve non-adjacent reference concepts, and areas that represent relationships of rank three where only two of the reference concepts are adjacent to each other.

---

---



## CHAPTER 4

# VISUALIZING BOOLEAN QUERIES

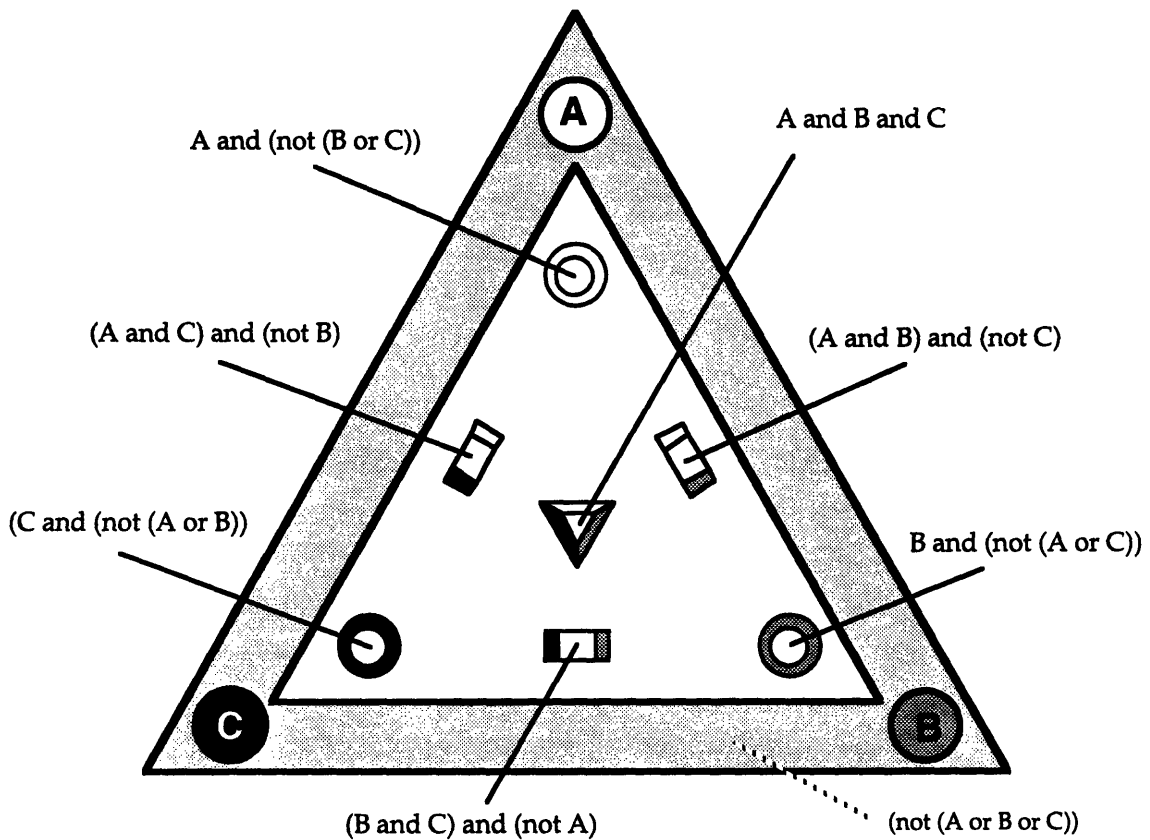
### 4.1 Introduction

How can we make Boolean retrieval more transparent and easy-to-use without limiting its expressive power? In this chapter we demonstrate how the InfoCrystal can be used to visualize and generate Boolean queries. We have discussed in chapter 2 how Boolean retrieval can be characterized along four dimensions. We will show that the way these four dimensions affect the broadness of a query can be visualized using the same simple visual analogy: the greater the area of the visual object representing the query, the broader the query. This will make it much easier for users to formulate and modify Boolean queries and to achieve the desired retrieval results.

Each interior icon of the InfoCrystal represents a distinct Boolean relationship among the input criteria (see Figure 4.1). Hence, users can specify Boolean queries by interacting with a direct manipulation interface. Appendix 1 provides an extensive tutorial that contains many examples of how a Boolean query can be specified using the InfoCrystal representation. The InfoCrystal acts as a *Boolean calculator*. Users do not have to use logical operators and parentheses explicitly to formulate queries. Hence, users do not have to concern themselves with the coordination problem. Instead they need to recognize the relationships of interest and select them. If an interior icon is selected, then it changes its visual appearance. In the figures of this thesis the center area of selected interior icons are displayed in black and the unselected ones in white.

In an InfoCrystal we partition the query space defined by its  $N$  inputs into  $2^N - 1$  disjoint subsets or *constituents* (the only disjoint constituent that is not explicitly represented in the form of an interior icon is the complement of all  $N$  inputs). It can be shown that any Boolean query that involves the inputs of an InfoCrystal and that applies the Boolean operations of union, intersection

---



**Figure 4.1:** shows the Boolean relationships associated with the interior icons for an InfoCrystal with three inputs or search criteria. The InfoCrystal represents all the possible queries involving its inputs in normal disjunctive form.

or negation can be represented by the union of a certain number of these constituents [Kuratowski and Mostowski 1976]. The InfoCrystal represents all the possible Boolean queries involving its inputs in *disjunctive normal form*. A query in disjunctive form is a disjunction of clauses, each of which is a conjunction of concepts, where some of them can be negated. Each interior icon represents a conjunction of the input concepts, where some of them are negated (see Figure 4.1). If a user selects several interior icons then the resulting Boolean query is equal to the disjunction of the Boolean queries associated with the selected icons.

We need to point out that the interior icons of an InfoCrystal can not represent any query that includes the constituent that represents the complement of all the inputs. We make the implicit assumption that users are interested in information where always at least one of their stated search

interests is satisfied. This is a reasonable assumption to make, because, first of all, the complement of all the inputs will cause a very large set of information to be retrieved, which is very expensive to compute and therefore many commercial retrieval systems will not permit users to formulate such queries. Second, if users are interested in the complement of all their stated interests, then here will come a point where they want to intersect this huge and unwieldy set with an additional, but positive search interest. Hence, they can create a hierarchical InfoCrystal query, as we will be discussed in section 4.2, where the search criteria, whose complement is of primary interest is organized in a separate InfoCrystal, whose output is connected to an InfoCrystal one level up in the hierarchy. In short, users can use the InfoCrystal to express the complement of several concepts by integrating it into a hierarchical InfoCrystal query structure. Furthermore, if the need arises that the complement is explicitly represented in a simple visual way, then the border area could represent the complement and its visual appearance could reflect its selection status (see Figure 4.1). However, the InfoCrystal in its current form does not explicitly represent the complement and thereby shields users from a large class of queries that are usually of little benefit to them.

Unlike traditional Boolean methods, the InfoCrystal provides a partial ordering of the retrieved documents based on the degree of coordination between the search criteria, which is visualized by the rank layout. In particular, the Null Output and Output Overload problems are addressed by ranking the retrieved documents based on the number of criteria that are satisfied. Any document that satisfies all criteria will be in the top rank of the output; any which satisfies all but one will be in the second rank, and so forth. This ensures that there will be no null output as well as that the output is presented to users in a structured and partially ranked fashion so to avoid the overload problem. Many retrieval specialists believe that ranked output provided by the InfoCrystal will produce better results than the traditional Boolean approach [Cooper 1988].

## 4.2 Query Space Visualized by the InfoCrystal

For an InfoCrystal with  $N$  inputs there are  $2^{2^N-1}$  queries that can be specified by just selecting the appropriate interior icons. There are  $2^N - 1$  interior icons.

---

Each can either be selected or so not, so each interior icon doubles the number of possible queries. Existing visual query languages allow users to formulate specific queries, but the InfoCrystal query language enables users to formulate a whole range of related queries by creating a single InfoCrystal [Anick et al. 1991, Young and Shneiderman 1993]. For example, if there are five inputs then there are over 2 billion possible queries and they are all represented compactly by an InfoCrystal !

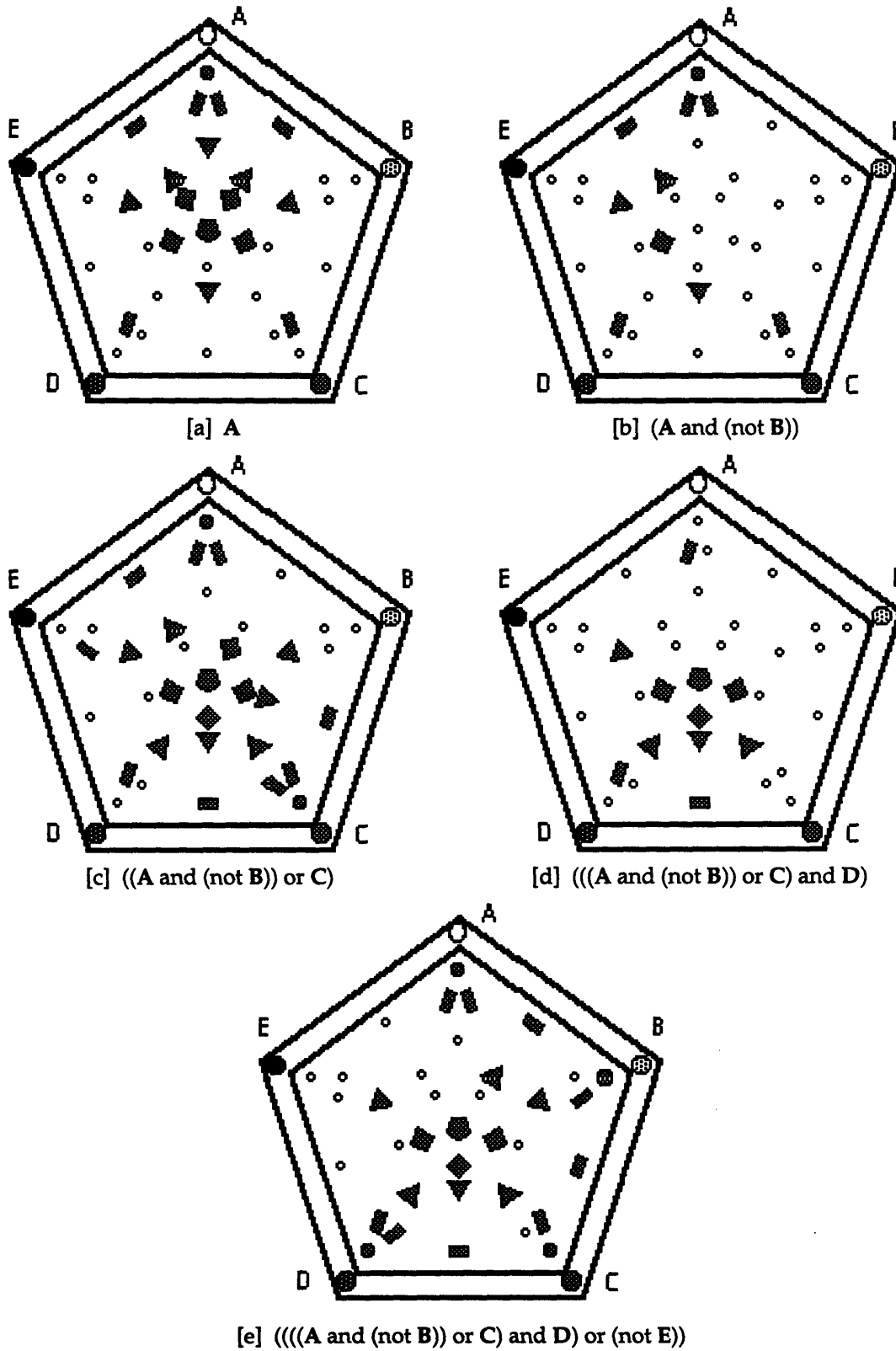
#### **4.2.1 Ways of Specifying a Boolean Query**

The great number of queries that can be expressed by a single InfoCrystal also illustrates the dilemma that such expressive power can pose for a user. How to specify the appropriate selection pattern from this large universe of possible queries ? We will provide next a summary of the multiple ways that users can use to specify queries: First, they can select specific relationships by clicking on the appropriate interior icons. Second, users can select subsets of interior icons by clicking on the criterion icons, thereby performing complex Boolean operations with only a few mouse clicks. Similar to a calculator, they can use the InfoCrystal as a Boolean Calculator, where the criterion icons represent the "numbers" or N different concepts to be operated on and the interior icons represent the accumulator (see 4.2.2 for discussion). Third, users can enter a specific Boolean query and there is a facility that automatically activates the appropriate interior icons. Fourth, users can select the interior icons by interacting with a threshold slider and/or the weighting sliders for the inputs (as will be shown in the next chapter).

#### **4.2.2 InfoCrystal as a Boolean Calculator**

Similar to a calculator, users can use the InfoCrystal as a Boolean Calculator, where the border icons represent the "numbers" or N different concepts to be operated on and the interior icons represent the accumulator, which can have  $2^{2^{N-1}}$  distinct states. How can we represent the Boolean functions AND, OR and NOT ? In our current implementation we represent these functions by holding down certain keys while clicking on a border icon: Command key = OR, Option key = AND, Command+Control = OR NOT, Option+Control = AND NOT.

---



**Figure 4.2:** [a], ..., [e] show the resulting selection pattern as users interact with the border icons in a step by step fashion to visualize the Boolean query  $(((A \text{ and } (\text{not } B)) \text{ or } C) \text{ and } D) \text{ or } (\text{not } E))$ . The text describes the particular key combinations that need to be pressed while the user clicks on the border icons to perform the appropriate Boolean operations.

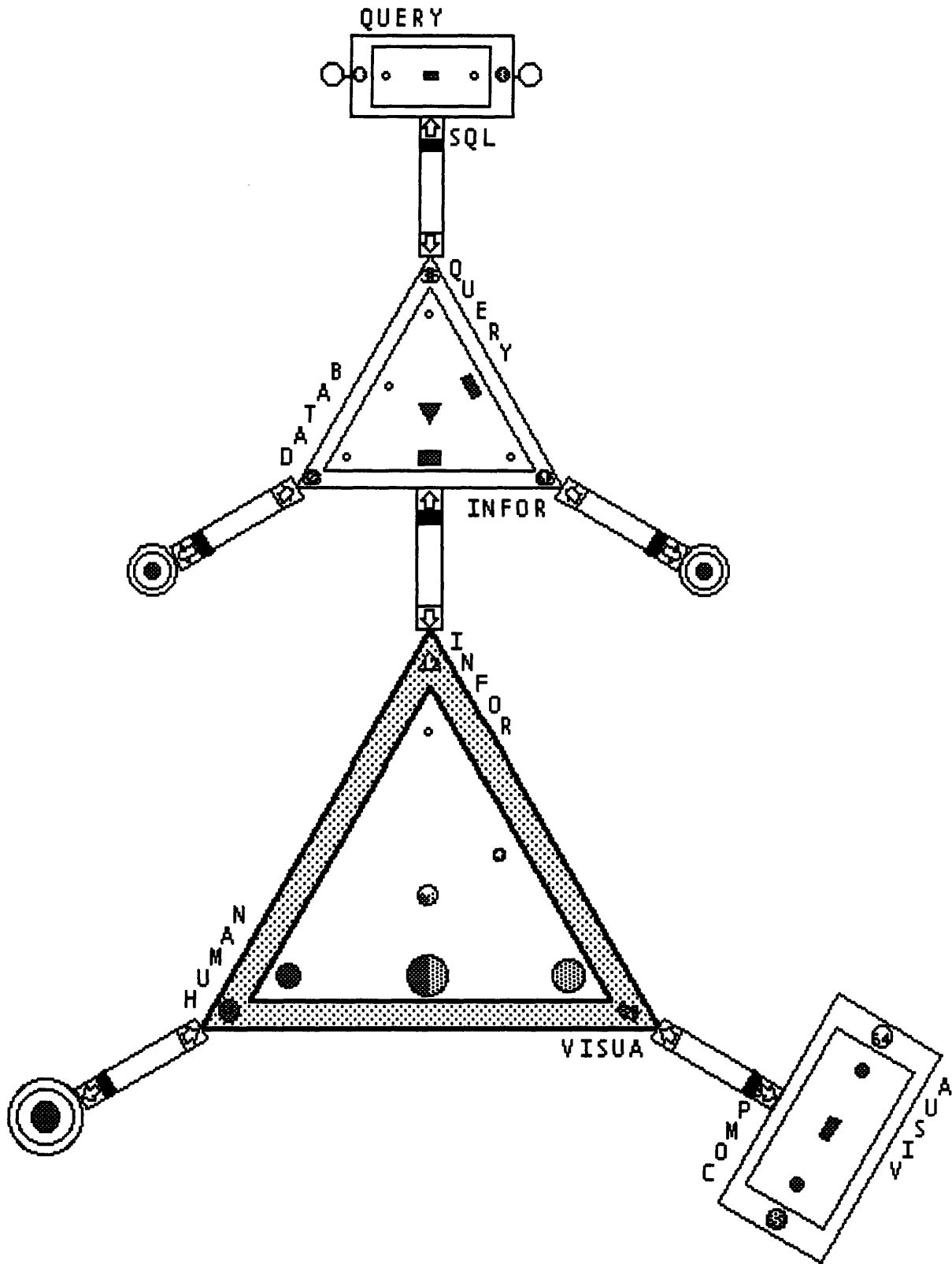
Figure 4.2 shows the resulting selection pattern as users interact with the border icons in a step by step fashion to visualize the Boolean query (((A and (not B)) or C) and D) or (not E)). First, users would click on the white border icon representing A while they hold down the Command key to specify the Boolean statement (A). Second, they would click on the border icon representing B while they held down the Option+Control key combination to specify the Boolean statement (... and (not B)), where the resulting selection pattern of the interior icons represents the Boolean expression (A and (not B)). Users can continue in a similar fashion to specify the remaining parts of the above Boolean statement. There is also the possibility to include a toolbar with buttons representing the different Boolean operators that users could select before clicking on a border icon to perform the same operations described above. To summarize, a Boolean query has a selection patterns of the interior icons associated with each of its components that need to be integrated and consolidated based on the operators joining them. The AND is equivalent to an intersection operation; the OR is equivalent to an union operation; and the NOT is equivalent to taking the inverse or complement.

### **4.3 Creating Complex and Nested Queries**

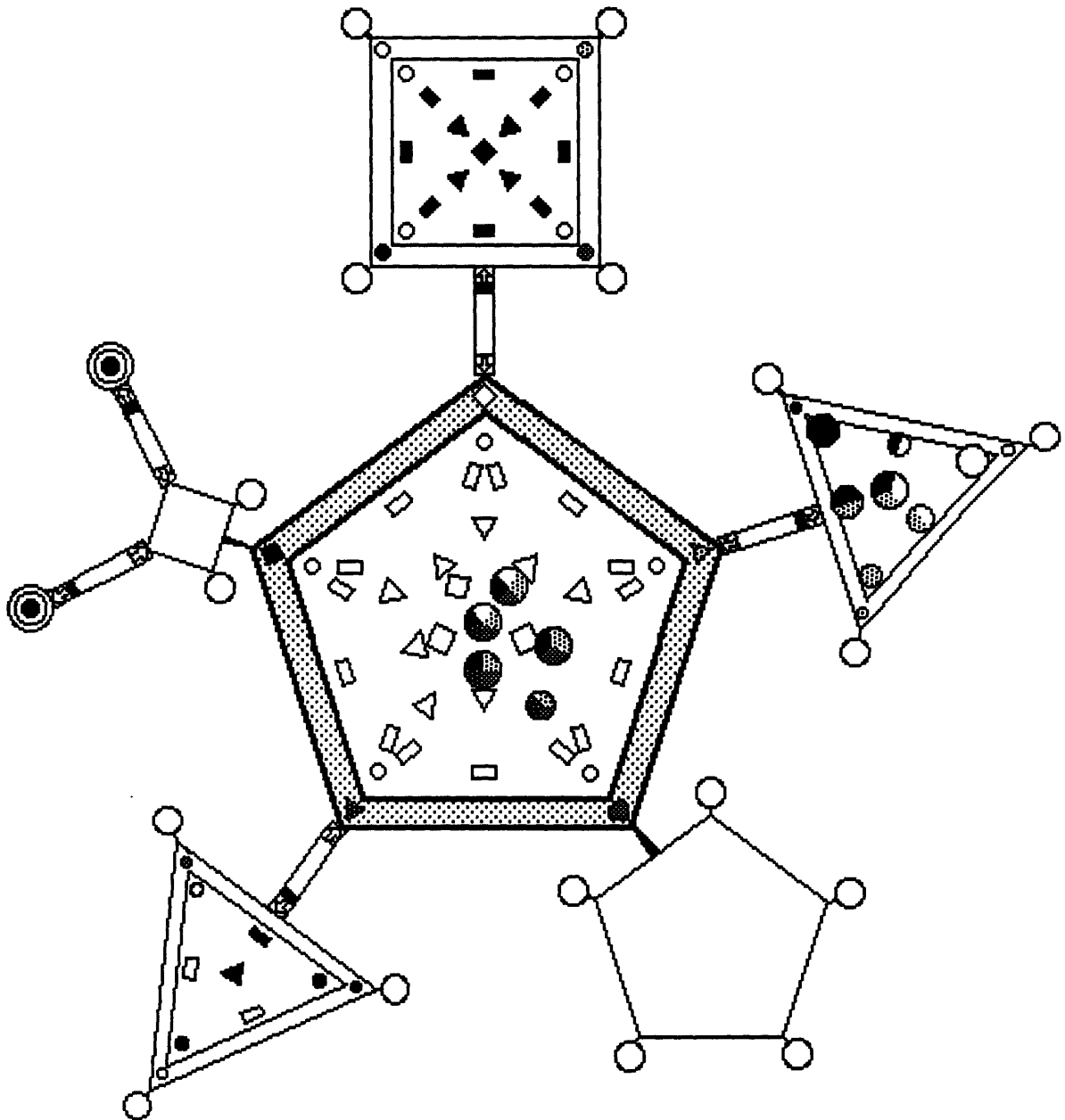
The InfoCrystals can be used as building blocks and organized in a hierarchical structure to create complex and nested Boolean queries. Figures 4.3 and 4.4 show how the InfoCrystals can be "chained together" to form a hierarchical query structure. Similar to a spreadsheet, users can ask "what-if" questions by changing which interior icons are selected in one InfoCrystal and observe how the contents of the dependent icons higher up in the hierarchy change dynamically. First, an InfoCrystal has several inputs, represented by the criterion icons, and has an output that is defined by the selected interior icons. Second, the output of an InfoCrystal will be one of the inputs to an InfoCrystal one level up in the query hierarchy. The leaf or atomic nodes of the InfoCrystal query structure represent the locations, where we interface with external information sources. The InfoCrystal is flexible in terms of the particular retrieval methods that are used to generate its input sets. Furthermore, it works for any data type. The items, which are retrieved based on the instructions specified in the leaf nodes, are then propagated through the query structure in a bottom-up fashion.

---





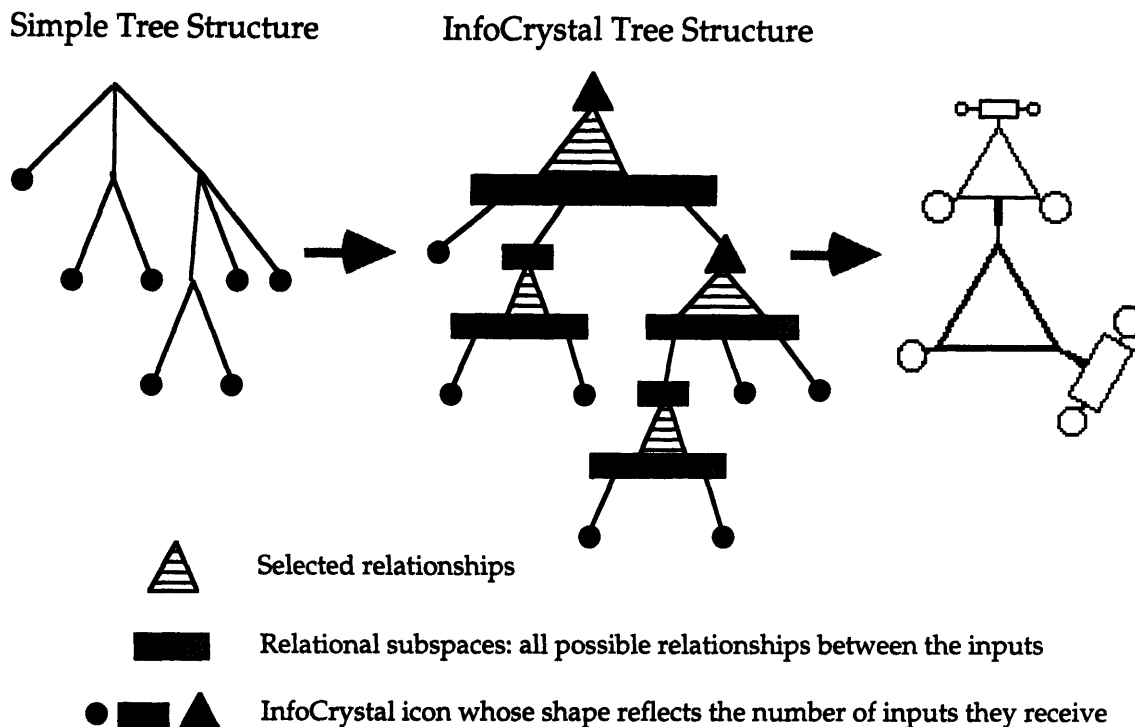
**Figure 4.3:** shows how the InfoCrystal tree structure shown in Figure 4.5 can be visualized by organizing the InfoCrystals in a hierarchical structure. The selected interior icons, displayed in solid black, define the output of an InfoCrystal. Users can see how the information distributes across the different relationships at the top level. In Figure 4.6 we show how this structure can also be represented by a text outline.



**Figure 4.4:** shows how another complex query can be represented by a series of InfoCrystals that are organized in a hierarchical structure. Some of the InfoCrystals are displayed only as an outline, but the user can just double-click them to view them in full detail. In the top-level InfoCrystal the selected icons are rendered, using a pie chart representation, to emphasize the quantitative information associated with them. Users can interactively change the way the InfoCrystals filter their inputs and they can dynamically observe how the information coming in through the circular InfoCrystals is propagated through the query structure.

---

---



**Figure 4.5:** shows how an InfoCrystal tree structure differs from a simple tree structure. The figure on the very right shows the InfoCrystal tree structure visualized as a hierarchical InfoCrystal, where the individual crystals are shown as outlines. Figure 4.3 shows the same structure visualized in more detail.

Figure 4.5 illustrates the difference between a simple tree structure and an InfoCrystal tree structure. In the latter the parent nodes do not *just* inherit the data elements associated with their children nodes. Instead there is an intermediary step where all the relationships between the children nodes are represented in an InfoCrystal. Users have to decide which relationships should be included in the InfoCrystal's output that is passed on to the parent.

Of all the many queries that can be specified with an InfoCrystal, not all of them can be reduced or simplified to an algebraic Boolean form that is easy to grasp by the users. Further, users are not equally likely to formulate all the possible queries that involve  $N$  concepts. They tend to create Boolean queries that are not too long, that do not possess too many brackets or nesting levels, and that use in any query component only one of the Boolean operators.

This raises the following issue: To what extent will users be able to exploit the expressive power of the InfoCrystal, given the types of Boolean queries they have tended to formulate so far? It can be argued that the above

observation is a result of the difficulty users have in formulating Boolean queries using the current text-based and algebraic interfaces. An advantage of the InfoCrystal is that makes it possible for users to formulate any Boolean query by just clicking on the interior icons, without having to worry about specifying the operators and using parentheses correctly. However, as far as simplicity is concerned, there exists an inverse relationship between the Boolean and InfoCrystal query language for certain queries. For example, the Boolean query that is equal to the OR of all concepts requires the selection of all the interior icons, whereas the Boolean query that corresponds to a circular icon can require the use of all three Boolean operators AND, OR, NOT and the use of brackets. On the one hand, there are queries that are conceptually straightforward to express using the Boolean query language, but require the selection of a great number or a complex pattern of interior icons, which can be a demanding and time-consuming task. On the other hand, there are queries of the form, for example, "not more, exactly or at least M out of N concepts should be satisfied" that are easy to express in an InfoCrystal, but they are difficult to formulate as a Boolean query even for very experienced users. In chapter 8 we present the results of a user study that reflect this inverse relationship between the Boolean and InfoCrystal query languages.

By way of an analogy, we want to address the issue that certain simple or common Boolean queries require users to select a large subset of the interior icons. The larger the number of the interior icons to be selected, the greater the probability that users will fail to select all the necessary icons. Computer drawing programs provide users with a set of standard shapes such as an ellipse or a rectangle that they can use to get started and create more complex drawings. Similarly, we can provide users with a pull-down menu of standard Boolean queries that they can apply to an InfoCrystal. They can then modify the resulting selection pattern to arrive at a query that corresponds more closely to what they are looking for, but that would have been too difficult for them to formulate initially.

We also need to address the issue of how nested Boolean queries can be expressed in the InfoCrystal representation. We can use substitution to rewrite nested queries so that we have at each level and for each query component an equivalent and simple query that uses a single operator to join

---

its components; i.e., (Green OR (Red AND Blue) OR (Red OR (Green AND Blue))) => (Green OR A OR B), where A=(Red AND Blue), B=(Red OR B'), where B'=(Green AND Blue). We have a choice how we want to represent a nested Boolean query in the form of an InfoCrystal. On the one hand, we could easily create a separate InfoCrystal for each of its simple components, and we could organize these crystals in a hierarchical structure. This has the advantage that the InfoCrystals are easy to program because their associated Boolean queries are simple and we can use one of the predefined Boolean queries from a pull-down menu. However, it has the disadvantage that we lose the flexibility we gain if we represent the overall query with a single crystal. On the other hand, we can represent the query in a single InfoCrystal. However, it may not be immediately apparent which of the interior icons need to be selected or which of the predefined Boolean queries should be selected from the pull-down menu to get us started, unless we could break the whole translation process into steps. There is a simple answer to this translation problem if we know how to formulate the Boolean query or it has somehow been given to us. For any Boolean query the InfoCrystal software provides the facility to automatically visualize it in a single InfoCrystal or in a series of simple InfoCrystals, depending on the user's preference. If, however, we want to determine ourselves which of the interior icons need to be selected, then we need help in addition to the predefined pull-down menu of simple and common Boolean queries.

At this point we have described a partial solution to the requirement that we want to be able users to translate by themselves any nested Boolean queries into a single InfoCrystal in a series of steps, where each step only involves simple selection operations. The general solution hinges on our ability to propose a visually elegant way that enables users to *save partial results and to access them at a later stage to combine them*. We are faced with the same problem that we encounter when using a calculator to compute a complex numerical formula. If the formula can be rewritten in such a way that we can operate on the partial and intermediary results to arrive at the correct solution, then we do not need the facility to be able to save intermediary results. In such instances, we have described how the InfoCrystal can be used as a Boolean Calculator.

---

## 4.4 The Outliner Tool

Creating a simple or even a complex InfoCrystal query structure is as simple as generating an outline for a paper using the developed *query outliner* tool. Users can use it to incrementally structure complex searches by creating a hierarchical query outline (see Figure 4.6). The query outliner has a similar functionality to the familiar outlining tools available in word-processing packages. The outliner solves the problem of how users can easily annotate and summarize in a word or two what the nodes in a query hierarchy represent. Information retrieval scientists often suggest to searchers to generate a structured list of their search interest, where quasi-synonymous words for each conceptual factor are ORed and the different synonym lists are then ANDed [Cooper 1988, Marcus 1991]. The query outliner enables users to easily generate such structured lists.

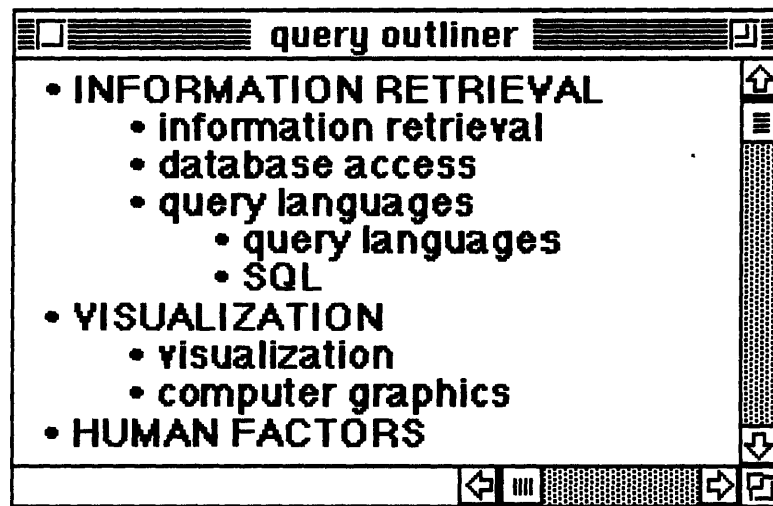


Figure 4.6: shows the outliner tool that users can use to generate an InfoCrystal query structure.

Users start to create a new outline item by pressing the return key, and they change the nesting level by pressing the tab or delete key. Users can use copy, cut and paste to modify the query outline. They can also select a part of the outline and move it to a different location in the query outline, where the InfoCrystal query structure is automatically updated to reflect this change. Hence, it is very easy for users to modify the query structure by interacting with the query outliner.

## 4.5 Narrowing and Broadening Techniques

In this section we will show how ways to broaden or narrow a Boolean could be visualized. For that purpose we repeat material already presented in section 2.3.1.2 to present the needed facts and steps in one place. A Boolean query can be described in terms of the following four operations: degree and type of coordination, proximity constraints, field specifications and degree of stemming as expressed in terms of word/string specifications. If users want to (re)formulate a Boolean query to retrieve documents then they need to make informed choices along these four dimensions to create a query that is sufficiently broad or narrow depending on their information needs. Most narrowing techniques lower recall as well as raise precision, and most broadening techniques lower precision as well as raise recall. Any query can be reformulated to achieve the desired precision or recall characteristics, but generally it is difficult to achieve both. Each of the four kinds of operations in the query formulation has particular operators, where for each operator with a narrowing effect, there is one or more inverse operators with a broadening effect [Marcus 1991].

If users want to (re)formulate a Boolean query, then they need to make informed choices along these four dimensions to create a query that reflects their information needs. This can be a daunting task, because the number of feasible choices grows exponentially as the number of concepts increases. Hence, users require help to gain an understanding of how changes along these four dimensions will affect the broadness or narrowness of a query.

Figure 4.7 shows how the four dimensions affect the broadness or narrowness of a query: 1) *Coordination*: the different Boolean operators AND, OR and NOT have the following effects when used to add a further concept to a query: a) the AND operator narrows a query; b) the OR broadens it; c) the effect of the NOT depends on whether it is combined with an AND or OR operator. Typically, in searching textual databases, the NOT is connected to the AND, in which case it has a narrowing effect like the AND operator. 2) *Proximity*: The closer together two terms have to appear in a document, the more narrow and precise the query. The most stringent proximity constraint requires the two terms to be adjacent. 3) *Field level*: current document records have fields associated with them, such as the "Title", "Index", "Abstract" or

---

"Full-text" field: a) the more fields that are searched, the broader the query; b) the individual fields have varying degrees of precision associated with them, where the "title" field is the most specific and the "full-text" field is the most general. 4) *Stemming*: The shorter the prefix that is used in truncation-based searching, the broader the query. By reducing a term to its morphological stem and using it as a prefix, users can retrieve many terms that are conceptually related to the original term [Marcus 1991].

Figure 4.7 also indicates how the broadness or narrowness of a query could be visualized. Each of the four dimensions that characterize a Boolean query can be represented by a visual object: The larger the visual extent of an object, the broader the effect of the corresponding aspect of the query. Figure 4.8b shows how the *stemming* dimension can be represented by a gray rectangle, whose width reflects the degree of stemming applied. The resulting stem is used for a truncated search. Currently, we do not provide for a visual signature to represent an exact search, but this could be signaled by having a transparent stemming curtain. The different fields are also represented by rectangles and the more that are selected, the larger the overall gray area. Figure 4.8a shows how a *proximity* constraint can be represented by grouping

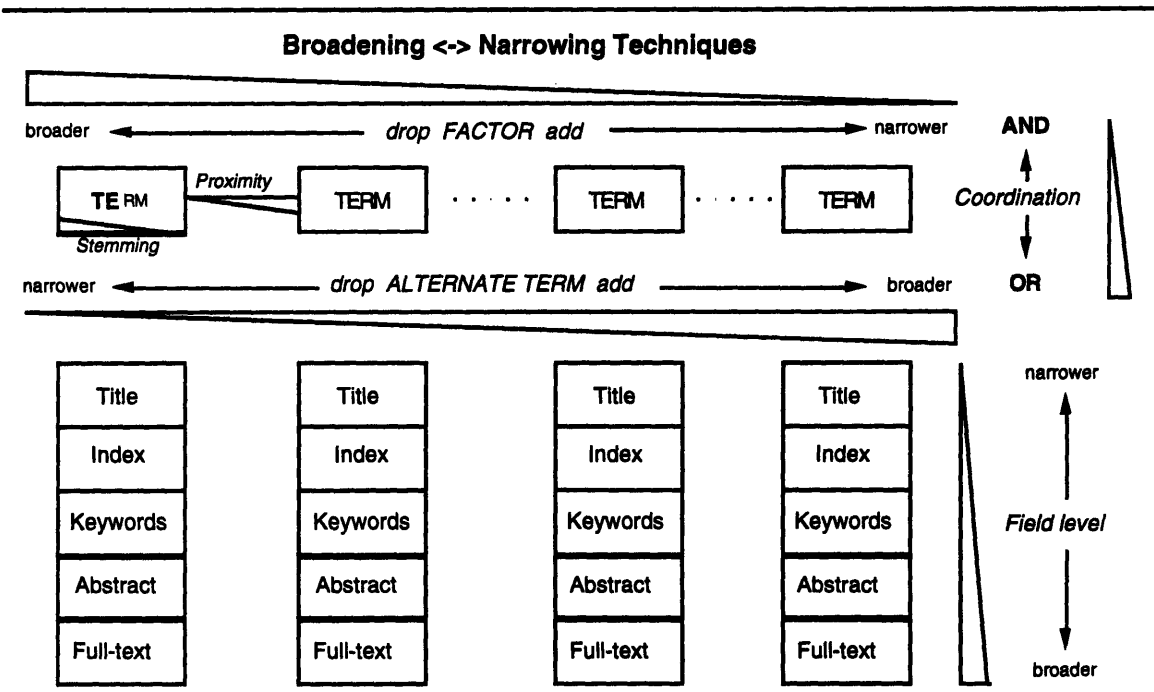
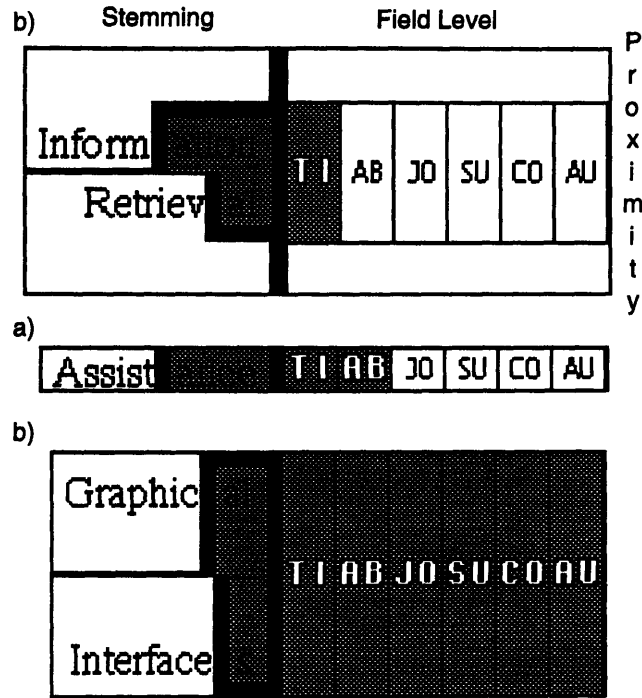


Figure 4.7: captures how coordination, proximity, field level and stemming affect the broadness or narrowness of a Boolean query. By moving in the direction in which the wedges are expanding the query is broadened.

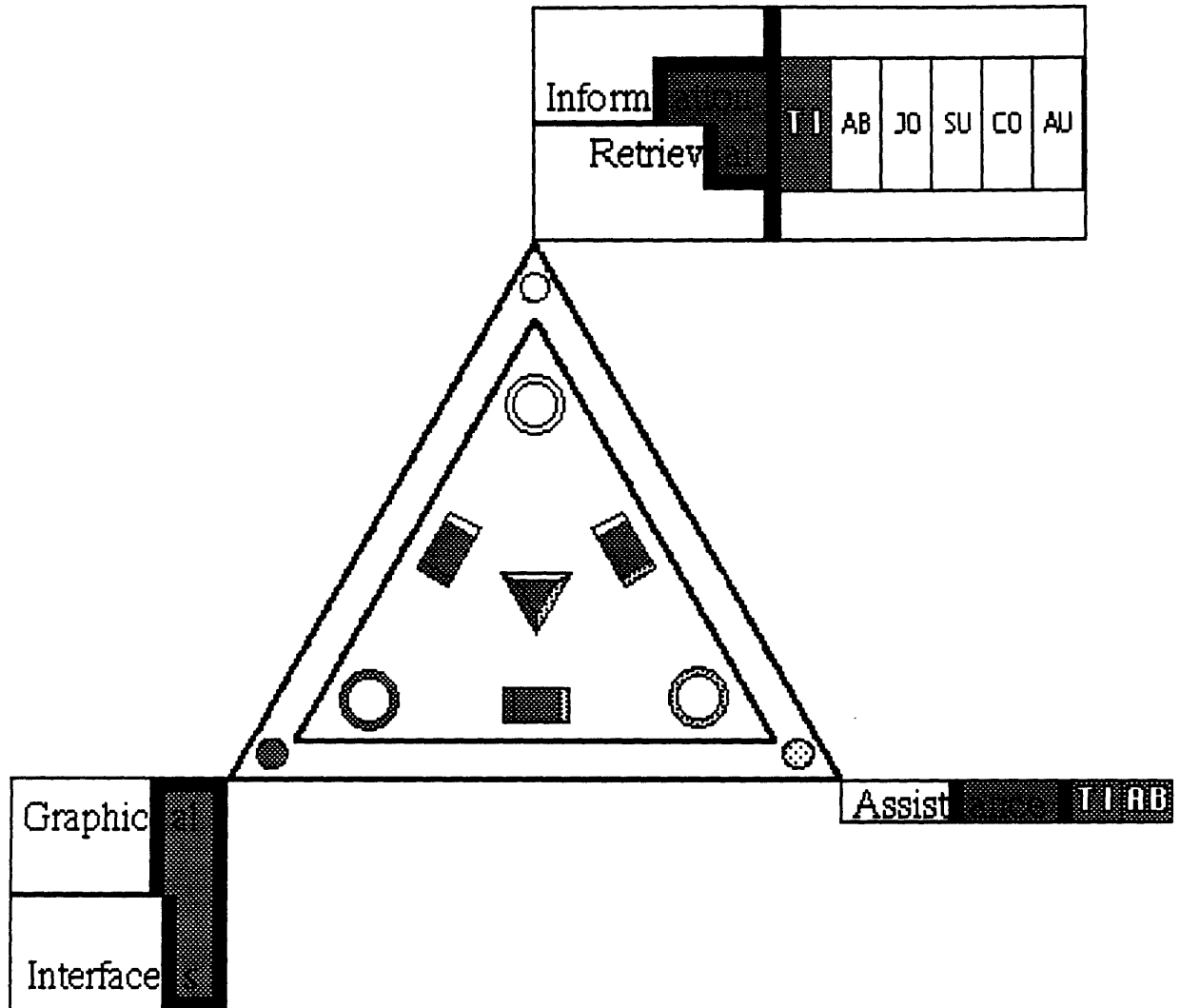




**Figure 4.8:** shows how the proximity, field level and stemming constraints for the information need "information retrieval assistance using graphical interfaces" can be visualized. a) Shows an example of how the degree of stemming and the field level are represented by their respective rectangles. The resulting stem corresponds to the part of the word not covered by the gray stemming rectangle. b) Shows different examples of the proximity constraint: the smaller the height of the rectangles, the closer together the terms have to be in a document.

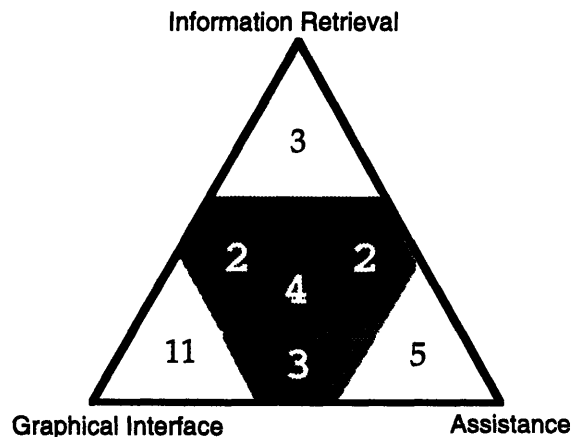
the involved terms and stacking their visual analogs. The degree of proximity is represented by the height of the rectangles. Finally, we can modify the way the InfoCrystal visualizes Boolean coordination so that the resulting broadness or narrowness is reflected by an area-based visual measure. Instead of using icons, we could associate a visual area with each of the disjoint subsets (see Figure 4.10).

The guiding visual metaphor is simple and intuitive: The broadness of a query is represented by the size of its visual analogs (the larger the visual area, the broader the query). Hence, the query characteristics in terms of coordination, proximity, field level and stemming and how they affect the broadness of the query can be made visually explicit and transparent. The visual representation can be thought of in several analogous ways:



**Figure 4.9:** shows how the visual objects representing the stemming, field and proximity constraints can be integrated with the part of the InfoCrystal that visualizes Boolean coordination to represent the information need "information retrieval assistance using graphical interfaces". This interface gives users a quick insight into the constructed query. Users can change the amount of information retrieved by changing the size of the gray area of the objects that represent the different aspects of a Boolean query. The larger the shaded areas, the more information is retrieved.

1) The *Sieve* analogy: each of the four dimensions can be thought of as controlling a gate through which information can flow. 2) The *Window* analogy: each of the four dimensions can be thought of as a window through which information becomes visible. 3) The *Optical Lens* analogy: each of the four dimensions can be thought of as a lens through which information can pass and users can manipulate the degree to which they want to focus the stream of information.



**Figure 4.10:** shows how we could visualize Boolean coordination using an area-based measure. The numbers represent the number of documents that are associated with the different relationships visualized by the InfoCrystal.

Once users have understood one of these simple analogies, it will be easy for them to modify or fine-tune their query depending on their information needs. They simply have to think of it in terms of manipulating the coarseness / fineness of the visual representation: For example, by enlarging the gates through which information can flow, they are increasing the information that becomes available. If users are dealing with a nested query, then they have a hierarchy of gates through which they can control the flow of information. At the lowest level users could use the "stemming", "field level" and "proximity" gates to control the flow of information, whereas higher-up they will use the "coordination" gates to broaden or focus the information.

To summarize, we have demonstrated how the InfoCrystal can be used to formulate and visualize arbitrarily complex Boolean queries. In addition, we have presented special visual tools for modern Boolean text retrieval.



## CHAPTER 5

# VISUALIZING WEIGHTED QUERIES

### 5.1 Introduction

One of the complaints about Boolean structured retrieval is that it does not ordinarily provide users with a ranked output of the retrieved documents. We have shown in a previous chapter that the InfoCrystal provides a partial ordering of the retrieved documents based on the degree of coordination between the search criteria. We want to show in this chapter how the InfoCrystal representation can be extended to formulate and visualize weighted queries.

It is desirable to be able to formulate weighted queries for several reasons. First, there are situations where the search criteria are not of equal importance to a user, and we want to use this information to rank the retrieved documents accordingly. Second, users can find it initially easier to formulate queries by creating just a list of their interest and assigning relevance weights to them [Frei and Qiu 1993]. These weights can be used by a retrieval system to generate a Boolean query. However, the question arises of how to exactly use these weights to create a Boolean query. We will show how the InfoCrystal can be used to translate weighted queries into Boolean queries. The InfoCrystal has the added advantage that it enables users to see and control in a visual way how the translation is performed. Further, we will show how the interior icons can be displayed in a ranked way that reflects the weights of the search criteria by developing the *bull's-eye layout*. Finally, we discuss and show why weighted queries do not have an expressive power equivalent to the Boolean query language does. We also address why the interior points of an InfoCrystal can not be used to specify every possible direction of a vector of weights.

---

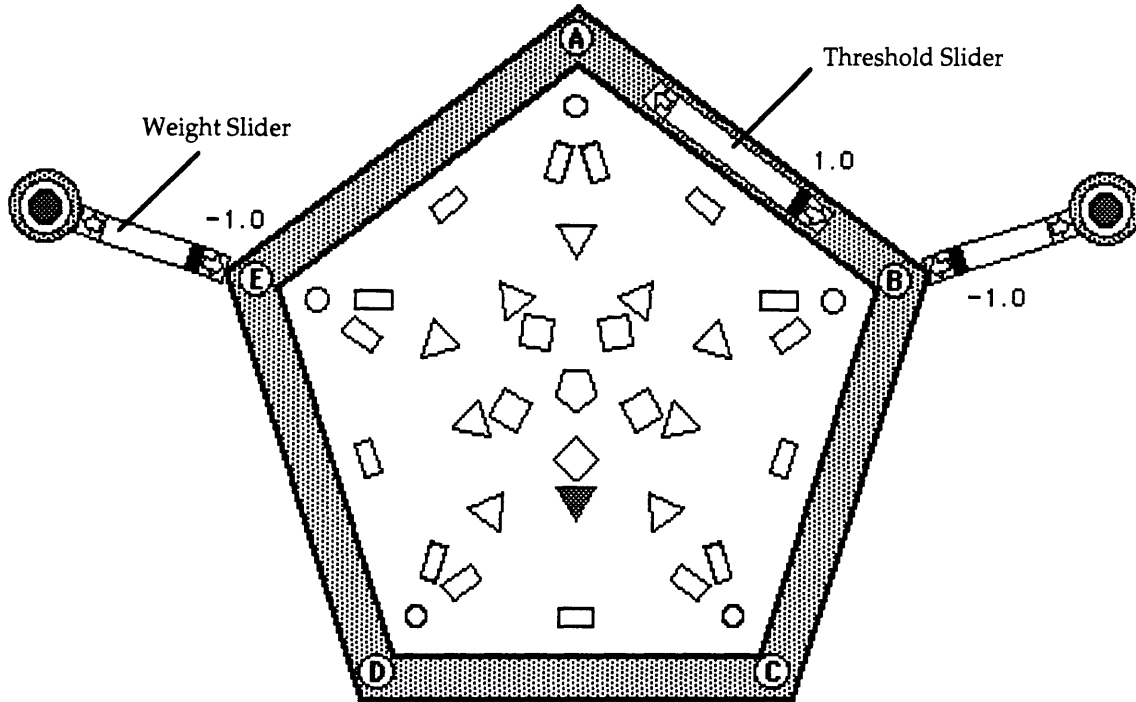
## 5.2 Formulating Weighted Queries using the InfoCrystal

There are two ways of extending the InfoCrystal representation to be able to formulate weighted queries. First, users can assign weights to the concepts listed in the query outliner window by entering weights, enclosed by square brackets, after each outline item. Second, we can associate a slider with each input criterion of an InfoCrystal, and users can interact with the weight sliders to specify the degree of importance they assign to the inputs (see Figure 5.1, which shows only the sliders for the criteria whose weights are not equal to the default weight of plus one). Users can choose values between -1 and 1, where negative weights indicate that users are more interested in documents that do *not* contain the concept represented by the input (i.e., the weight -1 is equivalent to the logical NOT). There is also the possibility that the initial values of the weights are computed automatically based on the statistical term frequencies of the search terms in a document collection.

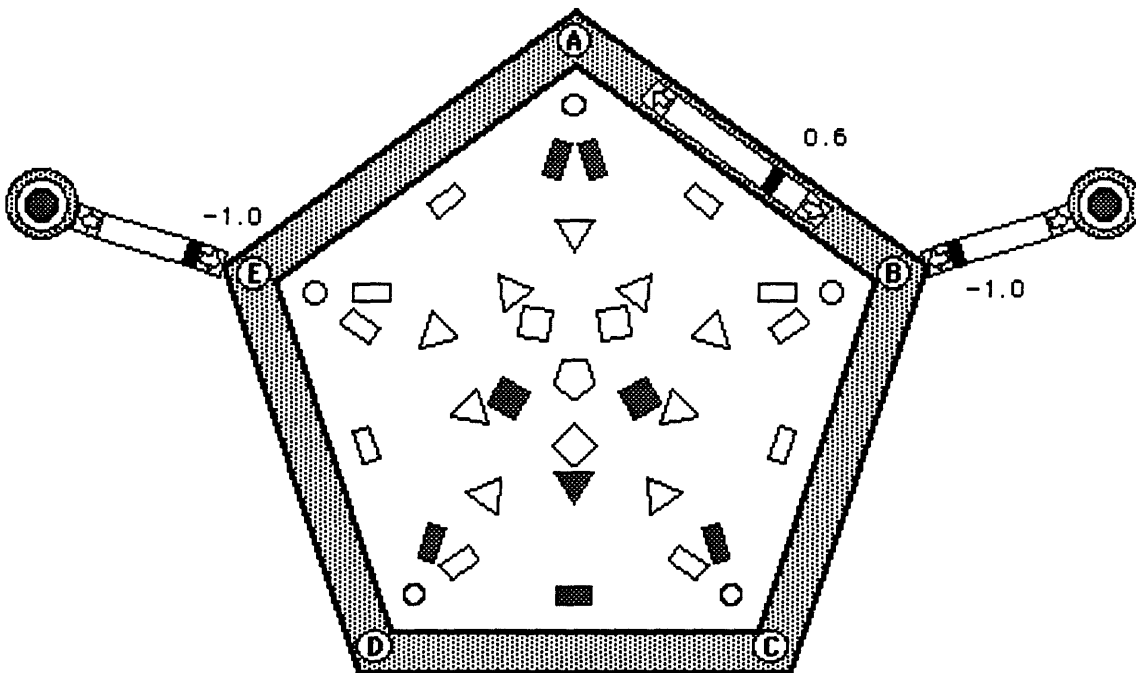
The assigned weights can be used to compute a relevance score for each interior icon. This score is equal to the normalized dot product between the vector of the input weights and a vector, whose values are equal to 1 or -1 depending on whether the corresponding search criteria are satisfied or not by the interior icon. By interacting with a threshold slider, users can select only those interior icons whose relevance score is above the threshold (see Figures 5.1 and 5.2).

The example shown in Figure 5.1 will help to explain how the relevance score is computed: First, we have assigned the weights 1, -1, 1, 1, and -1 to the search criteria A, B, C, D, and E respectively (only the sliders for the criteria whose weights are not equal to the default weight of plus one are shown). Hence, the vector of the input weights is equal to [1,-1,1,1,-1]. Second, we can define a vector for each of the interior icons, whose values are equal to 1 or -1 depending on whether the corresponding search criteria are satisfied or not by the icon. For example, the vector for the pentagon icon in the center of the InfoCrystal is equal to [1,1,1,1,1], because it satisfies all of the five search criteria. Hence, if we take the vector product of the weight vector and the vector associated with the pentagon, then we get a relevance score of  $\{(1 \cdot 1) + (-1 \cdot 1) + (1 \cdot 1) + (1 \cdot 1) + (-1 \cdot 1)\} = 1$ . If we normalize this score, then we get a value of 1/5, because the maximal score is equal to 5.

---



**Figure 5.1:** shows how the InfoCrystal can be used to formulate weighted queries by associating sliders with each of the input criteria (only the sliders whose weights are not equal to the default weight of plus one are shown). Users can set a threshold to approximate the weighted query by a Boolean query, where the weights are used to compute a normalized relevance score for each of the interior icons (see text). If the threshold is set equal to 1.0 and relevance weights for the search criteria A, B, C, D, and E are equal to 1.0, -1.0, 1.0, 1.0, and -1.0, respectively, then only the triangular icon, representing (A and (not B) and C and D and (not E)), is selected and therefore displayed in black.



**Figure 5.2:** shows which of the interior icons are selected when the threshold is lowered to 0.6 and the relevance weights remain the same as in Figure 5.1

Actually, this normalized score is equal to the cosine of the angle between the two vectors. To consider a further example, the shaded interior icon in the shape of a triangle has the vector  $[1,-1,1,1,-1]$ , because it satisfies A, C, and D, but not B and not E. Hence, its relevance score is equal to  $\{(1 \cdot 1) + (-1 \cdot -1) + (1 \cdot 1) + (1 \cdot 1) + (-1 \cdot -1)\} = 5$ , and its normalized score is equal to 1. In a similar fashion, we can compute the relevance scores for the other interior icons, and we can determine their selection status based on which of the normalized scores are above the current threshold. Figure 5.1 shows that only the shaded triangular icon is selected if the threshold is equal to its maximal value of 1.0. If we lower the threshold to 0.6, then Figure 5.2 shows the interior icons that have a relevance score equal or above this threshold.

The InfoCrystal makes it possible for users to specify a set of weighted search criteria and to approximate this weighted query as a Boolean query by setting a threshold that controls the exactness of the approximation. If they choose the maximal threshold of 1.0, then they require the approximation to be exact. However, there does not always exist a Boolean approximation for every weighted query if users require the threshold to be maximal. By lowering the threshold, they indicate how closely they want to approximate the weighted query by a Boolean query. If users choose the minimum threshold value of minus one, then they approximate the weighted query by a Boolean query that coordinates the search criteria with the OR operator. This is the broadest possible query and it will have a high recall and very low precision value.

We can think of the vector of weights as specifying a direction in a multi-dimensional information space in whose vicinity we want to explore the space. As we will discuss in the next chapter, this weight vector is equivalent to the query vector used in vector-space retrieval approaches to specify the user's information need. The vector-space approach represents the information items and the query as vectors, and it commonly uses the angle between the information item vectors and the query vector to determine their similarity. The threshold slider, which is used to control the degree of Boolean approximation, uses the same angular measure. As we will show, the InfoCrystal component that permits the formulation of weighted query just represents a special case of the more general vector-space retrieval

---



approach. In both cases we need to decide how similar we require the retrieved information to be to the specified weight or query vector. The difference between weighted query and vector-space approaches is that they operate on different basic groupings of information. On the one hand, the weighted query approach operates on the different possible relationships between the search criteria, where each relationship represents a collection of information items. On the other hand, the vector-space approach operates on the individual information items. Hence, the weighted query approach operates on *sets* of information items, where the membership is determined based on Boolean logic.

It is also relevant to briefly mention that the human visual system processes information at different levels of resolution and employs different grouping principles to arrive at higher-level perceptions [Marr 1982]. Similarly, we can think of the Boolean partitioning of an information space as a way to create higher-level constructs. This type of grouping will not always be the most appropriate one to perform or it will need to be replaced by other ways of organizing the information. Hence, the different retrieval approaches provide us with different ways of organizing information. Each of them has different expressive powers and is better suited for different tasks. As we hope to demonstrate in this thesis, the InfoCrystal offers a visual framework where users can explore and with little effort shift between different ways of retrieving and organizing large information spaces.

To summarize, one of the advantages of the InfoCrystal is that enables users to see and control in a visual way how a weighted query is translated into and approximated by a Boolean query. Further, users can easily employ a hybrid approach, where they use the weight and threshold sliders to get started and they can then proceed to select or deselect individual interior icons to arrive at a query that captures their information need more precisely. If the selection pattern of the interior icons is not consistent with the relevance weights and the threshold, then the threshold indicator is dimmed and not displayed in solid black.

---

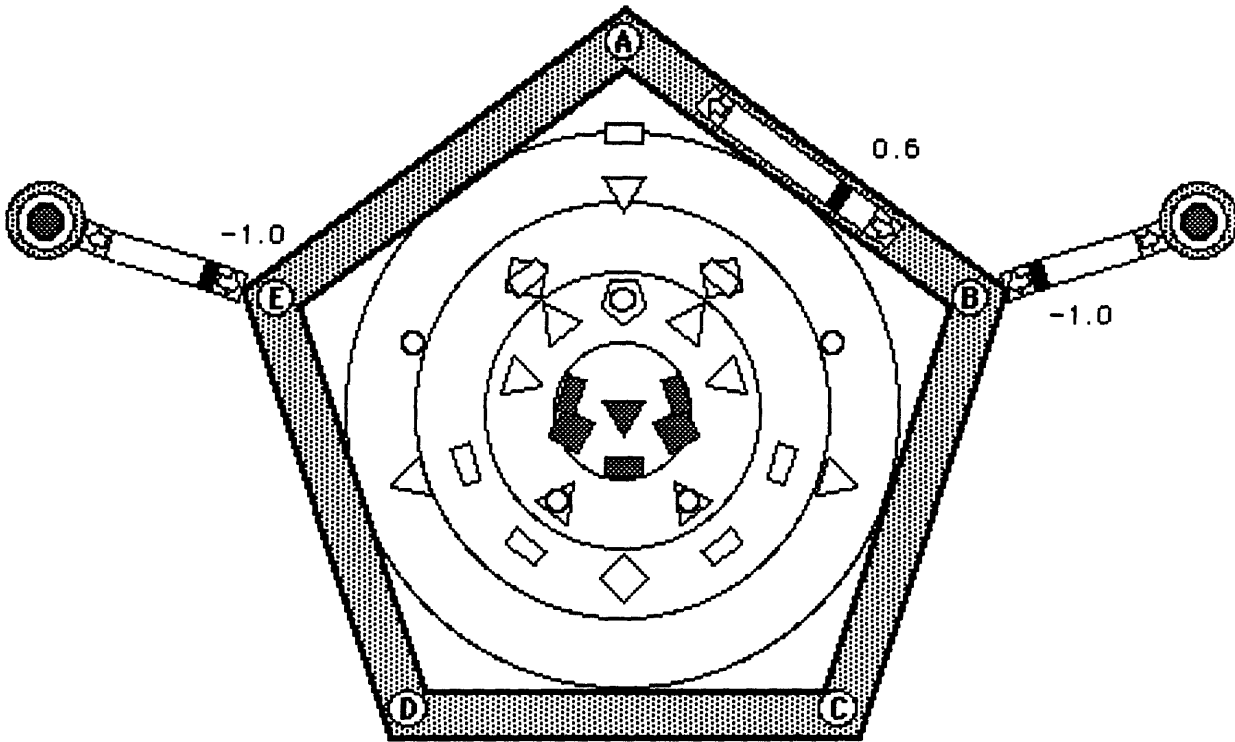
### 5.3 The Bull's-Eye Layout

By assigning weights to the query concepts, users can rank the interior icons and thereby impose a partial ordering on the retrieved documents, where documents associated with the same interior icon receive the same score. The key design principle used in the layout of the interior icons of an InfoCrystal is to ensure that users will find towards its center the relationships that they consider as more important. So far we have considered the *rank layout* that ensures that the number of criteria satisfied by an icon increases as we move towards the center of an InfoCrystal. We will now describe how users can display the interior icons to reflect their ranking based on the current setting of the relevance weights. This mapping, called the *bull's-eye layout*, causes the relationships with a higher relevance score to be placed closer towards the center of the InfoCrystal (see Figure 5.3).

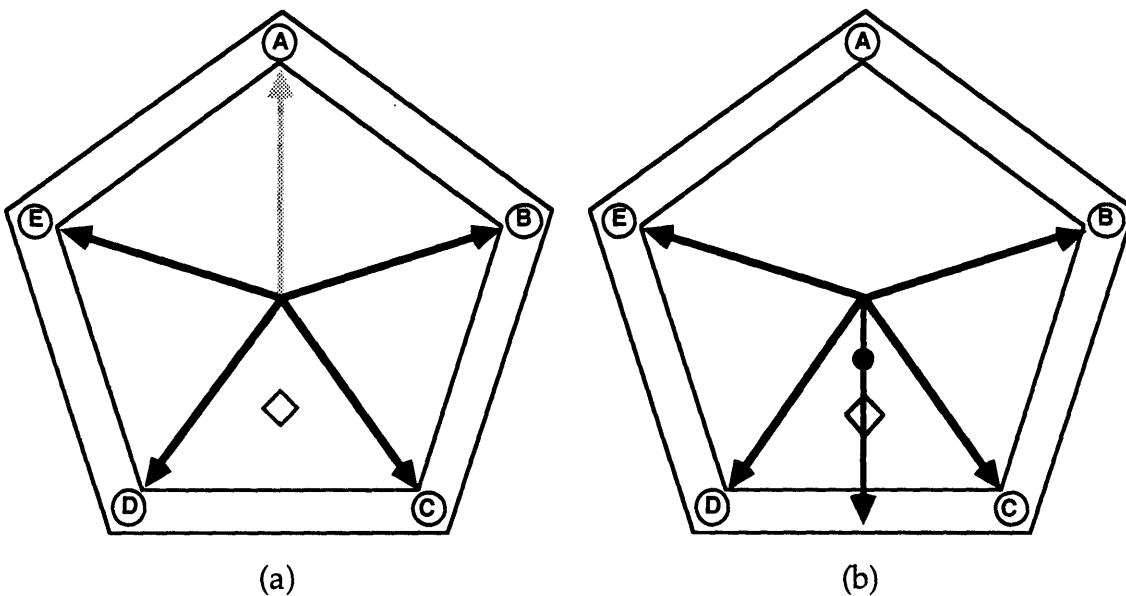
We use the following polar representation to determine the placement of the interior icons in the bull's-eye layout: 1) The *radius* value is determined by the inverse of the relevance score, which is equal to the cosine of the angle between the weight vector and the vector that we can define for each interior icon. 2) The *angle*, however, is not affected by the weights. It is a function of the line that passes through the InfoCrystal's center and the *center of mass* of the criterion icons that is computed as follows: a two-dimensional vector pointing from the center towards a criterion icon that is satisfied by an interior icon receives a positive mass of one, whereas the vector pointing towards a criterion that is not satisfied receives a negative mass of minus one. We take these vectors and their associated point masses to compute their center of mass (see Figure 5.4). Thus, this center of mass will be closer to those criterion icons that the interior icon satisfies than to those it does not. Finally, we place the interior icon where the line defined by the center of mass intersects the circle defined by the relevance score (see Figure 5.5).

There are degenerate cases, where the center of mass of an interior icon coincides with the center of the InfoCrystal, and therefore we can not specify the angle and its corresponding straight line. In these cases we place the interior icon where the line, which passes through the first criterion icon satisfied by the icon, intersects the circle (we have an implicit ordering of the input or criterion icons, which in the figures is made explicit by using the letters A, B, C, ... to label them).

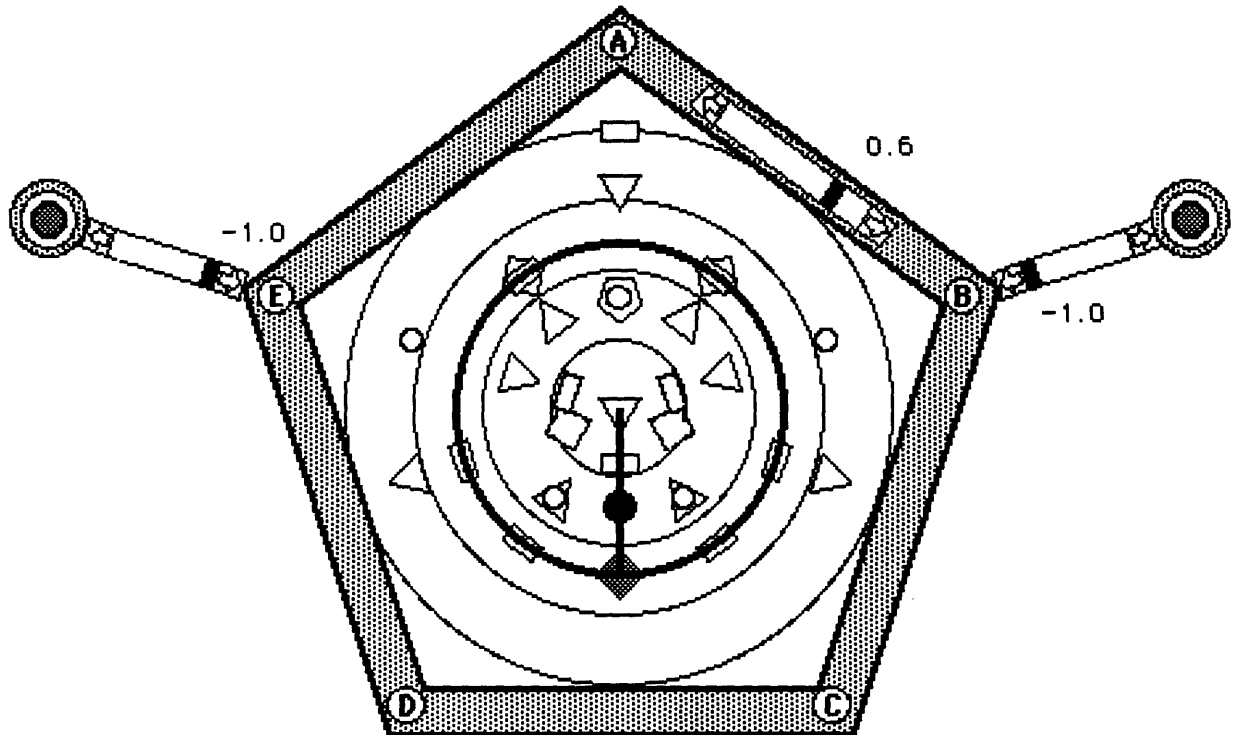
---



**Figure 5.3:** shows the *bull's-eye layout* of the interior icons, where they are placed in such a way that their relevance score increases as we move towards the center. This mapping visualizes the resulting ranking of the interior icons based on the current setting of the weights assigned to the search criteria, which in this case is 1, -1, 1, 1, and -1.



**Figure 5.4:** shows how to compute the *center of mass* for the interior icon of rank four that satisfies the criteria B, C, D, and E, but not A. The two-dimensional vectors pointing from the InfoCrystal's center towards the criterion icons B, C, D and E have a mass of plus one assigned to them and are therefore displayed in solid black. However, the vector pointing towards the criterion icon A receives a mass of minus one and is displayed in gray. If the vectors are scaled according to the masses associated with them then (b) shows the resulting vectors. The solid circle shows the location of the center of mass if the weighted average of the vectors is taken.



**Figure 5.5:** illustrates how the position of the icon of rank four, shown as a shaded square, is computed in the bull's-eye layout by using the following polar representation: 1) The radius value is determined by the inverse of the relevance score of the icon and it defines a circle on which it has to lie. 2) The angle is specified as follows: the center of mass of the shaded icon of rank four is shown as a solid circle and we define a line passing through it and the center of the InfoCrystal. We place the shaded square icon at the location where this straight line intersects the circle defined by the relevance score.

As Figure 5.3 shows, different interior icons can coincide and be mapped to the same location in the bull's-eye layout. Hence, a particular location in the bull's-eye does not have a unique interpretation in terms of how it is exactly related to the different search criteria. If we display the shape and color information associated with an icon, then we can tell more readily what the relationship is, but location information alone is not sufficient. This should come as no surprise, because information will be lost when we map a point in a  $n$ -dimensional space into a point in a two-dimensional display.

What is the relationship between the bull's-eye and the rank layout? The rank layout is a special instance of the bull's-eye layout, where all the input weights have the same value. However, there are following important differences between these two related layouts: 1) The rank layout algorithm will place the interior icons in different locations, whereas bull's-eye layout can map different interior icons to the same location. 2) The rank layout

treats the icons of rank two, which are related to non-adjacent criterion icons, in a special way and duplicates them to ensure that they are as close as possible to their related criterion icons as well as at the correct distance from the center. 3) The radius of the circle on which an icon has to be placed is determined in a slightly different fashion in the rank layout than in the bull's-eye layout (see section 3.4).

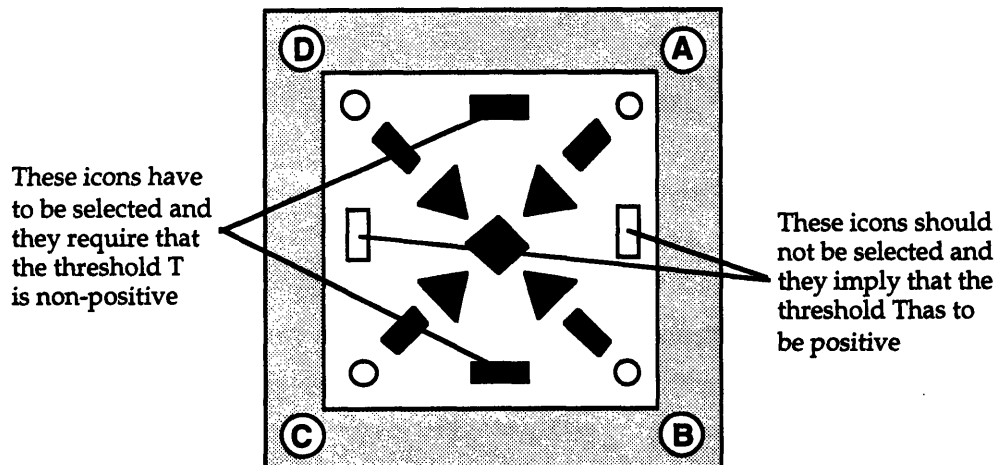
The goal of information visualization is to transform large, multi-dimensional data sets or information spaces into a visual abstraction that makes explicit the type of relationships users are interested in exploring. In the current context we want to visualize the ranking of the retrieved information based on the weights that have been assigned to the search criteria. The common way to display this type of information is to use a ranked list, which, by the way, we can use in parallel to the InfoCrystal (see Figure 7.3). The problem with a ranked list is that it provides us with a limited point of view, because it confounds and obscures how the retrieved documents are exactly related to the stated interests [Spoerri 1993, Hearst 1994]. The ranked list does not suggest how the space could be explored in other ways or how users could successfully modify a query if the need arises. The InfoCrystal in bull's-eye layout mode not only provides the information contained in a ranked list, but it has the attractive feature that it also provides users with a qualitative sense of how the ranked documents are related to the input criteria.

#### 5.4 The Expressive Limits of Weighted Queries

The question arises to what extent the interior icons of the InfoCrystal and the slider interface are equivalent in terms of their expressive power. The former lets users formulate queries by interacting with the interior icons directly. The latter lets users formulate queries by choosing weights and applying a threshold. This question is worth asking because if these two representations are equivalent, then we only need to support the one that is easier for users to use. The slider interface has an intuitive appeal, but it can be proven that only  $2^{N \cdot N}$  of the  $2^{2^N}$  possible Boolean queries can be generated by assigning weights and applying at threshold. This result follows from the analysis of Linear Threshold Networks or Boolean Perceptrons [Anthony and Biggs 1992].

---

For example, it is not possible to represent the Boolean query ((A or B) and (C or D)) by applying a threshold to the linear sums of the weights. Figure 5.6 shows the interior icons that need to be selected to specify this query (where the rank layout is used because the exact weights can not be inferred). In particular, Figure 5.6 shows that not all of the icons of rank two should be selected. Hence, our proof focuses on two of the four interior icons of rank two that should be selected and on the two interior icons that should not be selected. Two of the relationships of rank two that need to be selected and hence whose relevance scores need to be equal or above the threshold  $T$  produce the following constraints, where  $a$ ,  $b$ ,  $c$  and  $d$  represent the weights assigned to the criteria A, B, C, and D, respectively:  $(a + d) - (b + c) \geq T$  [1] and  $-(a + d) + (b + c) \geq T$  [2]. Equations [1] and [2] imply that  $T \leq (a + d) - (b + c) \leq -T$  [3], which in turn implies that  $T \leq 0$  [4]. The two relationships of rank two that should not be selected and hence whose relevance scores need to be below the threshold  $T$  produce the following constraints:  $(a + b) - (c + d) < T$  [5] and  $-(a + b) + (c + d) < T$  [6]. Equations [5] and [6] imply that  $-T < (a + b) - (c + d) < T$  [7], which in turn implies that  $T > 0$  [8], causing a contradiction with equation 4 that requires  $T$  to be non-positive. Hence, the type of selection pattern that corresponds to the Boolean query ((A or B) and (C or D)) can not be created by assigning weights to the inputs and applying a threshold.



**Figure 5.6:** shows the interior icons that need to be selected to specify the Boolean query ((A or B) and (C or D)). In particular, it shows that not all of the icons of rank two should be selected. However, this type of selection pattern can not be created by assigning weights to the inputs and applying a threshold. The icons of rank two, which need to be selected, imply that the threshold should be non-positive, whereas the two icons of rank two, which should not be selected, require that the threshold be positive. Hence, we have contradictory requirements.

## 5.5 Possible Alternative for Specifying Weighted Queries

Instead of having to specify the relevance weights by interacting with the weight sliders, would it be possible that users could point within the interior of the InfoCrystal to express their preferences in a qualitative way ? The InfoCrystal uses location as an organizing principle: users can expect the retrieved information to be related in specific ways to their interests based on the specific location they select in the interior of the InfoCrystal. However, the InfoCrystal emphasizes binary relationships rather than on continuous ones. Users initially interpret the interior icons by using the relative distances to the criterion icons as a major visual cue. Would it be possible to make use of this initial tendency of users ? We are asking if we could extend the location principle to enable users to infer degrees of relatedness instead of just binary relationships.

We will now show that the interior points of an InfoCrystal do not possess the expressive power to represent all the possible vectors of relevance weights, even if we restrict ourselves to just being able to specify the directions of these vectors. Let us suppose that we would like to specify all the possible directions in the subspace of a  $n$ -dimensional space, where all the coordinates are positive. It is sufficient to just consider the direction and to ignore the magnitude of the vector, since we use the angle between the vector of relevance weights and the vector representing an information item to determine similarity.

How could we infer a direction of a  $n$ -dimensional vector from a point chosen in the interior of an InfoCrystal ? First, we can measure the distances between this point and the  $N$  criterion icons. Second, we can take the inverse of these distance measures to achieve the effect that the degree of relatedness decreases as the distance increases. Third, the  $N-1$  ratios between these  $N$  distance measures allow us to define a unique direction in the positive  $n$ -dimensional subspace. The question is now: Can we generate all the possible positive directions in this manner? The answer is no ! There are several possible proofs. The short explanation consists of pointing out that the fixed configuration of the criterion icons and the interior point that can be chosen freely as long as it stays within the interior of the InfoCrystal do not possess the necessary number of  $N-1$  degrees of freedom. If we specify the exact

---

location of the interior point so that the ratio of the distances between it and the criterion icons  $i$  and  $j$  has a desired value, then all the other ratios are also specified. The only way we could change this is to ask users to change the location of the criterion icons, but then we are back to the slider scenario, where we have to change more than one variable to specify a weighted query.

## 5.6 Discussion

We have pointed out that we only use the locations where the interior icons are placed in the rank layout to communicate meaning. The other areas are not endowed with meaning, although the human visual system has a tendency to give it meaning. Hence, the possible information density is not fully exploited [Tufte 1983 and 1990]. Why do we not make use of this unused space and increase the information density? There are at least two ways to answer this question.

First, we make use of the whole space in the bull's-eye layout and all the different points within the concentric circles represent different relevance scores and provide some indication of how the different search criteria have contributed to the scores. However, we pointed out that different interior icons can be mapped to the same point in the bull's-eye layout. Therefore, it is not possible to infer in a unique fashion from a point in bull's-eye layout how it is exactly related to interior icons. The bull's-eye layout uses a many-to-one mapping.

Second, we are interested in developing a visual abstraction that leads to an one-to-one mapping, because such a mapping not only has *descriptive* power, since it enables us to see large amounts of information in a compact way, but that also has *expressive* power that enables us, for example, to interact with the data to issue commands. In order to achieve this goal, we need to group the information. The finite number of discrete relationships among  $N$  search concepts represents one way of organizing a large and multi-dimensional information space, and it achieves our goal to arrive at a one-to-one mapping. It enables us to use the InfoCrystal as a visualization tool *as well as* a visual query language. This versatility and expressiveness comes at a price, because we have to sacrifice information density. Hence, there is a trade-off between the information density we can achieve and the expressive power of the resulting visualization.

---



## CHAPTER 6

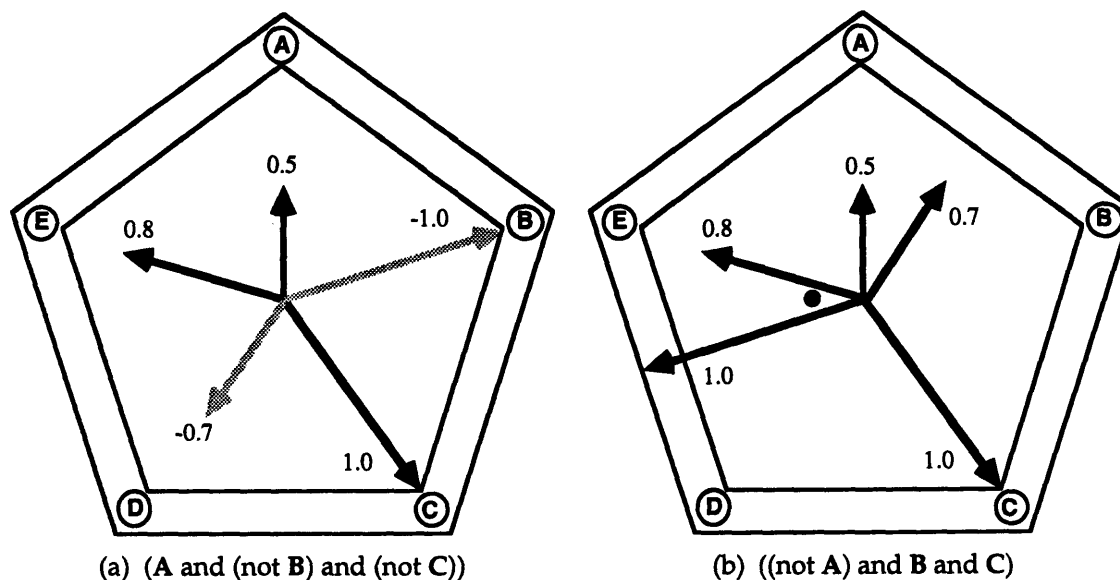
# VISUALIZING VECTOR SPACE QUERIES

### 6.1 Introduction

How is it possible to integrate and visualize the competing, but complementary Boolean and Partial Matching approaches in the same visual framework to enable users to make effective use of their respective strengths? The InfoCrystal can be generalized to formulate and visualize vector-space queries. The vector-space approach computes the relevance score from the weights assigned to the index terms that represent the query and the document, respectively [Salton 1983]. These weights reflect how well the index terms describe the content of a document and a query, respectively. In chapter 5, we have demonstrated how the InfoCrystal can be used to formulate and visualize weighted queries. We also introduced the bull's-eye layout that uses the centers of mass of the interior icons to compute their locations. We can generalize the way the center of mass is computed so that we can consider vector space queries: the components of a document vector can now have values between -1 and 1 to reflect the degree to which they do (not) describe the document's content. The two-dimensional vector, which points from the InfoCrystal's center in the direction of a criterion icon, is now scaled by the degree to which the document's content does (not) satisfy the criterion (see Figure 6.1).

Figures 6.2 and 6.3 show how the discrete version of the InfoCrystal is related to the continuous one that can visualize vector space queries. Figure 6.2 shows that the document vectors, whose components are positive with respect to criteria A and B, but negative for C, cluster in the vicinity of interior icon that represents the relationship (A and B and (not C)). Similarly, Figure 6.3 shows that the document vectors, whose components are positive with respect to criterion A, but negative for B and C, cluster in the vicinity of interior icon that represents the relationship (A and (not B) and (not C)).

---



**Figure 6.1:** shows how to compute the center of mass for a document vector equal to  $(0.5, -1.0, 1.0, -0.7, 0.8)$ . (a) The two-dimensional vectors pointing from the InfoCrystal's center towards the criterion icons are scaled based on the corresponding components of the document vector. If the vectors are scaled according to the masses associated with them then (b) shows the resulting vectors. The solid circle shows the location of the center of mass if the weighted average of the vectors is taken.

We can think of the discrete case as the limit of the continuous case. The documents, which satisfy the same criteria and are therefore represented by the same interior icon in the discrete mode, will cluster in an orderly fashion in the continuous mode (see Figures 6.5 and 6.7). The difference between the continuous and the discrete versions of the InfoCrystal is that in the former the dots displayed in its interior represent individual documents, whereas in the latter the interior icons represent how a collection of documents is related to the criterion icons. The continuous version of the InfoCrystal allows users to visualize an information space at the level of the individual documents based on their ranked relevance scores. Documents with high relevance scores are displayed closer to the center of an InfoCrystal. Documents with low relevance scores are displayed further away from the center, where the ones with lowest possible score lie on the outermost circle shown in the interior of an InfoCrystal (see Figure 6.4). The polar transform used in the bull's-eye layout to map the documents has the attractive feature that it not only visualizes the ranking of the relevance scores, but it also provides users with a qualitative sense of how the documents are related to the input criteria. Figure 6.4 shows the resulting distribution pattern of the relevance

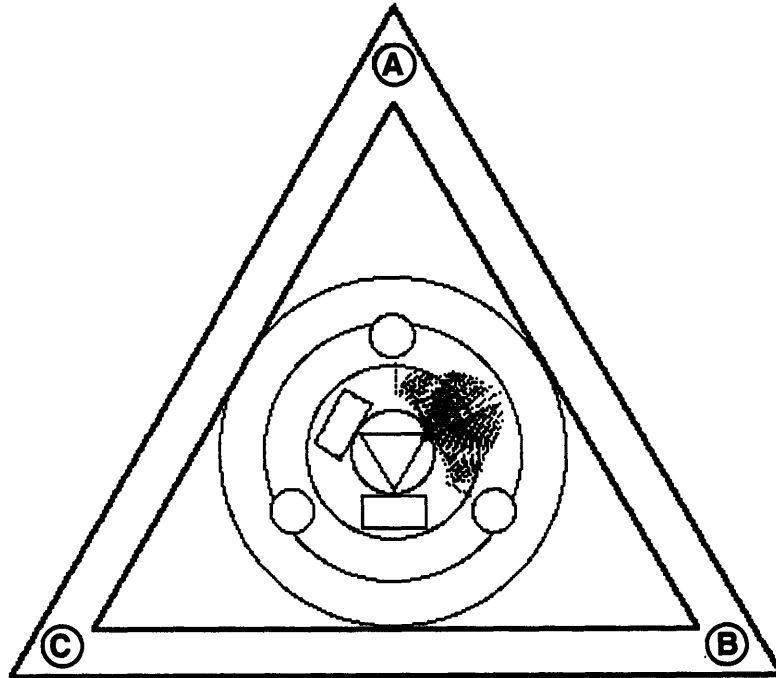
scores if we uniformly sample all the document vectors that lie in the cube  $\{[-1, 1]; [-1, 1]; [-1, 1]\}$ . In section 6.3 we analyze in more detail why we get the type of distribution patterns that we can observe in Figures 6.4, 6.6, 6.8 and 6.9. Figures 6.5 (a) to (c) show that documents that are related in a specific way to the search criteria will cluster in the locations where we would expect them to do so. Hence, the proximity or location principle is preserved by the polar transform. In Figure 6.6, for example, the input weights are equal to  $(1, 1, -1)$ , and as expected the documents with the lowest score are displayed close to the criterion icon A for the following reasons: 1) The documents that satisfy the criterion A but not the criteria B and C will receive the lowest score. Hence, these documents should be displayed the furthest away from the center. 2) On the basis of the proximity principle, we expect documents that only satisfy A to be displayed closer to the criterion icon representing A and further away from the other two criterion icons.

## 6.2 Visualizing Any Ranking Function

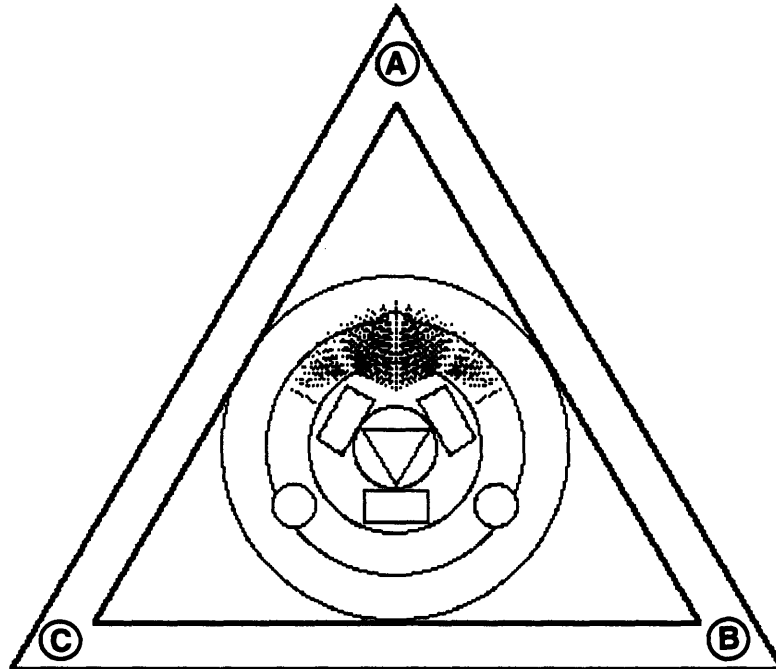
The InfoCrystal is flexible in terms how the relevance score and therefore the radius value is calculated. Hence, we can use, for example, the probability estimates of the document's relevance or the distance-based p-norm to rank the documents [Fox 1983, Belkin and Croft 1992]. We could also use the degree to which a document satisfies the query in terms of coordination, proximity, field level and stemming to rank the retrieved documents [Marcus 1991]. Further, we can decouple the computation of the relevance score from the specified interests and actually use many more criteria than ones that are made explicit in an InfoCrystal. The computation of the center of mass can remain linked to the specified criteria. In short, the InfoCrystal can be used to visualize any ranked list or fuzzy set, where the way the items relate to the specified reference or search criteria is used to compute the center of mass.

The InfoCrystal representation also opens up the possibility that users can visually specify (several) arbitrarily shaped areas, where the documents contained within them would define the output of the InfoCrystal. This way of selecting a subset from a ranked list would be impossible to perform by pruning a ranked linear list by setting multiple thresholds.

---

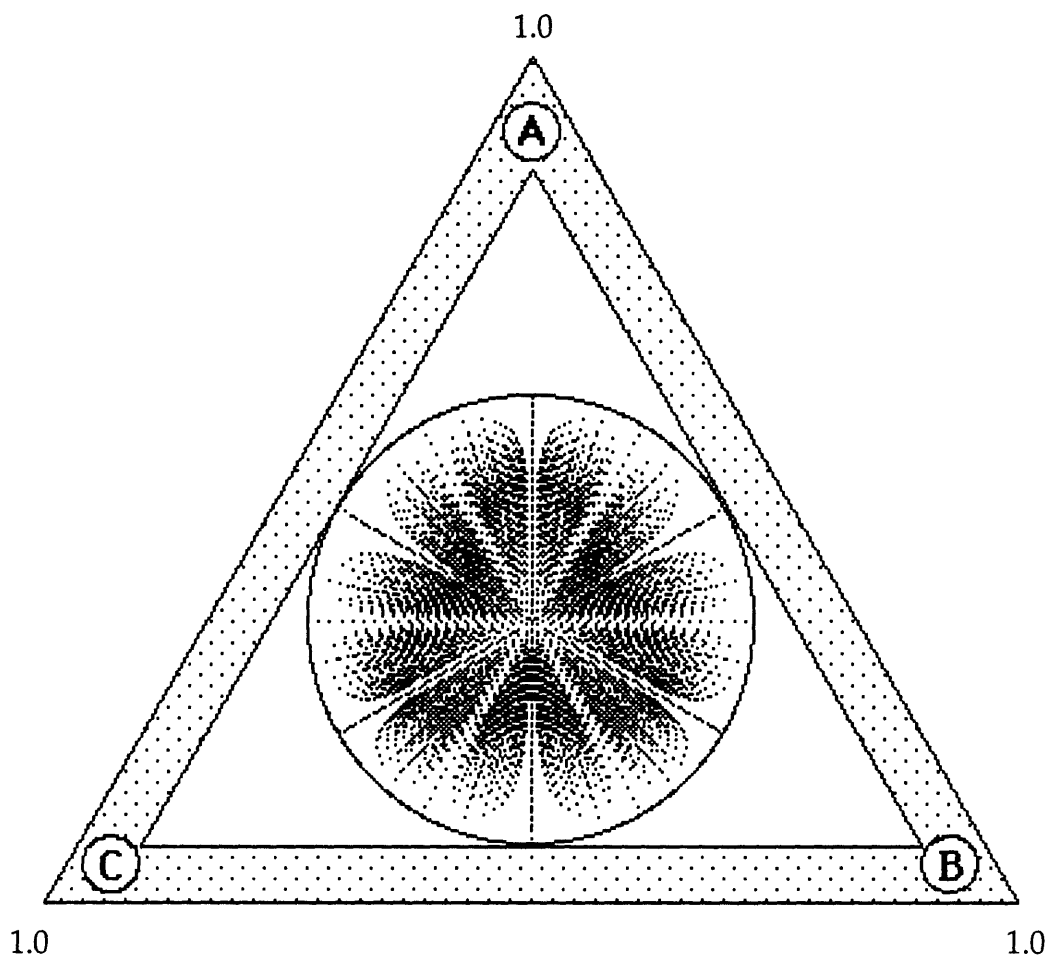


**Figure 6.2:** shows the relationship between the discrete and continuous version of the InfoCrystal, where the document vectors, whose components are positive with respect to criteria A and B, but negative for C, cluster in the vicinity of interior icon that represents the relationship (A and B and (not C)). The interior icons are displayed using the bull's-eye layout, where the weights assigned to the criteria are (1,1,1).

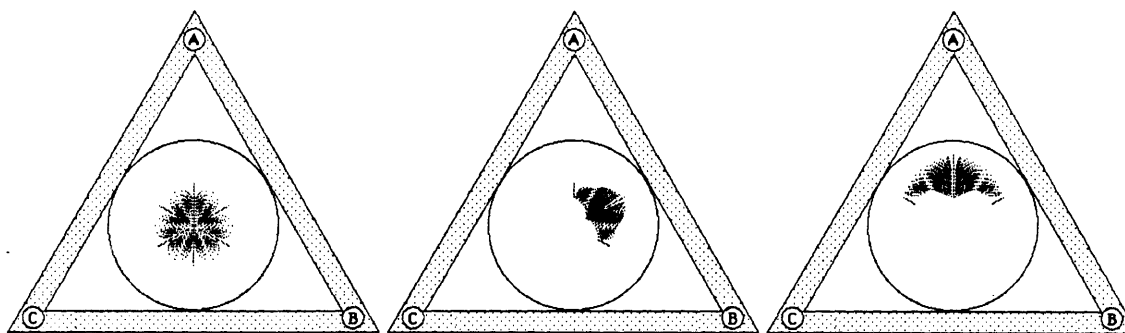


**Figure 6.3:** shows that the document vectors, whose components are positive with respect to criterion A, but negative for B and C, cluster in the vicinity of interior icon that represents the relationship (A and (not B) and (not C)). The weights assigned to the criteria are (1,1,1).

---

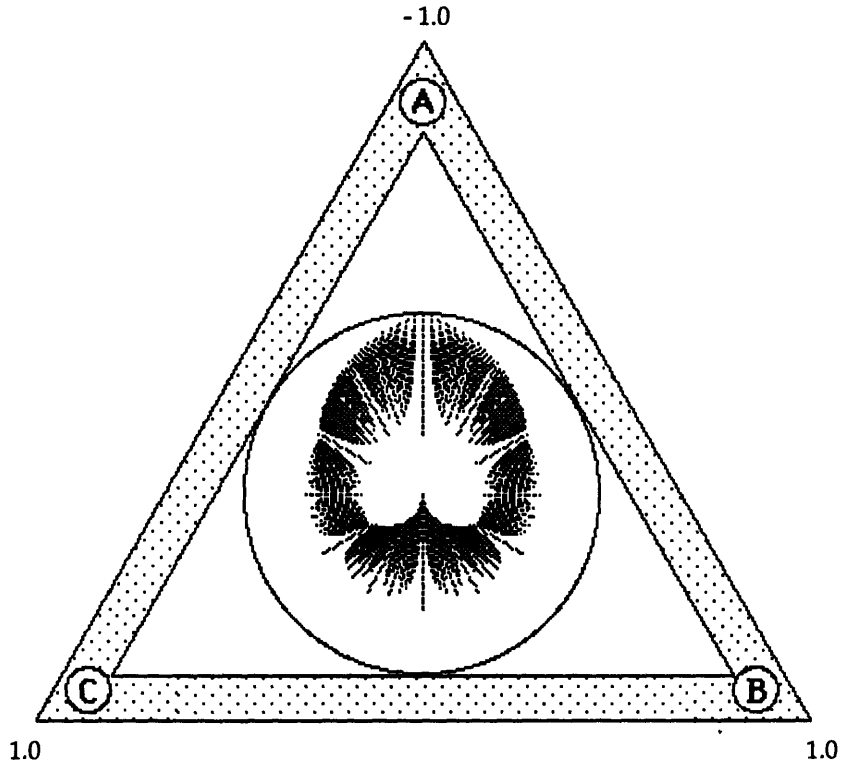


**Figure 6.4:** shows the distribution pattern of the relevance scores of a uniform sampling of all the document vectors that lie in the cube  $\{[-1, 1]; [-1, 1]; [-1, 1]\}$ , where we use the bull's-eye layout principle that reflects the values of the criteria weights, which are equal to  $(1.0, 1.0, 1.0)$ . The black points within the circle represent individual documents.

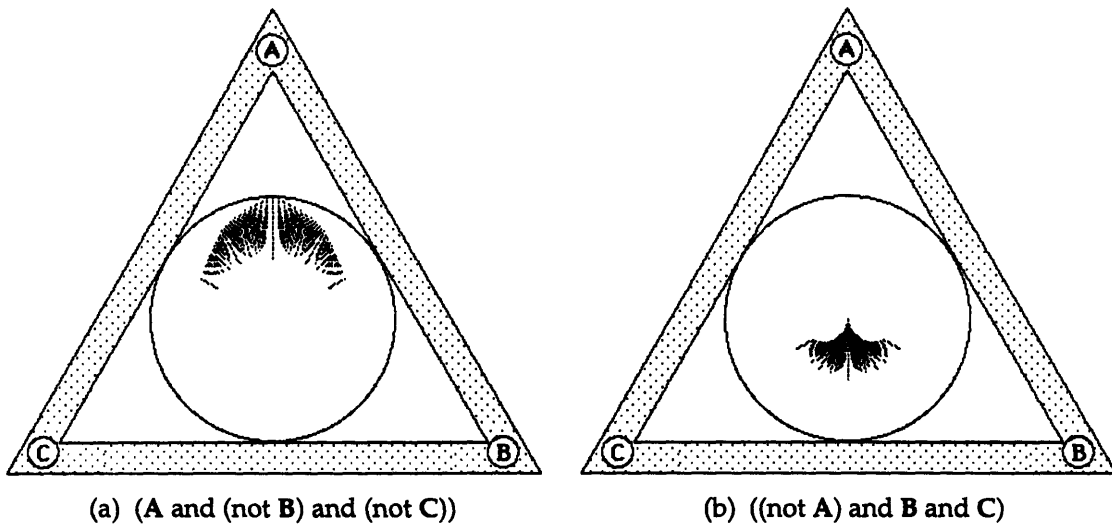


[a] (A and B and C)      [b] (A and B and (not C))      [c] (A and (not B) and (not C))

**Figure 6.5:** If the weights associated with the search criteria A, B and C are all equal to one, then (a) shows that the documents that are related to A, B and C in a positive way will cluster in the center of the InfoCrystal, which is where we would expect to find them; (b) displays where the documents that are related in a positive way to A and B and in a negative way to C will cluster. (c) shows where the documents that are related in a positive way to A and in a negative way to B or C will be located.



**Figure 6.6:** shows the distribution pattern of the relevance scores of a uniform sampling of all the document vectors that lie in the cube  $[-1, 1]; [-1, 1]; [-1, 1]$ , using the bull's-eye layout principle that takes into account the values of the criteria weights, which are equal to  $(-1.0, 1.0, 1.0)$ .



**Figure 6.7:** If the weights associated with the search criteria A, B and C are equal to  $-1, 1$  and  $1$ , respectively, then (a) shows that the documents that are related to A, but not B or C, will cluster close to A, which is where we would expect to find them; (b) displays where the documents that are related to B and C in a positive way and to A in negative will cluster in and close to the center, but away from A, which is again where we would expect them to be located.

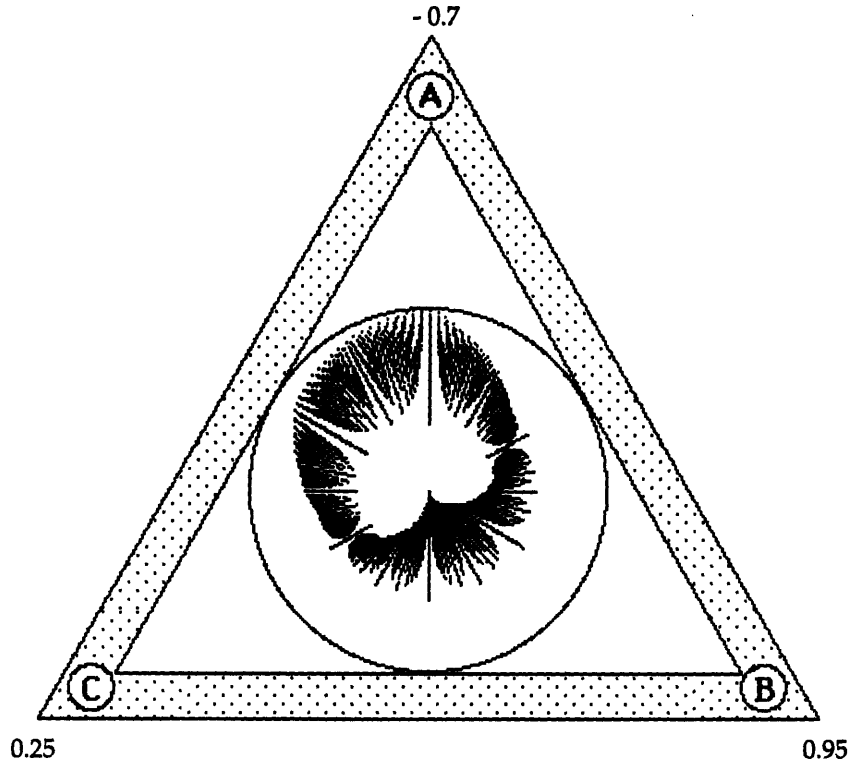


Figure 6.8: shows the distribution pattern of the relevance scores of a uniform sampling of all the document vectors, where the values of the criteria weights are equal to  $(-0.7, 0.95, 0.25)$ .

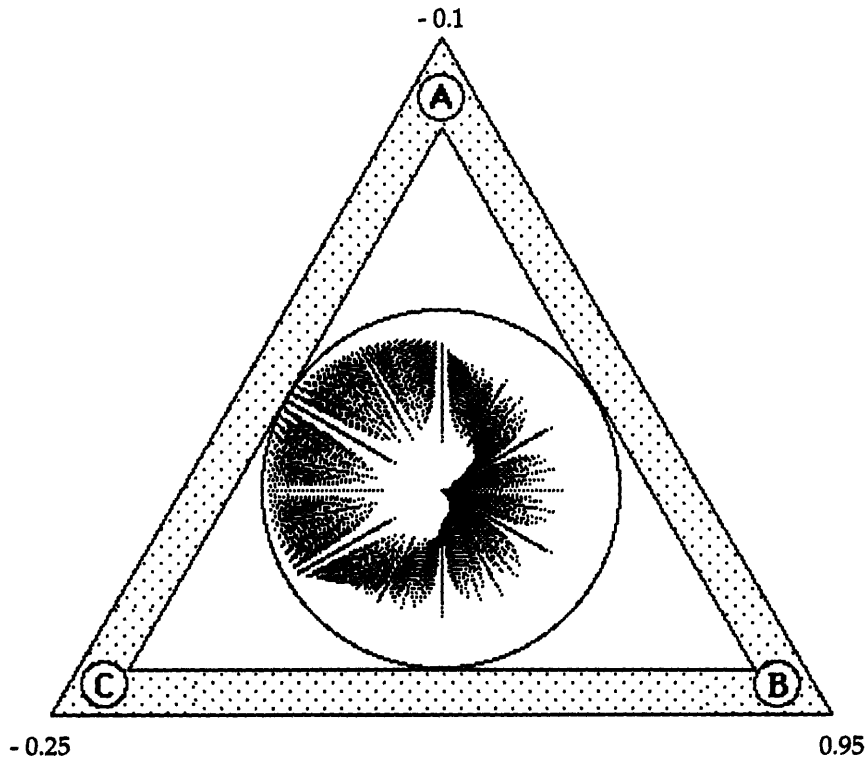


Figure 6.9: shows the distribution pattern of the relevance scores of a uniform sampling of all the document vectors, where the values of the criteria weights are equal to  $(-0.1, 0.95, -0.25)$ .

### 6.3 The Continuous Bull's-Eye Mapping

In this section we discuss in detail how we compute the continuous version of the bull's-eye layout. Further, we will analyze why we get the types of distribution patterns that we can observe in Figures 6.4 to 6.9 when we map a  $n$ -dimensional vector space representation into a two-dimensional InfoCrystal with  $n$  concepts.

To remind the reader, a document is represented by a  $n$ -dimensional vector, whose components can take values between  $-1$  and  $1$  inclusive. A negative value of  $-1$  implies that a concept is not at all present in the document, whereas a positive value indicates that the concept is present to a degree proportional to the component value. The query is also represented by a  $n$ -dimensional vector, whose components can take values between  $-1$  and  $1$  inclusive. A negative value implies that we are not interested in the corresponding concept and a positive one that we are interested to a degree proportional the weight.

The bull's-eye mapping uses a two-dimensional polar transform where the radius and the angle are defined as follows: 1) The radius is equal to the relevance score, which is computed by taking the cosine of the angle between a document vector  $\underline{d}$  and the weight vector  $\underline{w}$ . 2) The angle is defined by the line that passes through the InfoCrystal's center and the center of mass of the criterion icons. The center of mass is computed as follows: the two-dimensional vectors pointing from the InfoCrystal center towards the criterion icons are scaled based on the corresponding components of the document vector (see Figure 6.1). The center of mass is equal to the weighted average of these scaled vectors. Thus, the center of mass and therefore the document will be closer to those criterion icons that they are related to in a positive way than to those criterion icons for which this is not the case. Further, the angle is not affected by the weights. There are document vectors whose center of mass will coincide with the center of the InfoCrystal and therefore the angle can not be specified. In these cases, we place the document where the line, which passes through the first criterion icon that is satisfied to some degree by the document, intersects the circle defined by the relevance score .

---



To provide the reader with a better insight into the distribution pattern that result when we map a document space into an InfoCrystal, we want to characterize the geometrical surfaces that are defined by documents with a specific relevance value or whose center of mass lies along a particular line (which is equivalent to saying that they have a particular angle). We show below that a particular relevance score defines a cone and that the angle specified by the center of mass defines a plane passing through the origin. We begin with the surface defined by a relevance score:

$$r = \text{relevance-score} = \cos \alpha = \frac{\underline{d} \bullet \underline{w}}{|\underline{d}| \cdot |\underline{w}|}$$

which defines a cone with angle  $\alpha$  in the direction of  $\underline{w}$  and its apex is at the origin. In order to show that documents, whose centers of mass lie on a particular line in the InfoCrystal with  $n$  inputs, define a plane, we need to examine how we compute the center of mass<sup>1</sup>:

$$\text{center of mass for } \underline{d} = \sum_{1 \leq i \leq n} d_i \left[ \cos\left(\frac{(i-1) \cdot 360^\circ}{n}\right), \sin\left(\frac{(i-1) \cdot 360^\circ}{n}\right) \right]$$

if  $n = 3$  then

$$= [d_1 - 0.5 \cdot (d_2 + d_3), \sqrt{3}/2 \cdot (d_2 - d_3)]$$

This center of mass can be used to define a straight line that passes through the origin and that has an angle equal to  $\alpha$ .

If we want to select only documents whose centers of mass lie on a straight line that passes through the origin and that has an angle equal to  $\alpha$  then we get the following constraint:

$$k = \tan \alpha = \frac{\sqrt{3}/2 \cdot (d_2 - d_3)}{d_1 - 0.5 \cdot (d_2 + d_3)}$$

---

<sup>1</sup> For simplicity, we only show the formula that does not include the scaling factor, which is equal to the sum of the absolute components of the document vector, that is used to compute the weighted average.

We can rewrite this equation to arrive at the following equation:

$$0 = 2k \cdot d_1 - (\sqrt{3} + k) \cdot d_2 + (\sqrt{3} - k) \cdot d_3$$

which defines a plane that passes through the origin and its normal is equal to:

$$\text{normal to plane} = [2k, -(\sqrt{3} + k), (\sqrt{3} - k)]$$

To summarize, documents that have a specific relevance value lie on a cone, whose angle is equal to arc cosine of the relevance value and whose apex is at the origin. Documents whose center of mass lie along a particular line lie on a plane that passes through the origin. If examine the Figures 6.4, 6.6, 6.8 and 6.9 more carefully, then we notice that there are areas inside the interior circle of the InfoCrystal that do not have any dots. Hence, it appears that for specific relevance values and angles that there are no corresponding documents. The nature of the two surfaces described above enables us to explain why this is the case. We have a cone and a plane that both pass through the origin. These two surfaces will have points in common other than the origin only if the normal of the plane lies inside the space created by a cone with the same angle as the cone that is defined by the relevance score, called the relevance cone, and by sweeping its axis perpendicular to the axis of the relevance cone. Hence, we can easily imagine particular cone orientations and plane normals that only intersect at the origin, and in these cases the corresponding location in the interior of the InfoCrystal will remain empty. The frequency and the particular combinations of relevance scores and angles defined by a center of mass for which this occurs is a function of the chosen relevance weights.

---

## 6.4 Discussion

A primary goal of this thesis is to create visual abstractions that can be used both as a visualization tool and as a visual query language. We have shown how the InfoCrystal can be both used to visualize Boolean and vector space queries. The question arises: which parts of the InfoCrystal are used to accomplish this versatility and what are their relationships? In the discrete version of the InfoCrystal the area used for visualization purposes and the one for specifying queries coincide. In the continuous version of the InfoCrystal the interior area is used for visualizing the ranking of the contents of an information space, and the sliders attached to the InfoCrystal are used for specifying the retrieval request. Hence, the visualization and query language components are performed by different visual entities. Finally, the interior can be used for both for visualization and specifying Boolean queries in the case where the InfoCrystal is used to formulate weighted queries. The weight and threshold sliders can be used to specify and control how a weighted query is translated into a Boolean query.

---



# CHAPTER 7

## INFOCRYSTAL SOFTWARE

### 7.1 Introduction

This chapter provides a brief overview of the key features of the InfoCrystal that have been implemented and that have not been discussed in great detail elsewhere in this thesis. The InfoCrystal has been implemented using the object-oriented MacLISP programming language for the Macintosh computer.

We will demonstrate some of the key features of the InfoCrystal by describing how it can be used to retrieve information. We begin by creating a structured-list or outline of our information need by using the query outliner tool, which functions like the familiar outlining tool available in word-processing packages (see Figure 7.1). Once the query outline has been generated, we can issue the command to have it evaluated and visualized. What does it mean to have an InfoCrystal query evaluated? The atom or "leaf" nodes of the query structure represent the criteria that the user has decided not to break down any further. The atoms specify the query statement that a retrieval engine will use to search in the selected database(s). The choice of the query statement and its corresponding retrieval engine is absolutely flexible. The query statement could be a reference document and we specify that all the documents with a certain degree of similarity should be retrieved. The query statement could be a concept from a thesaurus and we could use the explode feature to retrieve all documents that have this concept as well as all its children concepts. The query statement could be a simple keyword or a complex Boolean statement. In short, the InfoCrystal works for any retrieval method and its retrieved set of data objects. The InfoCrystal uses an object-oriented design and it is therefore easy to support any data objects and their retrieval methods.

For the purpose of this thesis, we primarily used synthetic data sets created by using a random generator that would specify the database id of a data

---

object. For each of the atom inputs this generator would select in a random fashion from a range of possible id values. This way of generating the input data streams highlights that the InfoCrystal works for arbitrary data object. We also developed and experimented with a database driver to retrieve book and technical report abstracts stored in the on-line library of the Laboratory of Computer Science at MIT. However, we were not satisfied by the slow retrieval performance. We decided to not invest more energy at this stage to improve its performance characteristics. We will design a diverse set of fast database drivers in the future when we have migrated the InfoCrystal to a more powerful platform.

Once the input sets at the atom nodes have been computed, then the retrieved data items are propagated through the query structure based on the way the different InfoCrystals have been programmed, i.e., how the interior icons have been selected. When we create a structured-list, we do not have to specify any operators. The current default is to select all the interior icons in an InfoCrystal, which is equivalent to the Boolean OR. Hence, we can observe at the root and top-level InfoCrystal the results of performing the broadest possible query. We have mentioned that retrieval specialists often suggest to searchers to generate queries, where quasi-synonymous words for each conceptual factor are ORed and these different synonym lists are then ANDED [Cooper 1988, Marcus 1991]. Our default selection of the interior icons generates a query that is equivalent to one suggested by retrieval specialists. The Boolean AND operator is only of relevance at the top-level InfoCrystal, because its inputs are the ORed synonyms, and the center interior icon reflects the effect of applying the AND to these inputs. A key advantage of the InfoCrystal is that it not only shows the effects of the AND operation but all the other possible Boolean operations involving the inputs. Similarly, our default selection of the interior icons retrieves the same documents as would be retrieved by a vector space query. In contrast to the ranked list generated by a vector space query, the InfoCrystal not only presents the documents in a ranked order, but it presents them in a structured way that reveals how the documents are related to the specified interests. The InfoCrystal emphasizes relationships and ranks them based on their relevance. The ranked-list displays the documents based on their relevance. The current implementation also provides a *ranked-list interface* that displays the

---

individual items retrieved by an individual interior icon or by all the selected icons of an InfoCrystal (see Figure 7.3). As we mentioned in chapter 2, users can provide relevance feedback by selecting the items in the ranked-list that they consider as satisfying their information need. We could use this relevance feedback to determine which of the selected interior icons in the query structure should remain selected.

Once the InfoCrystal query structure has been visualized and its contents initialized, we can see how the retrieved data items distribute across different relationships provided we display the selected icons in the pie-chart or number mode (see Figure 7.3). An interactive *state-sheet* is associated with an InfoCrystal and it has a set of radio-buttons that indicate the visual style of the selected and not selected interior icons, respectively (see Figure 7.2). The following styles can be selected: icon&border, icon, point, number, and pie-chart. A state-sheet contains also buttons to perform the following actions (from top to bottom): to change the size of an InfoCrystal; to change the scale at which the interior icons using pie-chart style are displayed; to reveal or hide the interior icons; to display the interior icons using either the rank or the bull's-eye layout; and to descend in the query structure and make the selected child the new top node from which to visualize the query structure, or to ascend and make the parent InfoCrystal the new top node. The descend or ascend operations do not modify the query and they are equivalent to a zoom operation.

We can use the standard copy, cut, and paste operations to modify the InfoCrystal query structure. If we want to add a new input to an InfoCrystal, then we first need to specify the new input InfoCrystal by selecting it and using the copy or cut operation. Next, we need to select the recipient InfoCrystal and apply the paste operation. Hence, it is very easy to modify existing queries or to create new queries by combining and integrating existing InfoCrystal queries. We can also reorganize the query structure using click-drag-drop operations (see Figures 7.8, 7.10 and 7.11).

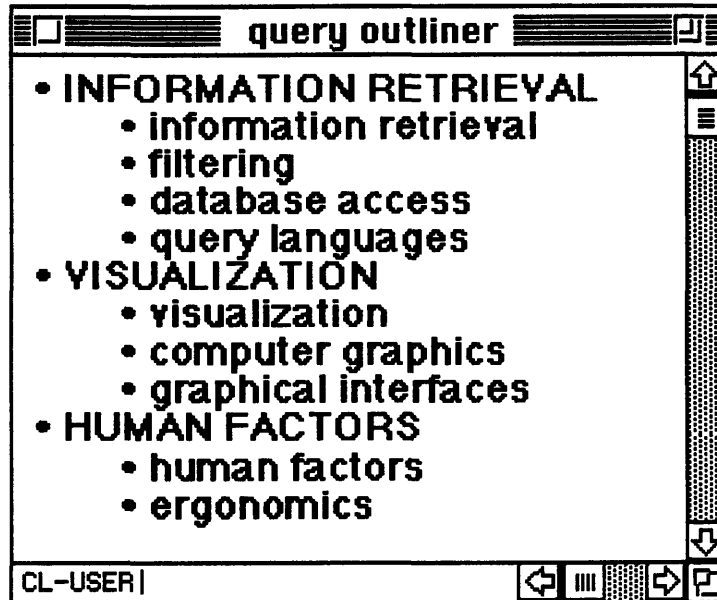
---

## **7.2 InfoCrystal Software in Pictures**

We will now provide visual examples of the major InfoCrystal operations that have been implemented. In particular, we will show how we can create an InfoCrystal query structure, change its appearance at the structural as well as at the individual InfoCrystal level. We will show how we can navigate InfoCrystal query structure and descend or zoom in to be able to examine an input InfoCrystal in more detail. We will perform a what-if analysis by changing how the retrieved data is propagated through the query structure. We will show how we can modify the query structure by selecting an InfoCrystal, dragging and dropping it in the desired new location, where the structure is automatically updated and the content assignments are recomputed.

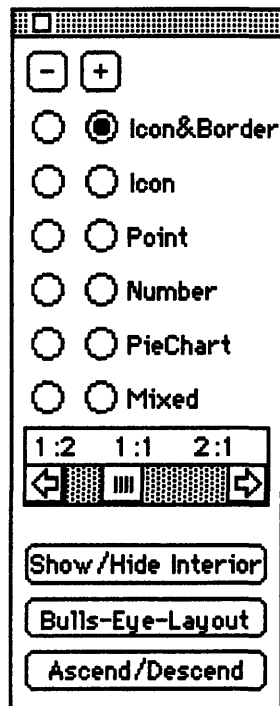


## How to get started ? Create an outline.



**Figure 7.1:** shows the query outline that we need to generate to begin the process of retrieving information.

## State-Sheet of an InfoCrystal



**Figure 7.2:** shows the *state-sheet* that is linked with an InfoCrystal and it consists of the following elements (top to bottom):

1) The buttons (+) and (-) change the size of the InfoCrystal.

2) Two sets of radio-buttons that indicate the visual style of the selected and the not selected interior icons (lined up below the (+) and (-) buttons), respectively.

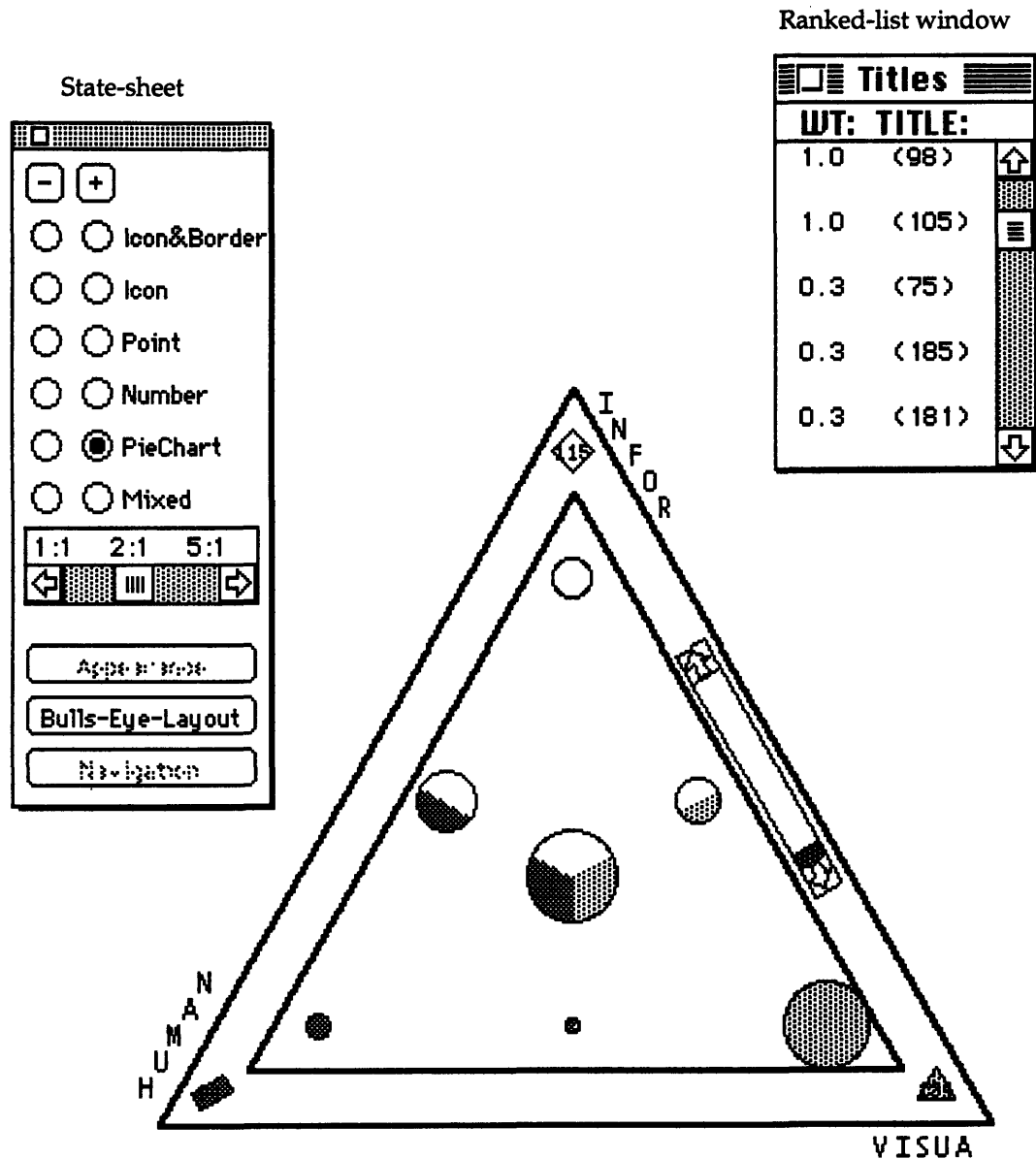
3) A slider that can be used to specify the scale at which to display the interior icons using the pie-chart style.

4) A button that can be used to either show or hide the interior icons.

5) A button to display the interior icons using either the rank or the bull's-eye layout.

6) A button that can be used to descend in the query structure and make the selected child the new top node from which to visualize the query structure, or to ascend and make the parent InfoCrystal the new top node.

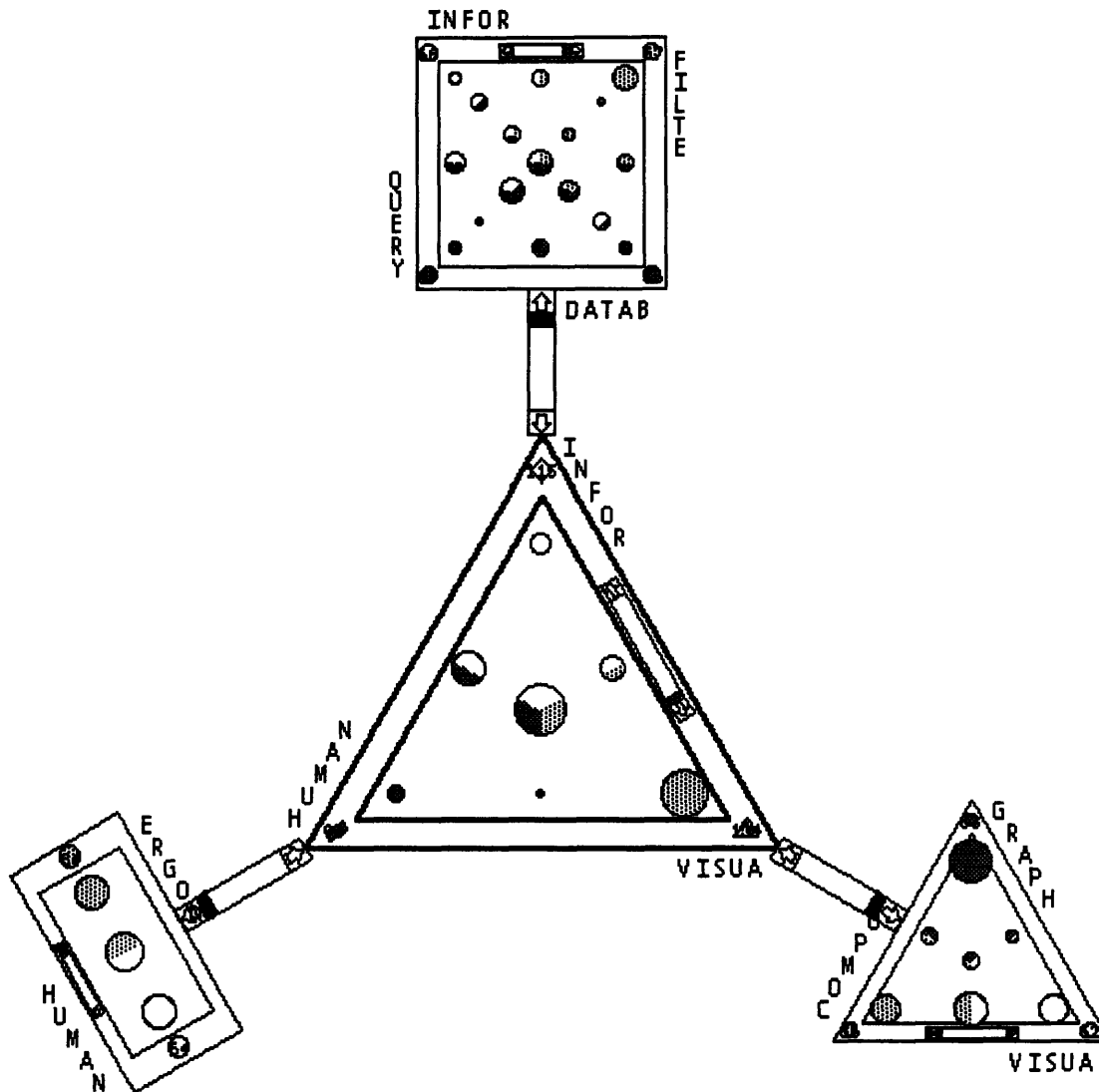
## The query outline has been visualized as an InfoCrystal



**Figure 7.3:** shows what users will see when they execute and visualize the query outline shown in Figure 7.1. Only the root-node InfoCrystal is displayed. Its interior icons use the pie-chart style to show how the retrieved documents distribute across the different relationships. The button of the state-sheet that lets users change the appearance of the InfoCrystal is inhibited, because the interior of the root-node InfoCrystal is always visible, provided the crystal is visible. The button that lets users navigate up/down in the query structure is also inhibited, because by definition the root node has no parent, hence there is nowhere to ascend to.

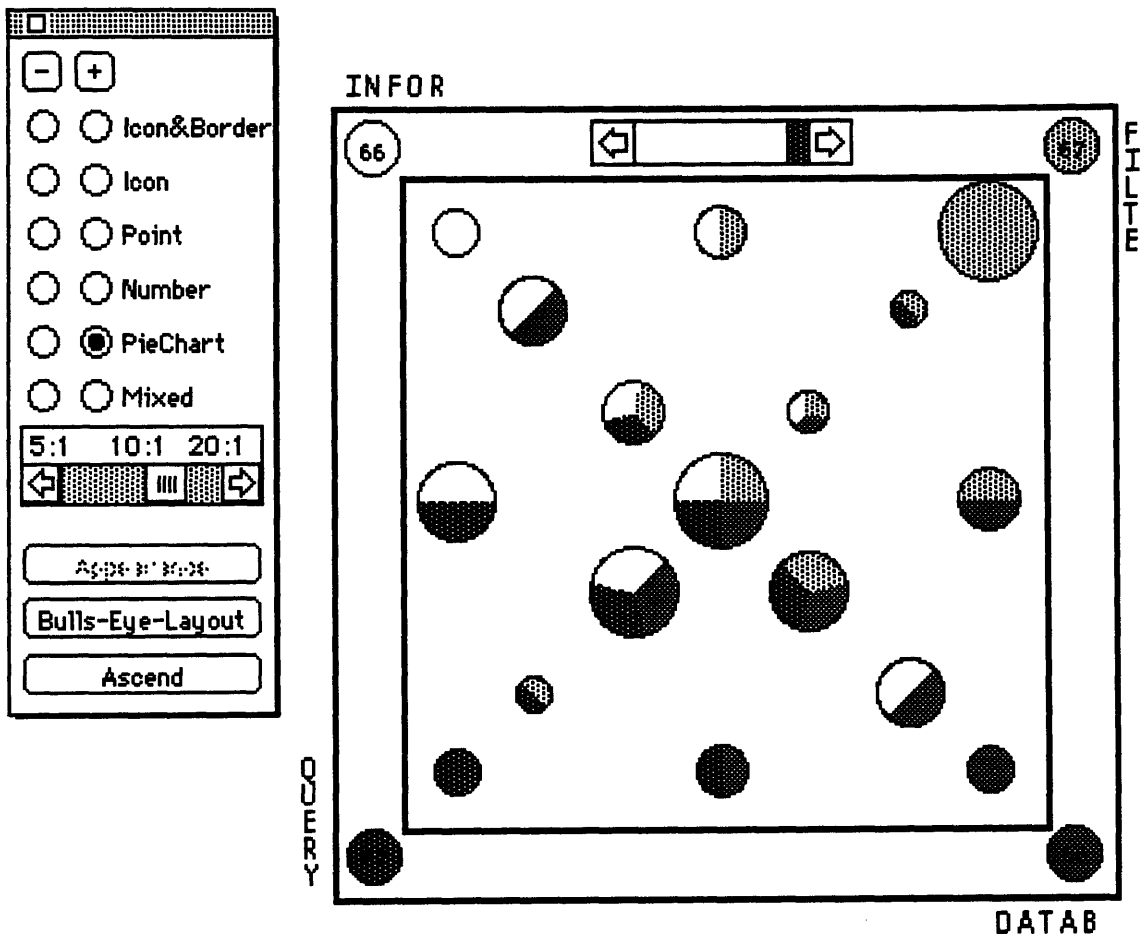
The ranked-list window displays a ranked list of all the retrieved documents, where the left column contains the weights and the right column the document ids. We can double-click on a list item to see its contents.

## Displaying the InfoCrystal query structure one level deep



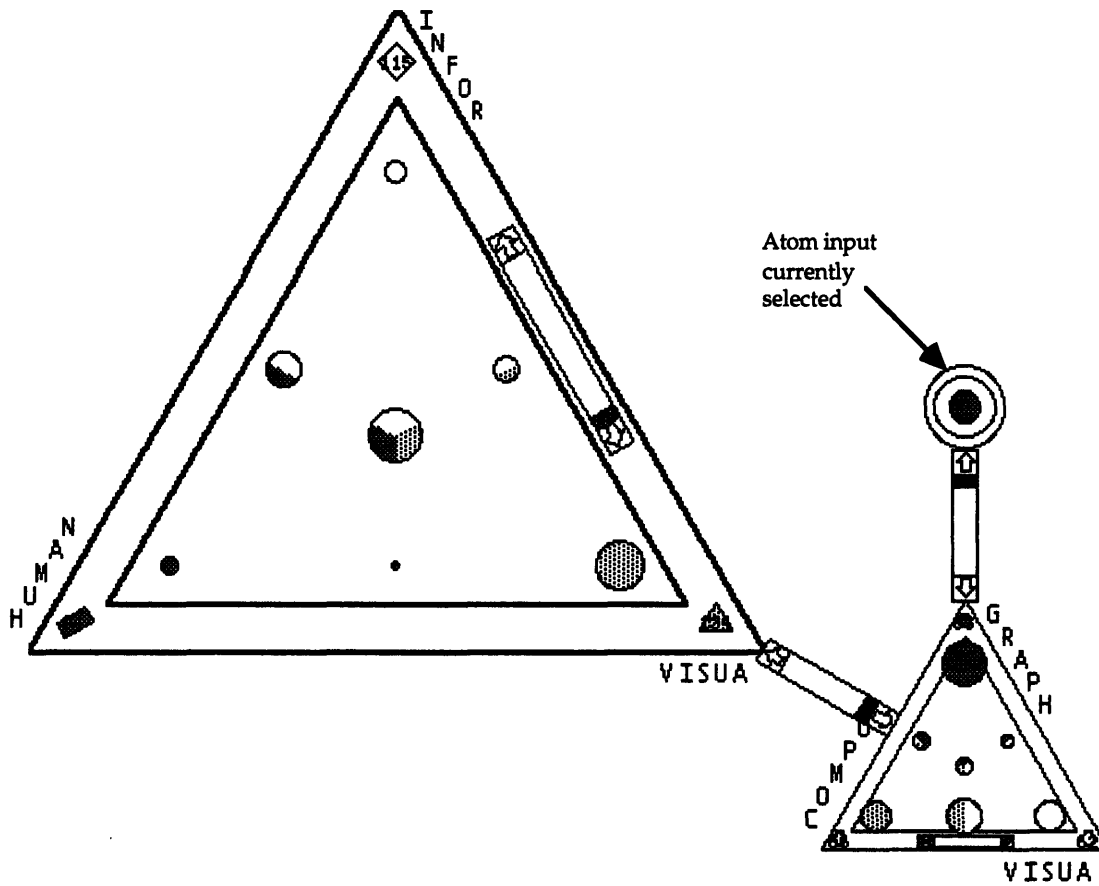
**Figure 7.4:** displays the InfoCrystal query structure one level deep by showing the root node and its children. Users can show or hide a child InfoCrystal by double-clicking on the criterion icon representing it in the parent InfoCrystal. Users can select a particular InfoCrystal by clicking on it, and the state-sheet will automatically be updated to reflect the states of the newly selected InfoCrystal. If users wanted to explore the four-concept InfoCrystal (shown at the very top) in more detail and promote it to be the new top node that is visible, then they need to select it and click on the "Descend" button (which is at the very bottom of the state-sheet, but not shown here).

## Descending in the query structure



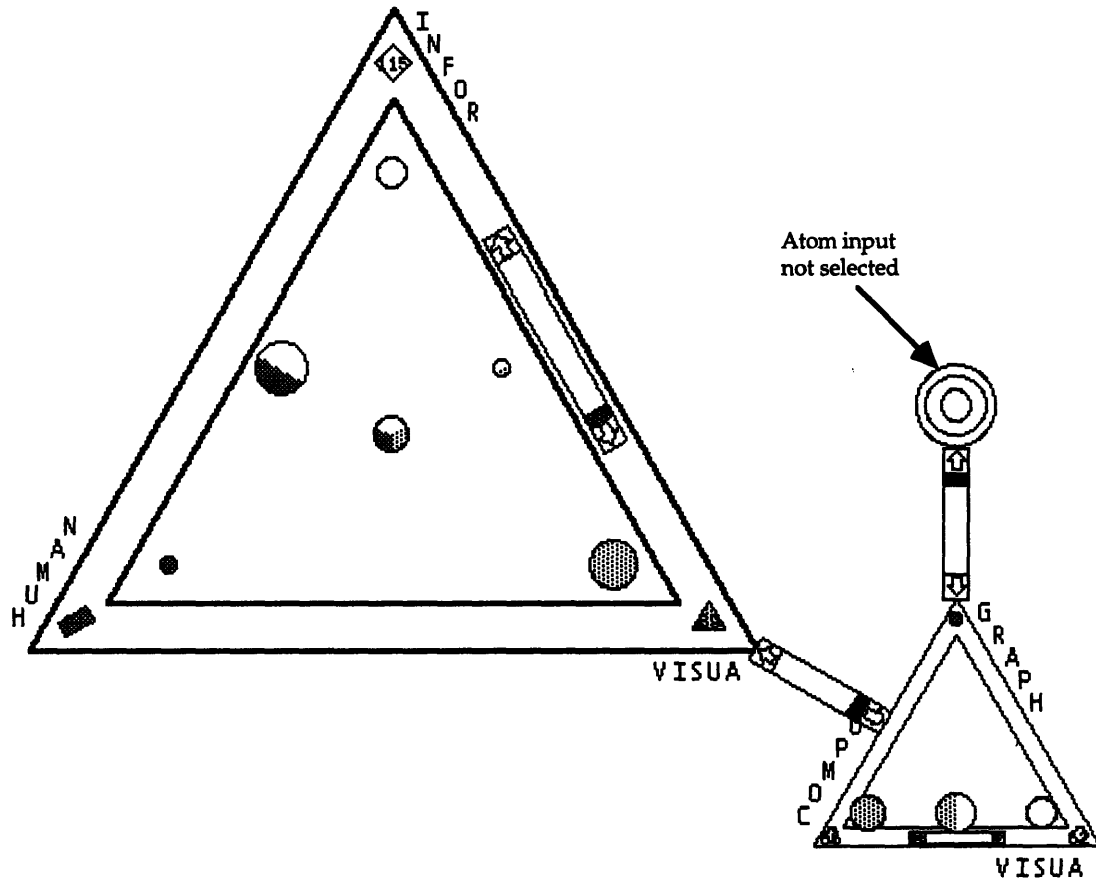
**Figure 7.5:** shows the result of selecting a child InfoCrystal and promoting it to be the current top-node that is visible. The navigation button in the state-sheet has changed to say "Ascend", because this four-concept crystal has a parent. If we were to select it then we would return to state of affairs shown in the previous figure.

### What-if Analysis (before)



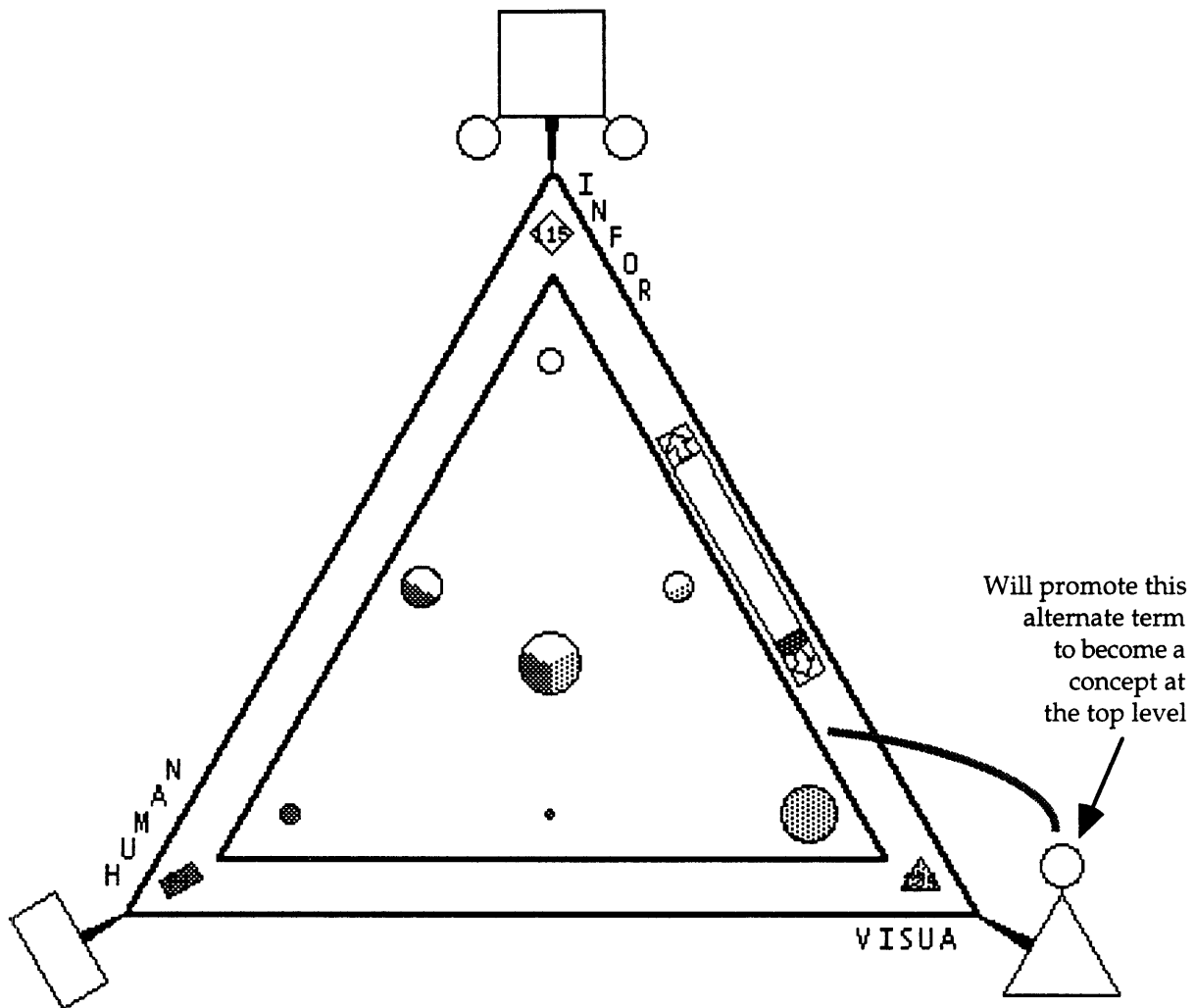
**Figure 7.6:** displays how the retrieved data distributes across the different possible relationships, when the data elements associated with the leaf-node, whose circular InfoCrystal is displayed in full detail and where we can see that its singular interior icon is selected (darkly shaded), are propagated through the query structure.

### What-if Analysis (after)



**Figure 7.7:** shows the effects of suppressing the propagation of the data elements that are associated with the circular InfoCrystal shown in full detail (its singular interior icon is shown in solid white to indicate that it is not selected). One of the clearly visible consequences of this action is that there are now no data elements anymore that are related to the concepts "visualization" and "human factors" but not "information retrieval". The change in the distribution of the data elements will be readily perceptible because the size of the pie-charts will change and hence create a motion or animation effect. The suppression of this circular crystal is equivalent to dropping an alternate term and in effect reducing its parent to a two-factor crystal.

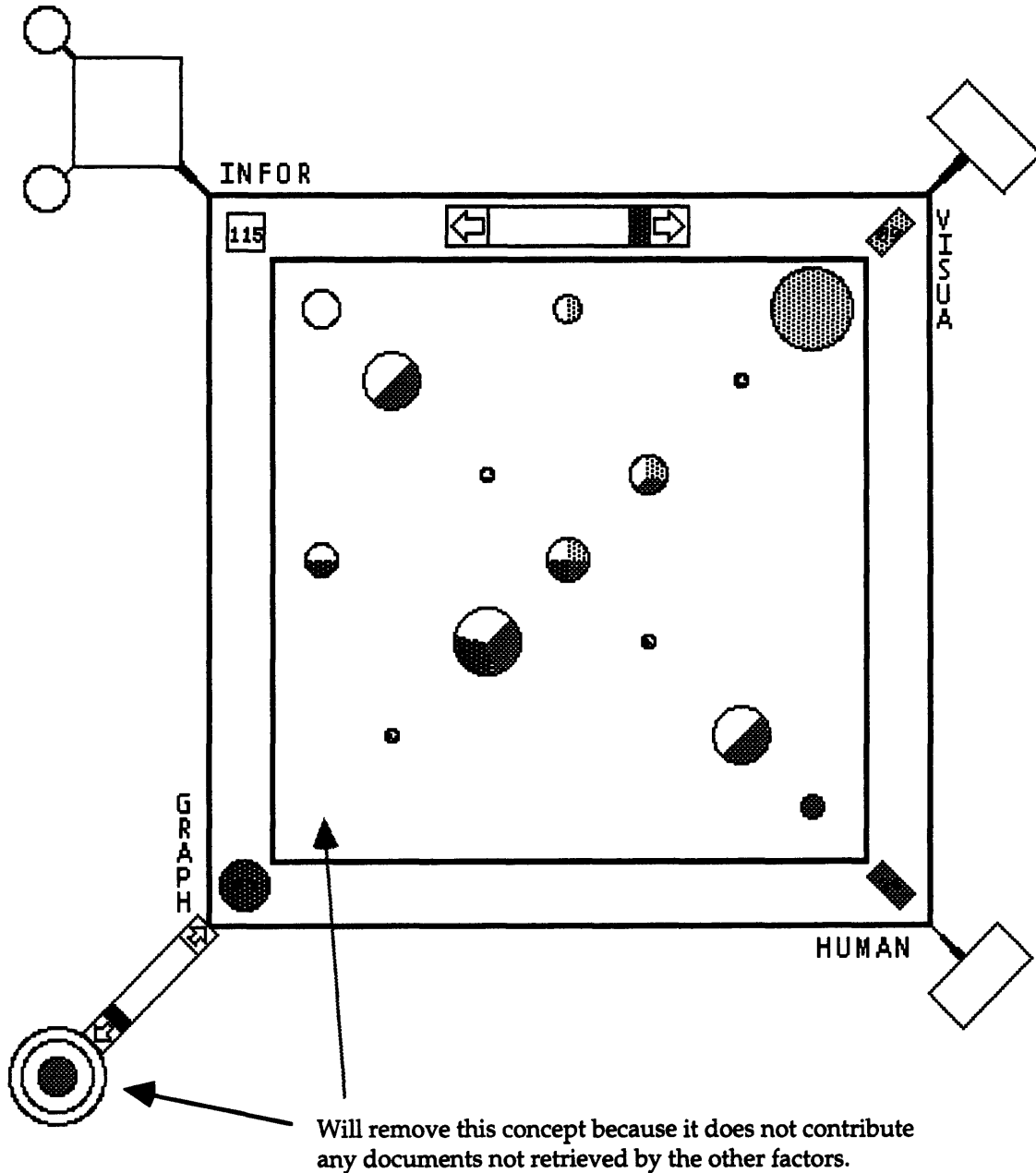
## Reorganizing the query structure using click-drag-drop (before)



**Figure 7.8:** shows the same query structure as in Figure 7.3 and where none of the circular inputs are suppressed. In the next figure we show how the distribution of the documents will change if we promote the circular InfoCrystal, which represents the concept "graphical interfaces" and whose parent is a triangular crystal shown in the bottom right. We will add it as a new concept to the top level InfoCrystal. We perform this change by selecting the circular InfoCrystal icon, dragging and dropping in the border area of the root node InfoCrystal.

We elect to perform this change of the query structure because we want to find out how many of the retrieved documents are only retrieved by the concept "graphical interfaces". We also want to see how the distribution of the documents changes if we consider an additional concept at the root level.

### Reorganizing the query structure using click-drag-drop (1st move)



**Figure 7.9:** shows how the distribution of the documents has changed after we have promoted the concept "graphical interfaces" and have added it as new input to the root-node InfoCrystal. We can observe that there are no documents that are only retrieved by the concept "graphical interfaces". Hence, we will drop this concept by selecting it and applying the cut operation.



### Reorganizing the query structure using click-drag-drop (2nd move)

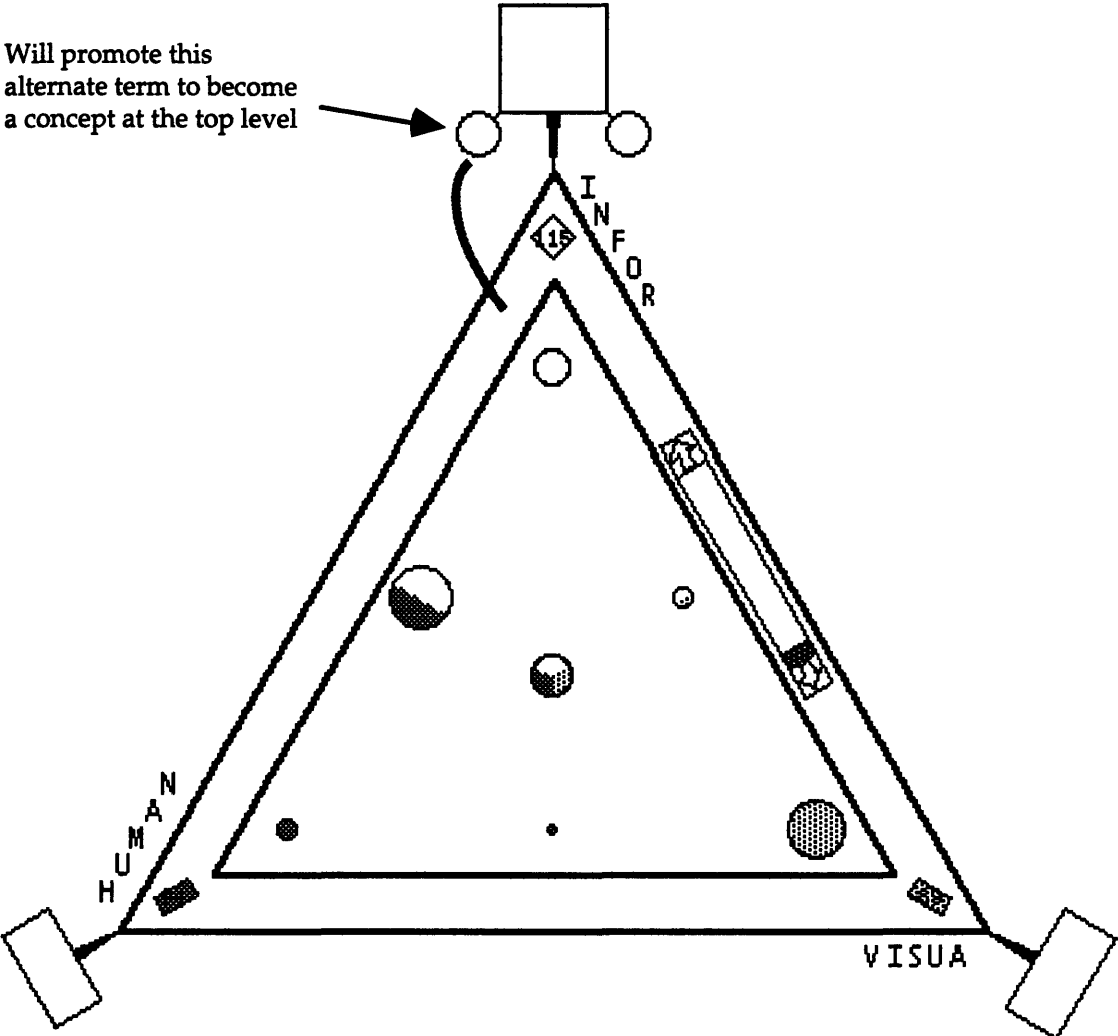
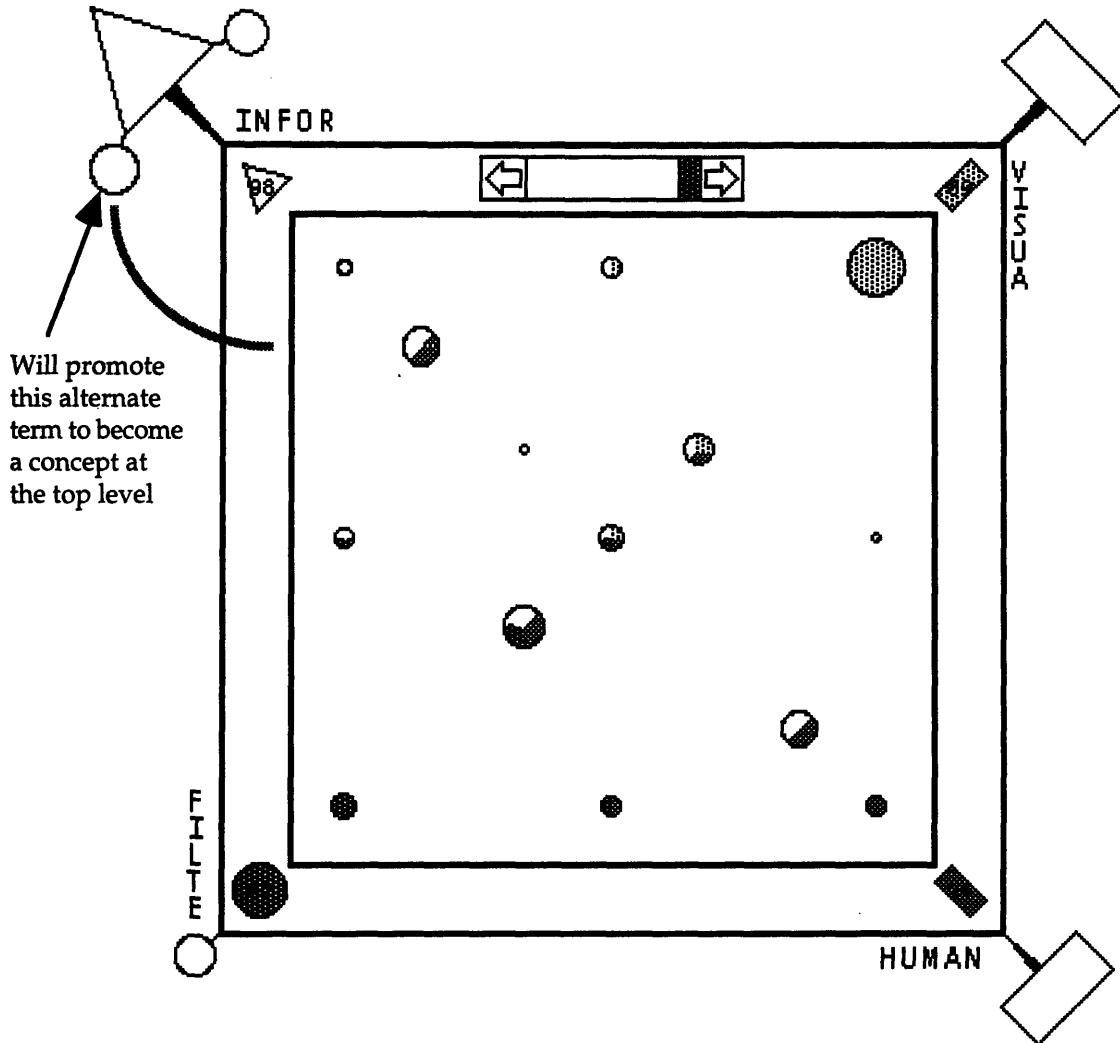


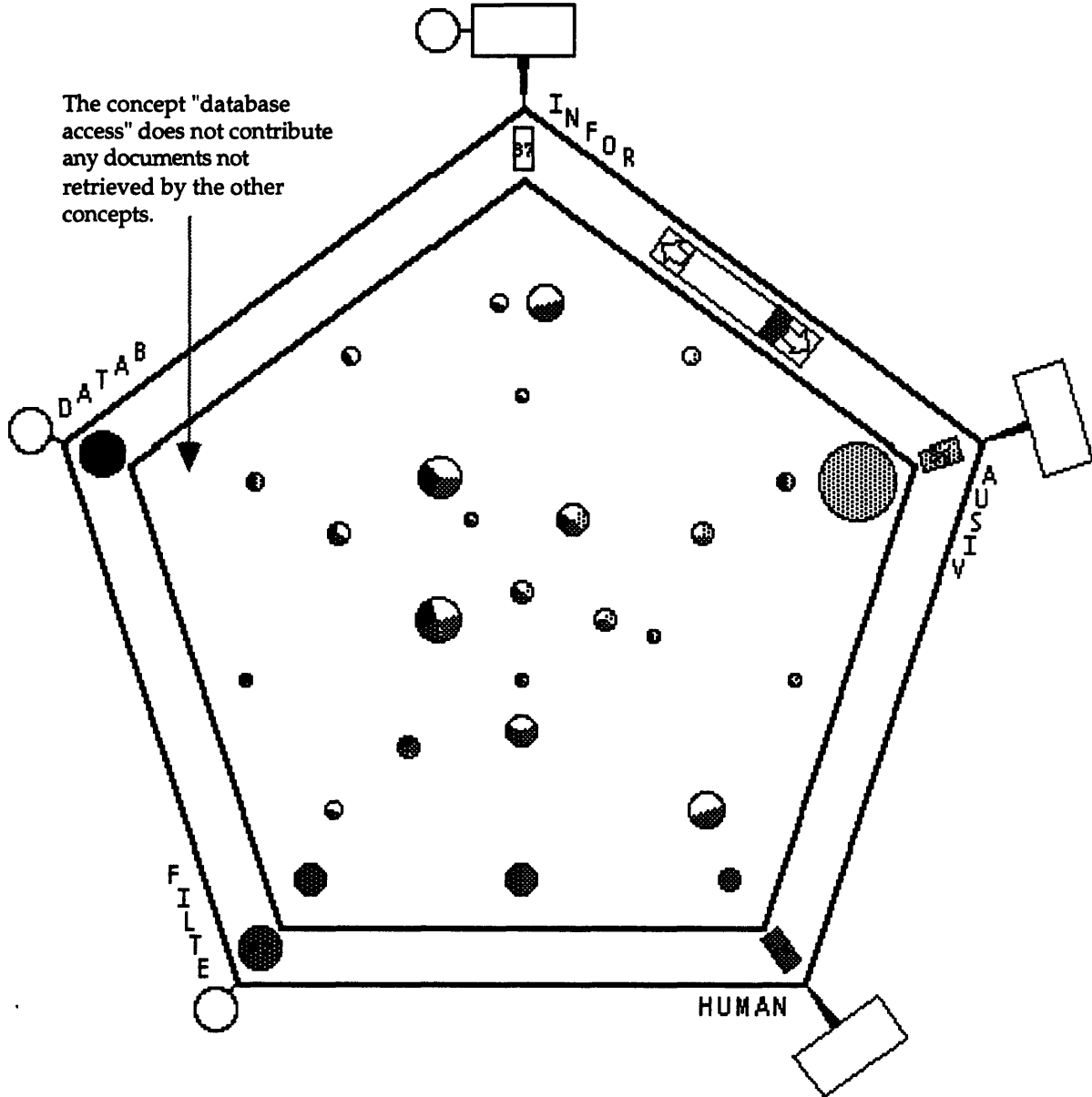
Figure 7.10: shows how the distribution of the documents has consolidated after we have dropped the concept "graphical interfaces" as an input concept. Next we will promote the alternate term "filtering" and add it as a new concept to the top level InfoCrystal. Again we perform this change because we want to find out how many of the retrieved documents are only retrieved by the concept "filtering". We also want to see how the distribution changes if we consider an additional concept at the root level.

### Reorganizing the query structure using click-drag-drop (3rd move)



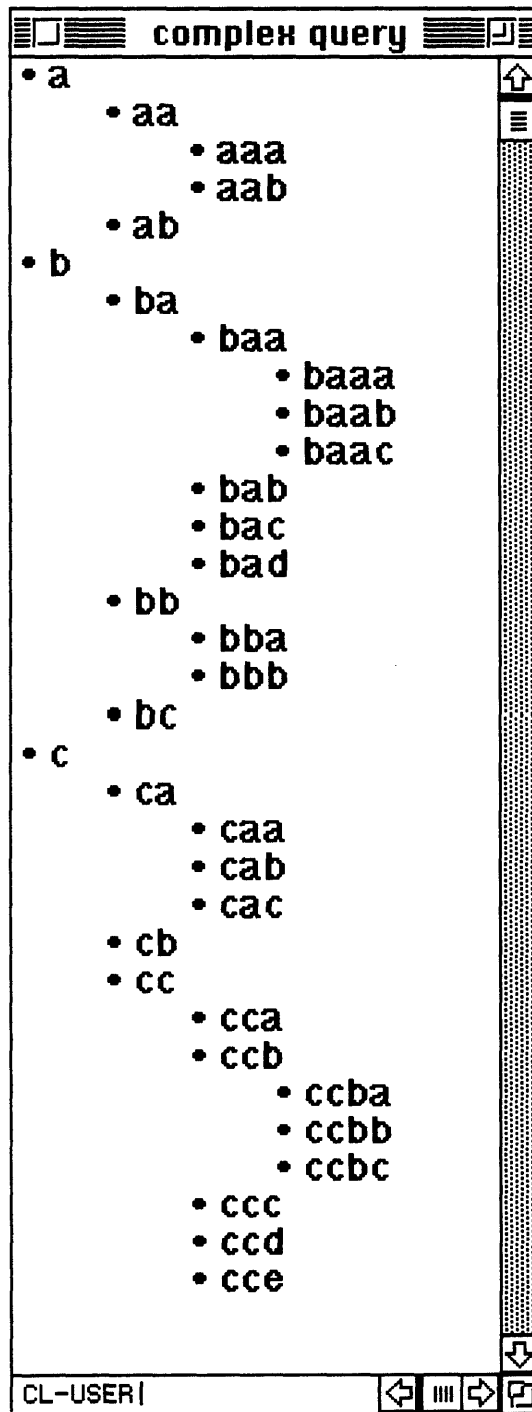
**Figure 7.11:** shows how the distribution of the documents has changed after we have promoted and added the concept "filtering" to the root node InfoCrystal. We can observe that the concept "filtering" retrieves documents that are not retrieved by any of the other input concepts. Finally we will promote the alternate term "database access" and add it as a new concept to the top level InfoCrystal.

### Reorganizing the query structure using click-drag-drop (4th move)



**Figure 7.12:** shows how the distribution of the documents has changed after we have promoted and added the concept "database access" to the root node InfoCrystal. We can observe that are no documents that are only retrieved by this concept, and therefore we could drop it without losing any information.

## Complex Query Structure



**Figure 7.13:** displays the outline for a deeply nested InfoCrystal query structure. We present this example to demonstrate that the InfoCrystal software can be used to formulate arbitrarily complex queries. In the subsequent figure we show one way that this basic outline can be visualized, where not all the InfoCrystals are shown in full detail.

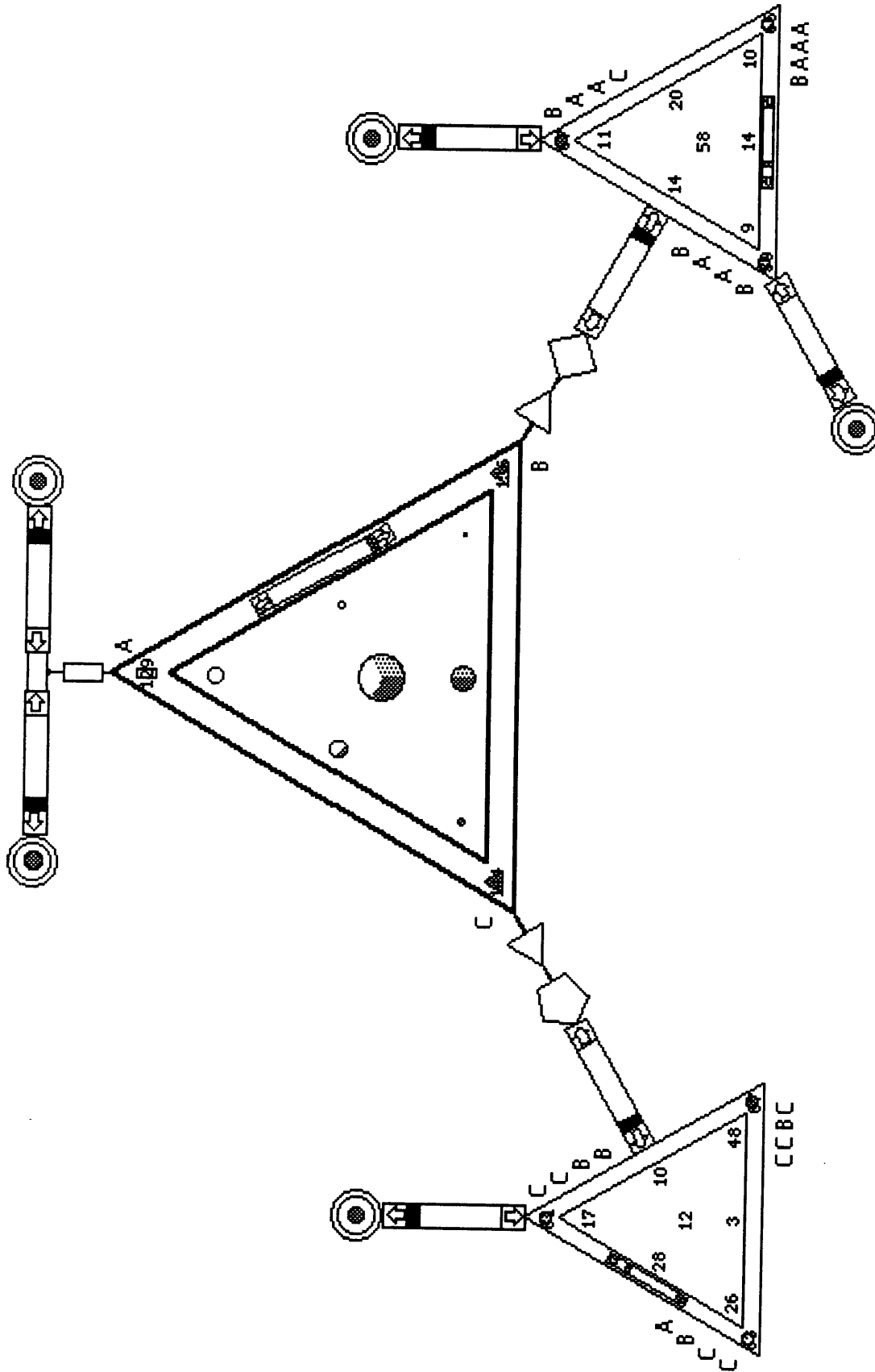


Figure 7.14: visualizes the query structure shown in Figure 7.13. Users can easily view the interior of a crystal only shown as an outline by double-clicking it and this crystal and its children will be redrawn appropriately. This figure demonstrates the spreadsheet quality of the InfoCrystal, where users can make changes deep in the query structure and observe how these changes are propagated and affect the higher levels.

### **7.3 How to Drop an Input from an InfoCrystal ?**

It is common that we have to initially experiment to find the most appropriate search concepts. There will be concepts that we will have to drop, because they do retrieve no or very few many documents that are not also retrieved by the other input concepts. There is also the possibility that too much information is retrieved, and users have to decide which concept(s) to drop. The InfoCrystal can be used to anticipate the effect of dropping a concept from a query. Each of the icons of rank one, i.e., the circular interior icon, represents the information that is not retrieved by any of the other concepts used to define the InfoCrystal. Depending on their specific needs, user can choose to drop the concept whose circular interior icon has a specific value or has the highest or lowest value of all the icons of rank one.

Users can drop a concept in the following two ways: First, they can just deselect the circular icon associated with the concept in question. This leaves open the possibility that users can easily change their mind at a later stage by selecting the circular icon again to include the information associated with it in the output of the InfoCrystal. There is also the possibility that they wish to suppress a concept that is in turn defined by several other concepts. In this case they just deselect all the interior icons of the InfoCrystal that represents the concept in question. Second, users can drop a concept by eliminating it as one the inputs to the InfoCrystal. This is an irreversible action, but it has the advantage that it reduces the complexity of the InfoCrystal, because it decreases its dimensionality and therefore it reduces the number of relationships made explicit simultaneously. In the current implementation users can eliminate a concept by clicking on the InfoCrystal that represents it and applying the familiar Macintosh cut command, Command-X.

### **7.4 How to Add an Input to an InfoCrystal ?**

There will be occasions where users want to add a further concept to an InfoCrystal, because they have discovered a further relevant concept in their exploration so far, or they want to move a concept from one part of the InfoCrystal query structure to another location. In former case users can add a further concept to the outline of the existing query by using the outliner tool. In the latter case users can make use of the fact that the current implementation of the InfoCrystal supports click, drag and drop operations.

---

Users can add a new factor or concept to an existing InfoCrystal, called the receiving InfoCrystal, by clicking on an InfoCrystal that is not a direct input to the receiving InfoCrystal, and dragging it to and releasing it over the receiving InfoCrystal. This will have the effect of adding the selected crystal as a further input to the receiving InfoCrystal. Hence, users can modify in a visual way the structure of a query by moving its members into new positions by selecting, dragging and dropping them in the desired location. By rearranging the structure of the query hierarchy users, decrease the complexity for the InfoCrystal that loses an input, and they increase it for the InfoCrystal that receives a new input.

### **7.5 How to Update the Selection Pattern in a Modified InfoCrystal ?**

If we modify an InfoCrystal by adding or removing one of inputs, then the question arises which of the interior icons in the modified InfoCrystal to select. If we add a further input, called  $I(\text{add})$ , to an InfoCrystal, then an interior icon representing the relationship  $R$  in the unmodified InfoCrystal will be split in two and will be represented by two interior icons in the modified InfoCrystal. These two interior icons represent the relationships  $(R \text{ and } I(\text{add}))$  and  $(R \text{ and } (\text{not } I(\text{add})))$ , respectively, and they will inherit the selection status of the interior icon satisfying  $R$  in the unmodified InfoCrystal. However, we can not infer the selection status of the icon with rank one and that satisfies only the criteria represented by the new input, unless the complement of the unmodified InfoCrystal has a selection status assigned to it. In this case we can elect to select this icon with rank one as a default.

If we remove input, called  $I(\text{remove})$ , from an InfoCrystal, then the situation is more complicated. The interior icons, which satisfy the same criteria and differ only with respect to the criterion for  $I(\text{remove})$ , can be paired. These pairs of interior icons will be represented by a single icon in the modified InfoCrystal, which will inherit the same selection status as its corresponding pair of interior icons in the unmodified InfoCrystal, provided these two icons share the same selection status. However, if these two icons do not share the same selection status, then we can not infer the selection status of the corresponding interior icon in the modified InfoCrystal.

---





# CHAPTER 8

## EXPERIMENTAL EVALUATION

### 8.1 Introduction

In this chapter we will present the experimental design and the results of the user study that was performed to investigate and evaluate a specific aspect of the InfoCrystal. The user study consisted of comparing the standard, text-based Boolean query language with the InfoCrystal, where subjects had to perform a *recognition* and a *generation* task. In each task the subjects were given a series of natural language statements of the information needs. In the recognition task subjects had to recognize for each information need the correct expression from among three possible queries. In the generation task we required subjects to generate a Boolean or InfoCrystal query that captured a given information need.

Although this study did not test all the valuable or promising features of the InfoCrystal, it produced the following useful results: 1) It showed that novice users, who received only a short, fifteen minutes long tutorial, could successfully use the novel InfoCrystal interface. 2) The study showed that the InfoCrystal, even at an early stage of development, performed as well as the familiar Boolean interface, although the study was biased in favor of the Boolean mode (see section 8.4.2.1 for discussion). 3) The user feedback concerning the InfoCrystal interface was very encouraging and it helped to pinpoint possible improvements.

The Boolean query language is the predominant retrieval language and users have difficulties using it effectively [Borgman 1989]. The InfoCrystal is a novel query language and it offers the possibility, among other things, to formulate Boolean queries in a visual way. However, users need to be able to translate their information need into an InfoCrystal by selecting the appropriate interior icons. The InfoCrystal query language raises these specific questions: Are users able to identify easily which particular interior icons

---

contain the information that they are looking for ? Are users able to distinguish correctly between the different interior icons in terms of how they are or are not related to their current information need ? Both the recognition and the generation task address these questions, where the latter task does it in the most direct fashion. An advantage of the InfoCrystal is that it presents all the possible relationships among several concepts at once. Hence, a user gets a complete overview. However, this can also represent a drawback or hindrance, especially to a novice user: So many choices and which ones are relevant to the current information need ? Hence, users have to be able to identify which selection pattern of the interior icons corresponds to their current information needs.

## **8.2 Experimental Design**

The experiment was conducted by having the subjects first view an interactive presentation, created in MacroMind Director, that explained the purpose of the experiment. It described the Boolean query language and it provided an extensive tutorial of how the InfoCrystal could be used as a query language. Appendix 1 describes the tutorial in detail by showing the actual displays and examples used. We had initially a brief introduction to the InfoCrystal that consisted only of an abstract description of general principles without providing concrete examples. However, the preliminary tests showed that this novel query language needed to be explained in more depth. One of the test subjects responded downright hostile to the InfoCrystal because it had not been explained sufficiently with the help of some concrete examples. On average, it took the experimental subjects fifteen minutes to complete the more extensive tutorial. Second, subjects were asked to perform the recognition task, which was conducted in two parts: first a training experiment in which subjects received feedback on their answers, and then the actual experiment without feedback. Third, subjects performed the generation task, which was also conducted in two parts: first a training experiment with feedback and then the actual experiment without feedback. On average, it took the experimental subjects a little more than an hour to complete the tutorial, the recognition and the generation task. For both tasks and in both the training mode and the actual experiment, each subject was presented with each query in both the Boolean and InfoCrystal mode. This

---

fact enabled us to compute the *paired-differences* in performance between the two query languages to reduce the noise and unwanted variability in the collected data. Further, a randomized complete block design was used to minimize learning effects. The performance was measured as follows: 1) A score was computed, which reflects how well a selected or generated query agrees with the correct query. 2) The time it took a subject to chose or generate a query was measured.

In both the training and actual experiment, subjects were presented with a natural language description of the information that they had to retrieve. All the examples were drawn from the domain of finding a film in a video store. This choice of query domain helped to make the experiments more realistic and enjoyable for the subjects. Subjects had to either recognize or generate queries that asked for videos satisfying certain features (e.g., "romance", "adventure", etc.). Table 8.1 shows the queries and the detractors used in the actual experiment for the recognition task. Figures 8.1 and 8.2 show the screen designs used to perform the recognition experiment for the InfoCrystal and Boolean mode, respectively. Table 8.2 displays the queries used in the actual experiment for the generation task. It also shows which ones of the interior icons of an InfoCrystal had to be selected to retrieve the requested information. Figures 8.3 and 8.4 show the screen designs used to perform the generation experiment for the InfoCrystal and Boolean mode, respectively.

We faced several challenges when generating the different information needs. First, we had to create queries that would not lead to Boolean expressions that would be too complicated in terms of the degree of nesting, the need to use brackets and to mix the different Boolean operators. Second, we needed to use a language and sentence structure that was not ambiguous in terms of the intended meaning. However, we did not want to generate natural language statements where it would be straightforward for the subjects to infer the Boolean query with little effort by stripping off some of the fill words. It was not always easy to disguise the structure of the Boolean query in the natural language statement. In the recognition task, we tried to counteract this problem, first, by using a different ordering of the features in the queries than was used in the information need statement; and, second, by varying the natural language statements.

---

In both the recognition and generation task, the training set consisted of six queries, two with two features, two with three, and two with four. The queries with two features were presented first, followed by those with three and finally those with four. The only parameter that was randomized in the training set was the presentation order of the Boolean or InfoCrystal version of a query. However, this randomization was restricted to ensure that the two query languages were presented first in equal numbers within a set of queries that had the same number of features.

Once the subjects had performed the training experiment, they would perform the actual experiment, where we also presented two examples with two, three, and four features, respectively. The order of presentation of these six queries was fully randomized in terms of their rank. However, it was ensured that the Boolean and the InfoCrystal versions of a given query were not presented in successive order. Further, the two query languages were presented first in equal numbers within a set of queries that had the same number of features.

---

<p>1) <b>We are interested in documentaries that are suitable for children.</b>                  Correct: ("children" AND "documentary")                  Detractor: ("children" OR "documentary")                  Detractor: ("documentary" AND (NOT "children"))</p>
<p>2) <b>We are interested in movies that have violent elements or are suitable for children but not both.</b>                  Correct: (("violence" OR "children") AND (NOT ("violence" AND "children")))                  Detractor: ("violence" OR "children" OR (NOT ("violence" AND "children")))                  Detractor: (("violence" AND (NOT "children")) AND (NOT ("children" AND "violence")))</p>
<p>3) <b>We are interested in movies that have suspense, action and no romance.</b>                  Correct: ("action" AND "suspense" AND (NOT "romance"))                  Detractor: ("suspense" OR "action" OR (NOT "romance"))                  Detractor: (("suspense" AND "action") OR (NOT "romance"))</p>
<p>4) <b>We are interested in movies that have just one (but not more than one) of the following three features: suspense, action, and romance.</b>                  Correct: (("suspense" AND (NOT ("action" OR "romance"))) OR ("action" AND (NOT ("suspense" OR "romance"))) OR ("romance" AND (NOT ("suspense" OR "action"))))                  Detractor: (("suspense" AND (NOT ("action" AND "romance"))) OR ("action" AND (NOT ("suspense" AND "romance"))))                  Detractor: (("suspense" OR "action" OR "romance") AND (NOT ("suspense" AND "action" AND "romance")))</p>
<p>5) <b>We are interested in movies that have been directed by Woody Allen, where Diane Keaton does not appear in them and that are comedies and that have a romantic theme.</b>                  Correct: ("Woody Allen" AND "comedy" AND "romance" AND (NOT "Diane Keaton"))                  Detractor: ("Woody Allen" AND "comedy" AND ("romance" OR (NOT "Diane Keaton")))                  Detractor: ("Woody Allen" AND (NOT ("comedy" OR "romance" OR "Diane Keaton")))</p>
<p>6) <b>We are interested in movies that have been directed by Woody Allen and that satisfy at least two of the following requirements: comedy, drama or mystery theme.</b>                  Correct: ("Woody Allen" AND (("comedy" AND "drama") OR ("drama" AND "mystery") OR ("mystery" AND "comedy"))                  Detractor: ("Woody Allen" AND ("comedy" OR "drama" OR "mystery"))                  Detractor: ("Woody Allen" AND (("comedy" OR "drama") AND ("drama" OR "mystery")))</p>

**Table 8.1:** displays the six queries used for the recognition task, where we have two queries with two, three, and four features, respectively.

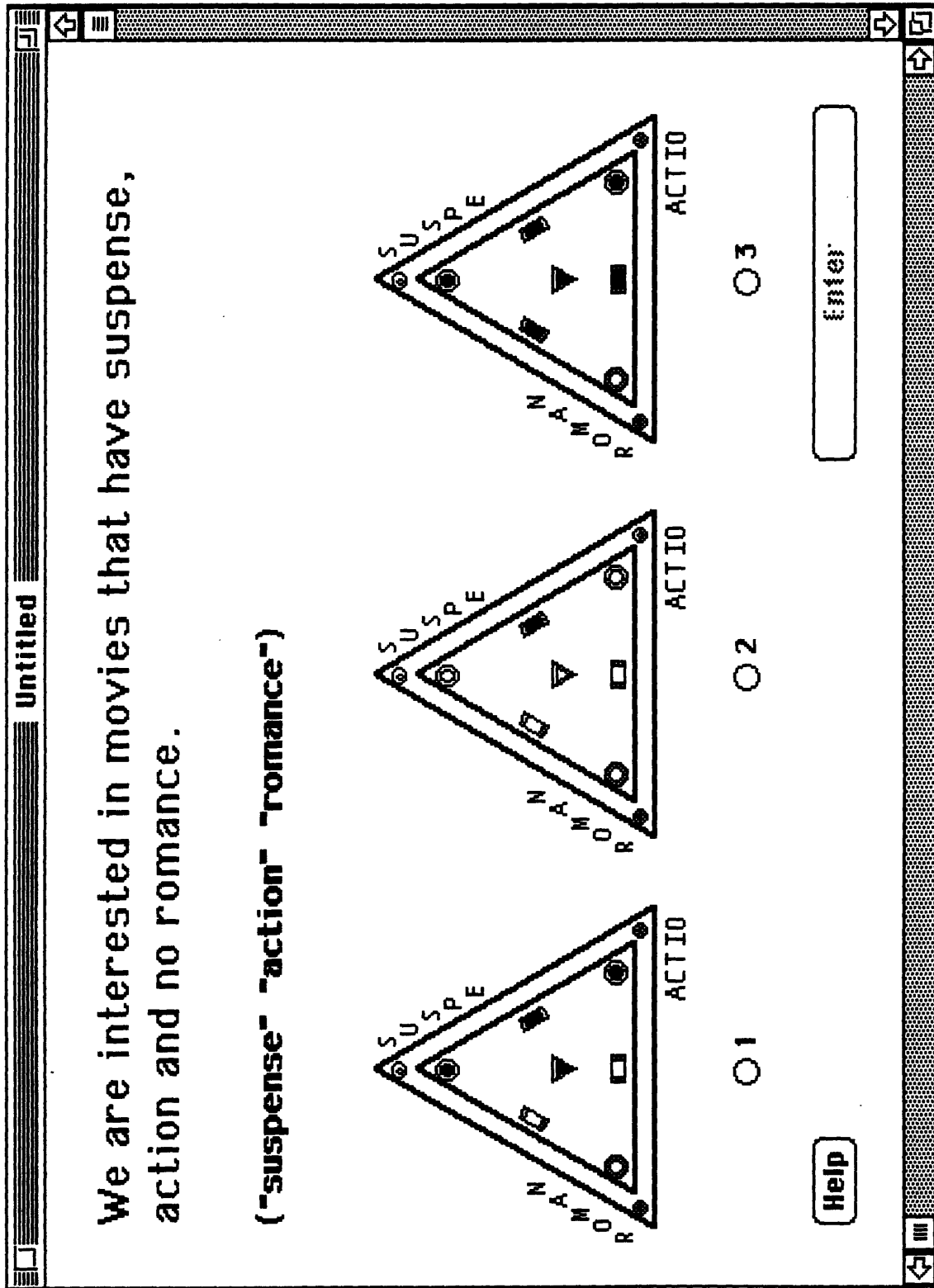


Figure 8.1: shows the recognition task display for the InfoCrystal mode. Subjects had to select the correct choice by clicking on a radio button (for this information need, the second InfoCrystal is the correct choice).

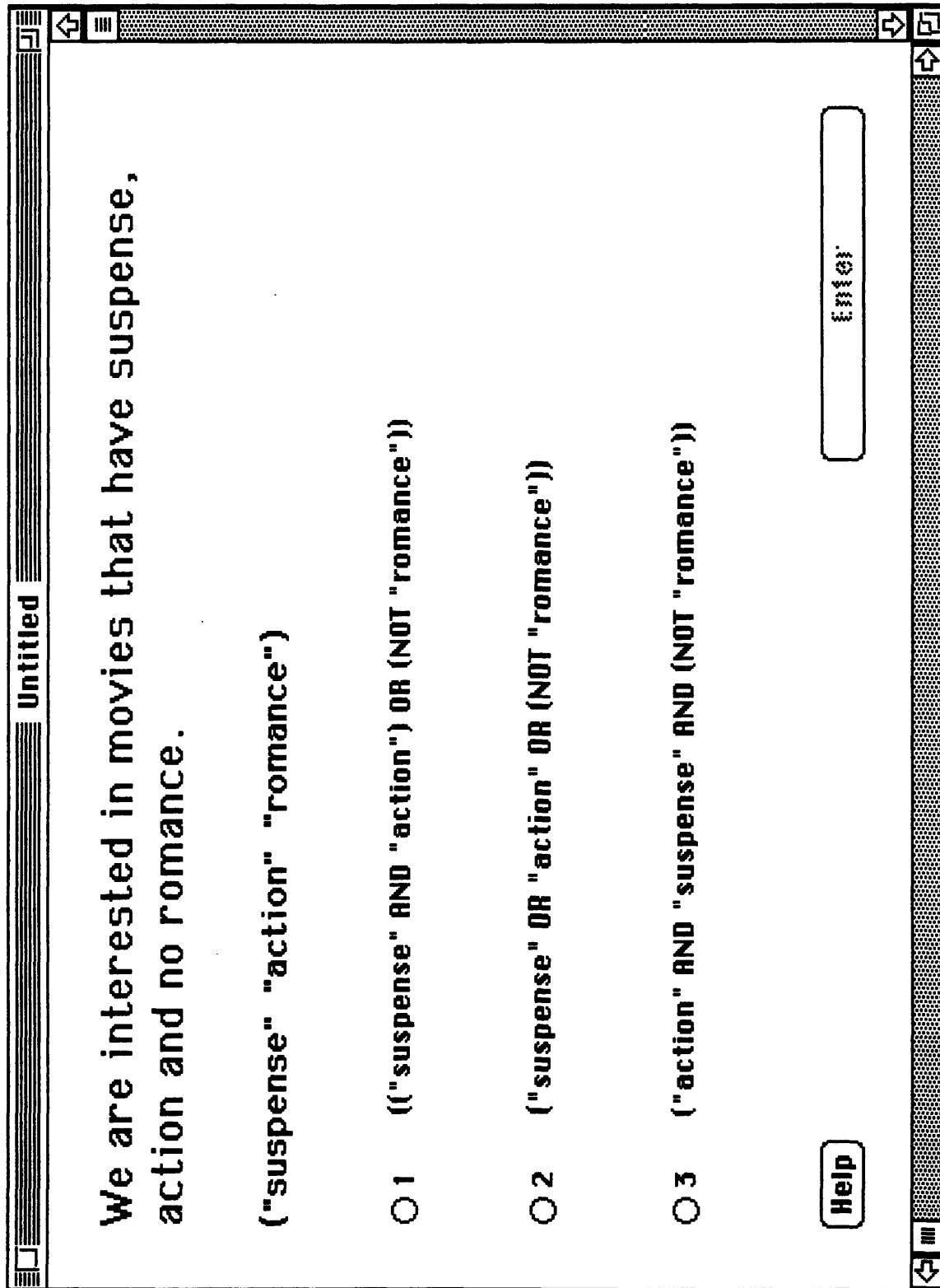


Figure 8.2: shows the recognition task display for the Boolean mode. Subjects had to select the correct choice by clicking on a radio button for this information need, the third query is the correct choice).

<p>1) We are interested in movies that are in a foreign language or that are experimental.</p> <p>Correct: ("foreign" OR "experimental")</p>	
<p>2) We are interested in movies that have been directed by Martin Scorsese and where there is no violence.</p> <p>Correct: ("Martin Scorsese" AND (NOT "violence"))</p>	
<p>3) We are interested in movies that have romance or no action or no violence.</p> <p>Correct: ("romance" OR (NOT "action") OR (NOT "violence"))</p>	
<p>4) We are interested in movies that have exactly two of the following three features: suspense, action, and romance.</p> <p>Correct: (("suspense" AND "action" AND (NOT "romance")) OR ("action" AND "romance" (NOT "suspense")) OR ("romance" AND "suspense" AND (NOT "action")))</p>	
<p>5) We are interested in movies where Al Pacino appears in them or that have not been directed by Coppola and that are romantic, but not violent.</p> <p>Correct: (("Al Pacino" OR (NOT "Coppola")) AND "romance" AND (NOT "violence"))</p>	
<p>6) We are interested in movies that have been directed by Coppola or where Al Pacino appears in them. Further, they should have romance or drama.</p> <p>Correct: (("Al Pacino" OR "Coppola") AND ("romance" OR "drama"))</p>	

**Table 8.2:** displays the six queries used for the **generation task**, where we have two queries with two, three, and four features, respectively. The interior icons shown in black need to be selected.



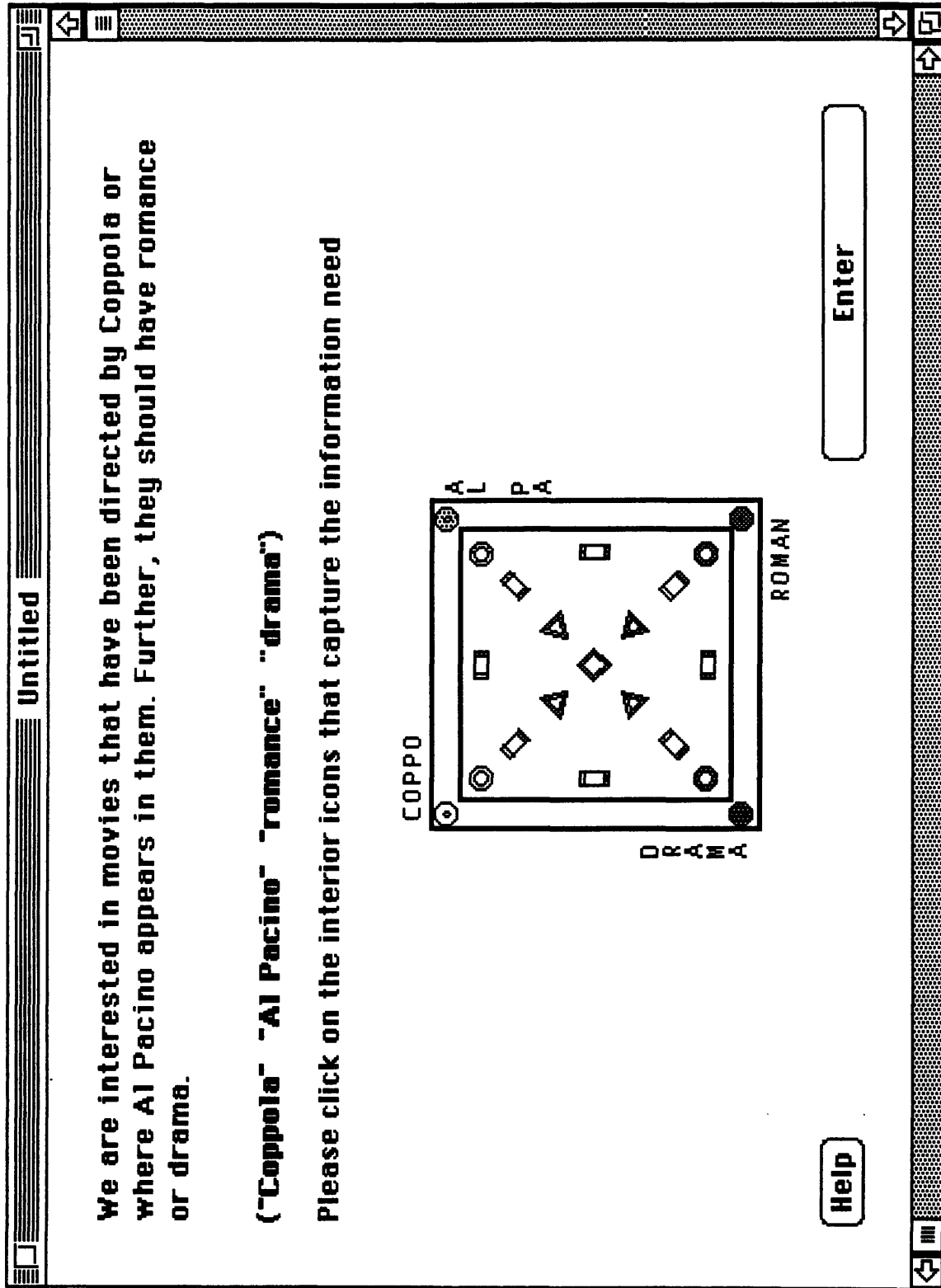


Figure 8.3: shows the generation task display for the InfoCrystal mode. Subjects had to select the appropriate interior icons, where initially none were selected.

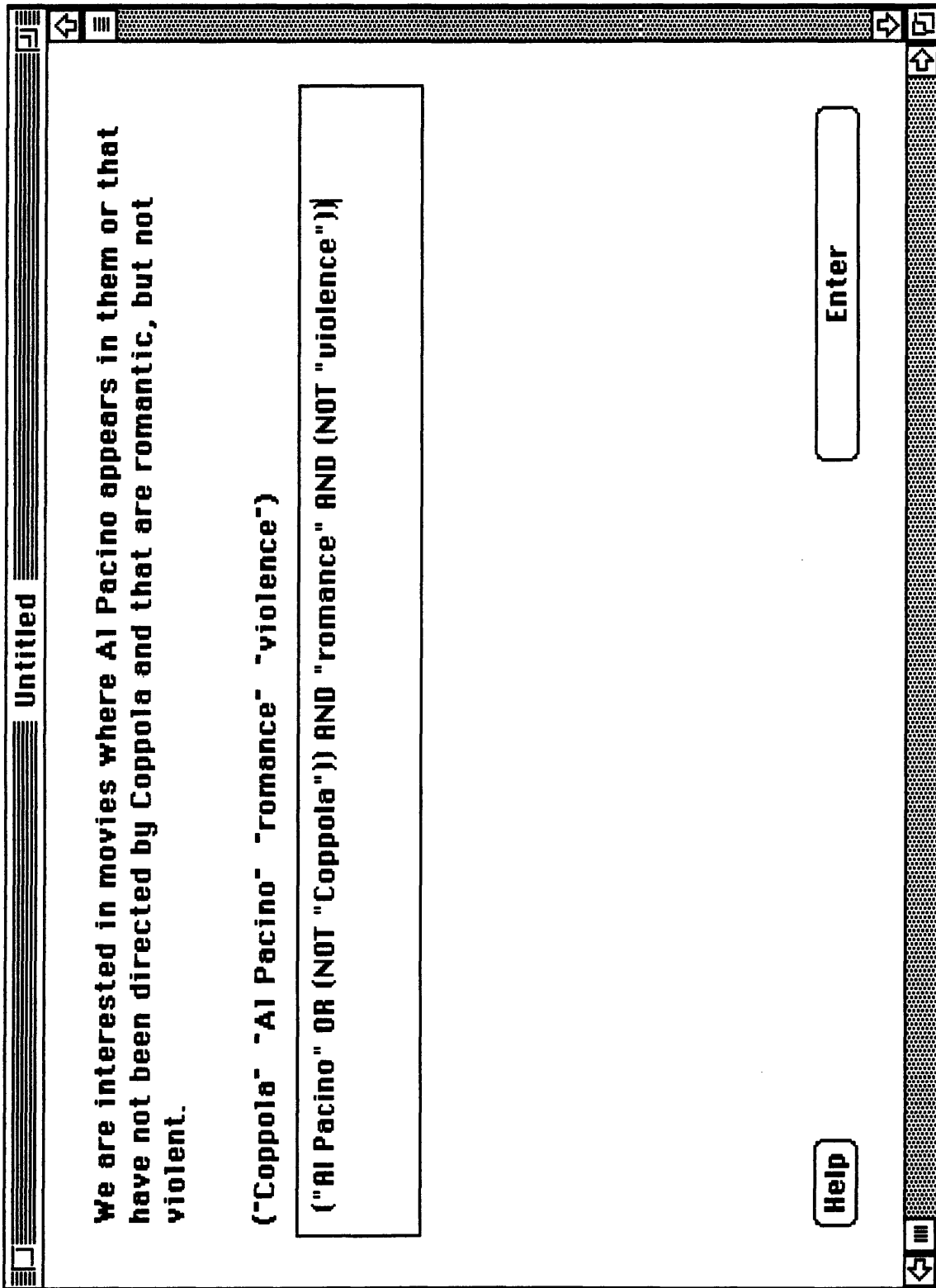


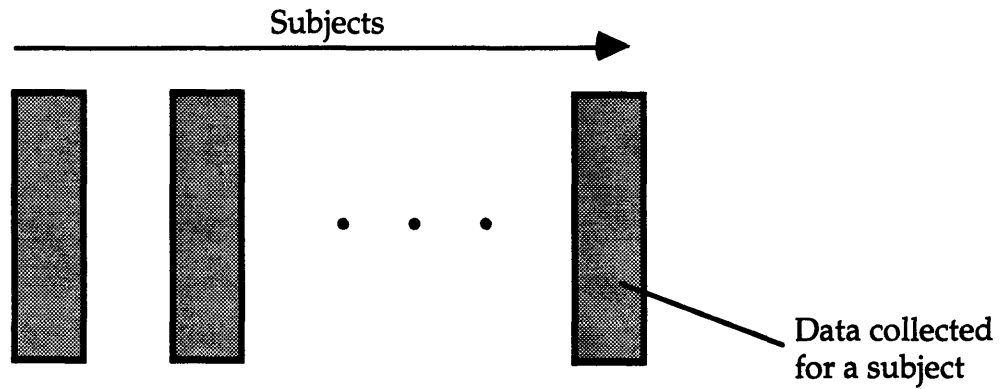
Figure 8.4: shows the generation task display for the Boolean mode. Subjects had to generate the appropriate Boolean expression as shown here.

### 8.3 Experimental Analysis

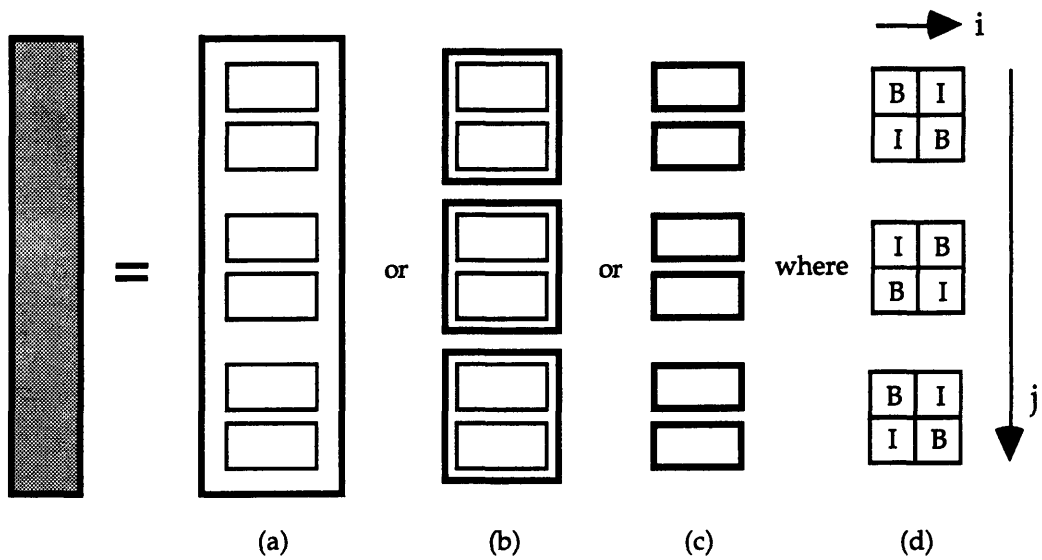
The material describing the statistical designs and formulas used to analyze the data is based on the textbook by Montgomery (1991) that deals with the design and analysis of experiments. We deliberately designed the experiments so that each subject was presented with each query in both the Boolean and InfoCrystal mode. This fact enabled us to compute the *paired-differences* in performance between the two query languages for the same query and for the same subject. Hence, we measured for each subject the relative difference in their performance between the two query languages, and we could use these paired-differences to make a statistical inference. The advantage of the paired comparison design is that we can reduce the noise by being able to focus on the relative performance difference between the two query languages for the same query and the same subject. We thereby increase the homogeneity of the responses and we can better control for the variability among the different subjects in terms of their skills and experience [Montgomery 1991].

Figure 8.5 provides a schematic overview of how the collected data has been analyzed. For each subject, we start out by grouping the scores for all the six queries together and taking their average. Next we only group and take the average the scores for the queries that have the same number of features, and finally we compare the scores for the individual queries. At the coarsest level of analysis we use a **Paired-Difference T-Test** to infer if there is statistically significant difference between the Boolean and the InfoCrystal query language. For the two other ways of grouping and averaging the data, we use an **Analysis of Variance** of the paired-difference scores to make statistical inferences. The main purpose for performing the analysis of variance is to investigate if the number of features used in a query affects the performance. Further, we are also interested to see if there are significant differences in performance between the individual queries.

---



where



where  = InfoCrystal-score minus Boolean-score

$i$  = presentation order ( $1 \leq i \leq 2$ )

$j$  = queries ( $1 \leq j \leq 6$ )

**Figure 8.5:** shows a schematic overview of how the collected data has been grouped and analyzed:  
 (a) We take the average of the paired-differences for the six queries and use the T-test to test the hypothesis whether the mean is equal to zero.  
 (b) We take the average of the paired-difference scores for each pair of queries that have the same number of features. We perform an analysis of variance (ANOVA) of a single factor at three levels.  
 (c) We perform an analysis of variance (ANOVA) of a single factor that is equal to the individual queries, and we conduct the ANOVA at six levels.  
 (d) We compute the paired-differences by using a 2x2 Latin-Square for the pair of queries that have the same number of features to ensure that both query languages are presented first the same number of times.

### 8.3.1 Paired-Difference T-Test

We first calculate the average of the paired-difference scores for all six queries for each subject. Next we compute the mean of these average and the estimated standard deviation of these averages for all the ten subjects. We then calculate the T-value by dividing the mean by its associated estimated standard deviation. We apply the one-sided T-test because we are interested in the probability that the observed superior performance of a query language could be due to chance. We set the T-level at 5% or 1% and the degree of freedom of the T-distribution is  $10 - 1 = 9$ .

For the paired-difference T-test we use a t-distribution as the test statistic:

$$t_0 = \frac{\bar{d}}{S_d/\sqrt{n}} \text{ with } (b - 1) \text{ degrees of freedom}$$

where  $b$  is equal to the number of subjects (in our case ten subjects), and

$$\bar{d} = \frac{1}{b} \sum_{j=1}^b d_j$$

$$d_j = \frac{1}{a} \sum_{i=1}^a \left( score_{InfoCrystal_i} - score_{Boolean_i} \right)$$

where  $a$  is equal to the number of queries (in our case six queries), and

$$S_d = \sqrt{\frac{\sum_{j=1}^b (d_j - \bar{d})^2}{b - 1}}$$

where  $S_d$  is equal to the estimate of the standard deviation.

### 8.3.2 Analysis of Variance

The Analysis of Variance (ANOVA) uses a randomized block design to test if the difference in performance between the two query languages is affected in a statistically significant way by the different treatments. In our case the treatments either correspond to the number of features used in a query or to the individual queries. The ANOVA requires that the treatments are

completely randomized within each subject block. Hence, we have a randomization restriction in the form of the subjects. This restriction is important, because it implies that it is not possible for us to test the difference in performance between the different subjects. Furthermore, we can not study the interaction effects between the query languages and the subjects, because we only have one data point for each query and each subject.

The statistical model for the Randomized Complete Block Design is

$$d_{ij} = \mu + \alpha_i + \beta_j + \varepsilon_{ij} \begin{cases} i = 1, \dots, a \\ j = 1, \dots, b \end{cases}$$

where  $\mu$  is the overall mean,  $\alpha_i$  is the effect of the  $i$ th treatment,  $\beta_j$  is the effect in the  $j$ th block, and  $\varepsilon_{ij}$  is the random error term that is assumed to be normally distributed with mean zero and standard deviation  $\sigma$ . Each block corresponds to an individual subject, and a treatment either corresponds to a grouping of queries with the same number of features or to the individual queries, respectively. Hence,  $a$  can take the values 3 or 6, and  $b$  is equal to the number of experimental subjects, which in our case is equal to ten.

The treatment and block effects are defined as deviation from the overall mean so that

$$\sum_{1 \leq i \leq a} \alpha_i = 0 \quad \text{and} \quad \sum_{1 \leq j \leq b} \beta_j = 0$$

which enables us to partition the total sum of squares in the following way:

$$SS_{Total} = SS_{Treatments} + SS_{Blocks} + SS_{Error}$$

$$\sum_{i=1}^a \sum_{j=1}^b d_{ij}^2 - \frac{d_{..}^2}{N} = \sum_{i=1}^a \frac{d_{i.}^2}{b} - \frac{d_{..}^2}{N} + \sum_{j=1}^b \frac{d_{.j}^2}{a} - \frac{d_{..}^2}{N} + SS_{Error}$$

where

$$d_{..} = \sum_{i=1}^a \sum_{j=1}^b d_{ij} \quad \text{and} \quad d_{i.} = \sum_{j=1}^b d_{ij} \quad \text{and} \quad d_{.j} = \sum_{i=1}^a d_{ij} .$$

Source of Variation	Sum of Squares	Degrees of Freedom	Mean Square	F-value
Treatments	$\sum_{i=1}^a \frac{d_{i.}^2}{b} - \frac{d_{..}^2}{N}$	a - 1	$\frac{SS_{Treatments}}{a - 1}$	$\frac{MS_{Treatments}}{MS_{Error}}$
Blocks	$\sum_{j=1}^b \frac{d_{.j}^2}{a} - \frac{d_{..}^2}{N}$	b - 1	$\frac{SS_{Blocks}}{b - 1}$	Can not be computed because we have a randomization restriction in the form of the subject blocks
Error	by subtraction	(a - 1)(b - 1)	$\frac{SS_{Error}}{(a - 1) \cdot (b - 1)}$	
Total	$\sum_{i=1}^a \sum_{j=1}^b d_{ij}^2 - \frac{d_{..}^2}{N}$	N - 1		

**Table 8.3:** shows the analysis of variance table, where the total sum of squares is partitioned into the sums of squares for the treatments and block effects as well as the error term. These sums and their associated degrees of freedom are used to compute the mean square values, which in turn can be used to compute the F-value that tells us if the differences between the treatments are statistically significant.

This partitioning of the total sum of squares is used to construct the ANOVA table as shown in Table 8.3. We can compute the degree of variability that is due to the treatments and the blocks, respectively. The remaining variability is attributed to the error term. We can compute the mean square for the different sum of squares by using the degrees of freedom associated with these different sums. Finally, we can calculate the F-value for the treatments to test if the difference in performance between the different treatments is statistically significant.

If we would like to investigate multiple factors and their interactions, then we could use a *factorial design*. The simplest type of a factorial design involves only two factors. There are *a* levels of factor A and *b* levels for factor B, and these are arranged in a factorial design: each replicate of the experiment contains all *ab* treatment combinations. The model for a two-factor factorial design with one replicate looks exactly like the randomized complete block design. However, the experimental designs that lead to the

randomized block and factorial models are very different. In the factorial model, *all ab* runs have been performed in random order, whereas in the randomized block model, randomization occurs only *within* the block. Hence, it is not appropriate to analyze our collected data as if it been generated by a factorial design.



## 8.4 Analysis of the Experimental Results

In this section we present and analyze the results for the recognition and generation tasks, where ten subjects participated in the user study. The sample included four women and six men, where their ages ranged from the early twenties to the middle forties. All the subjects had at least a college education and they had been exposed to the Boolean retrieval language during their education or professional life. In chapter 9 we present a table that summarizes the feedback we received from the subjects and it also contains more information about their background. Although it does not necessarily constitute a representative sample of ordinary users, it is sufficiently diverse to serve as an initial sample to begin to study the effectiveness of the InfoCrystal as a Boolean query language.

### 8.4.1 Results for the Recognition Task

In the recognition task subjects had to select the correct Boolean or InfoCrystal query from among three possible choices. We computed a *score* for each query that could take the following categorical values: 1) If a subject chose the correct query or the wrong query for both languages, then the score was set equal to zero. 2) If a subject chose the correct query only when viewing it in the InfoCrystal mode, then the score was equal to plus one; whereas in the opposite case (correct in the Boolean mode and incorrect in InfoCrystal) it was set equal to minus one. Further, we recorded for each query the amount of time it took a subject to make a final selection. We then computed the difference between the recorded time when the query was presented in the InfoCrystal mode minus the time for the Boolean mode. Hence, a negative difference value implied that it took a subject less time to make a selection using the InfoCrystal than using the Boolean interface.

The main conclusion that we can draw from the analysis of the results for the recognition task is that there is no statistically significant difference between the two query languages that can be inferred based on the T-test of either the scores or the time measurements. Further, the analysis of variance of the scores does not show that there is a significant difference that could be attributed to the number of features used in a query or to the individual queries themselves. The only statistically significant difference exists for the






---

analysis of variance of the time measurements, both in terms of the number of features used in a query and the individual queries, respectively.

#### 8.4.1.1 Categorical Paired-Difference Scores

Table 8.4 displays the scores and it clearly shows that the subjects predominately selected the correct query for both query languages with very few exceptions. Hence, it should come as no surprise that the T-test of the average

	1	2	3	4	5	6	7	8	9	10
(A and B)	+0	+0	-0	+0	+0	+0	+0	+0	+0	+0
(A xor B)	+0	-1	+0	+0	1	+0	+0	+0	+0	+0
(A and B and (not C))	+0	+0	+0	+0	1	+0	+0	+0	+0	+0
Exactly 1 out of 3	+0	+0	+0	+0	+0	+0	+0	+0	1	+0
(A and B and C and (not D))	+0	+0	-1	+0	+0	+0	+0	+0	+0	+0
(A and at least 2 out of 3 remaining)	+0	+0	-1	+0	+0	+0	+0	+0	-1	+0
Total	0	-1	-2	0	2	0	0	0	0	0
Average	0	-0.17	-0.33	0	0.33	0	0	0	0	0

Categorical Scores	1	2	3	4	5	6	7	8	9	10
<b>(A and B)</b>										
<b>(A xor B)</b>										
<b>(A and B and (not C))</b>										
<b>1 out of 3</b>										
<b>(A and B and C and (not D))</b>										
<b>(A and at least 2 out of rest)</b>										

**Table 8.4:** shows the scores for the recognition task, which have been calculated as follows: 1) If a subject chose the correct query or the wrong query for both languages, then the score was set equal to +0 or -0, and this is visualized in the bottom table using a gray or stripped pattern, respectively. 2) If a subject only chose the correct query when viewing it in the InfoCrystal mode, then the score was equal to 1 and was visualized using light gray; whereas in the opposite case it was set equal to -1 and visualized using black. The leftmost column displays the Boolean structure of the queries used in the experiment. The query (A xor B) uses the exclusive OR operator to arrive a shorter expression for this table, although the actual Boolean is more complicated. Finally, we have grouped the queries with the same number of features by enclosing them by a thick black border.

Paired-Difference T-test			-1.833	-2.821
Mean	Est.St.Devi.	to	t(0.05,9)	t(0.01,9)
-0.02	0.17	-0.32	no	no

**Table 8.5:** shows the mean and the estimated standard deviation of the averages of the scores for the recognition task. The resulting t-value of -0.32 is not significant for a t-distribution with 9 degrees of freedom at the 1% or 5% level, where the corresponding values of the t-distribution are equal to -1.833 and -2.821, respectively, at those levels.

ANOVA (treatments = number of features used in a query)					4.41	6.01
	S.o.S.	D.o.F.	M.S.	Fo	(2,18;0.05)	(2,18;0.01)
Treatments	0.32	2	0.16	2.41	no	no
Blocks	0.74	9	0.08			
Error	1.18	18	0.07			
Total	2.24	29				

ANOVA (treatments = individual queries)					2.43	3.5
	S.o.S.	D.o.F.	M.S.	Fo	(5,45;0.05)	(5,45;0.01)
Treatments	0.68	5	0.14	1.28	no	no
Blocks	1.48	9	0.16			
Error	4.82	45	0.11			
Total	6.98	59				

**Table 8.6:** shows the analysis of variance tables of the scores for the recognition task, where the treatments are either the pairs of queries with the same number of features or the individual queries. The resulting F-values are not significant at the 1% or 5% level for either ANOVA.

of all six scores is not significant at either the 1% or the 5% level (see Table 8.5). Similarly, the analysis of variance of the scores does not reveal any statistically significant differences between the treatments (see Table 8.6).

8.4.1.2 Time Measurements

Next we examine the time it took subjects to make a final selection from among the three choices. Table 8.7 (a) shows the difference between the time measurements for the InfoCrystal and the time for Boolean version of a query. The table (c) shows the time measurements for the InfoCrystal. The table (d) shows the percentile difference in the time measurements between the two query languages, where we divide the time difference between the query modes by the time for the Boolean mode.

(a) Time Differences between the InfoCrystal and the Boolean mode

	1	2	3	4	5	6	7	8	9	10		
(A and B)	-1	-12	14	7	-6	10	9	7	22	15	65	-66
(A xor B)	-13	-44	-8	0	1	-7	-23	-14	-12	-11	-131	
(A and B and (not C))	2	5	27	29	16	4	4	4	-14	3	80	-176
Exactly 1 out of 3	-18	-39	-10	-54	-20	3	-24	-30	-39	-25	-256	
(A and B and C and (not D))	6	-27	29	-1	32	15	-14	-9	9	12	52	314
(A and at least 2 out of 3 remaining)	17	52	40	50	-10	5	42	-3	46	23	262	
Total	-7	-65	92	31	13	30	-6	-45	12	17		
Average	-1.17	-10.83	15.33	5.17	2.17	5.00	-1.00	-7.50	2.00	2.83		

(b) Time Differences between the two modes represented visually

Time	1	2	3	4	5	6	7	8	9	10		
(A and B)	Light Gray	Light Gray	Black	Black	Light Gray	Black	Black	Black	Black	Black	65	-66
(A xor B)	Light Gray	Light Gray	Light Gray	Light Gray	Black	Light Gray	Light Gray	Light Gray	Light Gray	Light Gray	-131	
(A and B and (not C))	Black	Black	Black	Black	Black	Black	Black	Light Gray	Black	Black	80	-176
1 out of 3	Light Gray	Light Gray	Light Gray	Light Gray	Light Gray	Black	Light Gray	Light Gray	Light Gray	Light Gray	-256	
(A and B and C and (not D))	Black	Light Gray	Black	Light Gray	Black	Black	Light Gray	Light Gray	Black	Black	52	314
(A and at least 2 out of rest)	Black	Black	Black	Black	Light Gray	Black	Black	Light Gray	Black	Black	262	

Table 8.7: (a) shows the difference in the time measurements, measured in seconds, between the InfoCrystal and the Boolean query language for each query; (b) displays the time differences in a graphical way, where light gray indicates that the InfoCrystal was faster, and black that the Boolean interface took less time.

(c) Times for the InfoCrystal

	1	2	3	4	5	6	7	8	9	10
(A and B)	9	7	26	17	19	19	17	16	31	24
(A xor B)	10	11	11	27	29	12	15	8	24	19
(A and B and (not C))	15	16	37	42	44	30	22	19	36	16
Exactly 1 out of 3	10	18	13	10	17	45	13	15	27	14
(A and B and C and (not D))	26	22	55	29	74	41	24	25	46	34
(A and at least 2 out of 3 remaining)	40	74	75	63	35	28	79	35	102	43

(d) Percentile Difference between the two query languages

	1	2	3	4	5	6	7	8	9	10
(A and B)	-10%	-63%	117%	70%	-24%	111%	113%	78%	244%	167%
(A xor B)	-57%	-80%	-42%	0%	4%	-37%	-61%	-64%	-33%	-37%
(A and B and (not C))	15%	45%	270%	223%	57%	15%	22%	27%	-28%	23%
Exactly 1 out of 3	-64%	-68%	-43%	-84%	-54%	7%	-65%	-67%	-59%	-64%
(A and B and C and (not D))	30%	-55%	112%	-3%	76%	58%	-37%	-26%	24%	55%
(A and at least 2 out of 3 remaining)	74%	236%	114%	385%	-22%	22%	114%	-8%	82%	115%

Table 8.7 (cont.): (c) displays the time measurements for the InfoCrystal; (d) shows the percentile difference between the two query languages by dividing the entries in table (a) by the time measurements for the Boolean mode of a query.

Table 8.8 shows the outcome of performing the T-test on the average of all six time differences between the two query languages and it is not significant at either the 1% or 5% level.

Paired-Difference T-test			1.833	2.821
Mean	Est.St.Devi.	to	t(0.05,9)	t(0.01,9)
1.20	7.19	0.53	no	no

Table 8.8: shows the mean and the estimated standard deviation of the averages of the time difference for the recognition task. The resulting t-value of 0.53 is not significant for a t-distribution with 9 degrees of freedom at either the 1% or 5% level.

ANOVA (treatments = number of features used in a query)					4.41	6.01
	S.o.S.	D.o.F.	M.S.	Fo	(2,18;0.05)	(2,18;0.01)
Treatments	3305.00	2	1652.50	19.91	yes	yes
Blocks	1395.30	9	155.03			
Error	1494.00	18	83.00			
Total	6194.30	29				

ANOVA (treatments = individual queries)					2.43	3.5
	S.o.S.	D.o.F.	M.S.	Fo	(5,45;0.05)	(5,45;0.01)
Treatments	16380.6	5	3276.12	13.03	yes	yes
Blocks	2790.6	9	310.07			
Error	11310.4	45	251.34			
Total	30481.6	59				

**Table 8.9:** shows the analysis of variance tables of the time measurements for the recognition task, where the treatments are either the pairs of queries with the same number of features or the individual queries. The resulting F-values are both significant at the 5% and the 1% level for either ANOVA, where the corresponding values of the F-distribution are equal to 2.43 and 3.5, respectively, at those levels.


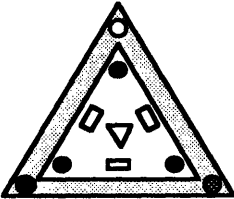
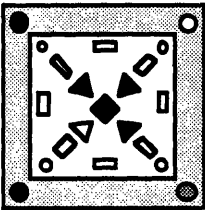
However, the analysis of variance of the time differences does reveal that there is a statistically significant difference between the treatments (see Table 8.9). Some queries took longer with the Boolean interface and others took more time with the InfoCrystal.

#### 8.4.1.3 Discussion

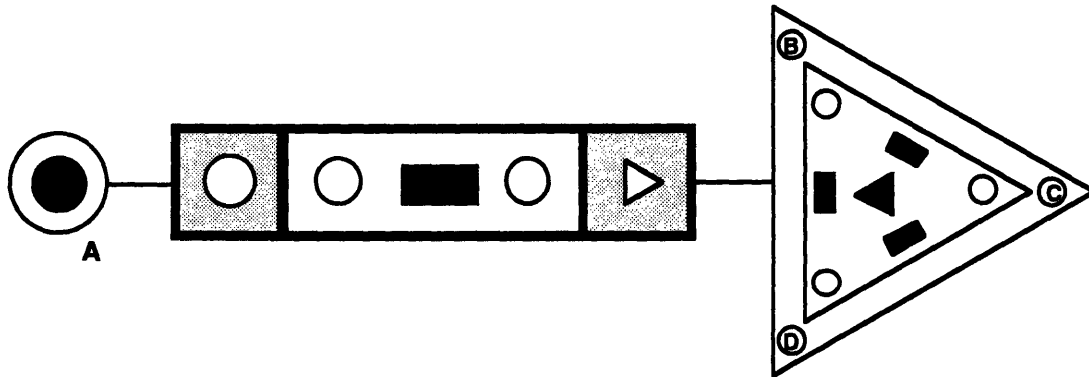
If we examine Table 8.7 (a) more carefully, then we can identify the following three clusters: 1) The second and fourth query clearly take less time in the InfoCrystal mode. The superior time performance can be explained by the fact that these two queries are easy to represent and recognize in an InfoCrystal, whereas they require the recognition of quite complicated query expressions in the Boolean mode (see Table 8.10). 2) The first, third and fifth query take slightly less time in the Boolean mode. This can be attributed to the fact that the natural language statements can be translated in a quite straightforward way into a Boolean query. These queries primarily use the AND operator and at times the NOT operator, whereas the occurring NOT operators require a greater cognitive effort for novice users when using the InfoCrystal. 3) The

sixth query clearly takes less time in the Boolean mode. This query has a hybrid structure because it combines requirements that are easy to express using the Boolean mode (e.g., "A AND ...") as well as ones that are easier to express using the InfoCrystal (e.g., "at least n out of m features"). Subjects have to be able to superimpose these two requirements when using the InfoCrystal, which can be especially challenging for novice users. In practice, it would be easier to construct a hierarchical query as shown in Figure 8.4, but that option, although implemented, was not made available to the subjects.

The analysis of variance of the time differences indicates that there are significant differences depending on the number of features used in a query. Looking at the rightmost column in Table 8.7 (a) we can see why this inference is possible. The grouped scores (-66, -176, and 314) are sufficiently different. However, the above discussion indicates that it would be more appropriate to distinguish between queries that are easier to represent in the

<p><b>InfoCrystal superior</b> 2nd (A xor B)</p>		<p>((A or B) and (not (A and B)))</p>
<p>4th Exactly 1 out of 3</p>		<p>((A and (not (B or C))) or (B and (not (A or C))) or (C and (not (A or B))))</p>
<p><b>Boolean superior</b> 6th (A and at least 2 out of 3 remaining)</p>		<p>(A and ((B and C) or (C and D) or (B and D)))</p>

**Table 8.10:** The first two rows show the two queries for which the InfoCrystal took less time than the Boolean mode. These queries are easy to represent in an InfoCrystal, whereas they require quite complicated query expressions in the Boolean mode. The bottom row shows the query for which the Boolean mode takes less time. This query has a hybrid structure because it combines requirements that are easy to express using the Boolean mode (e.g., "A AND ..."), and also ones that are easier to express using the InfoCrystal (e.g., "at least m out of n features").



**Figure 8.6:** shows a hierarchical InfoCrystal query that is equivalent to the InfoCrystal query shown in Table 8.10 (third row), but that is easier to program so as to represent the specified information need (A and at least 2 out of 3 remaining).

---

InfoCrystal than in the Boolean mode and vice-versa. Furthermore, the queries that use two or three features, respectively, do not consistently have a faster performance in the InfoCrystal. Actually, both groupings have a query that takes less time in the Boolean mode and one in the InfoCrystal mode, respectively. However, the latter ones take much less time than the former ones, causing the time difference for these grouping to be in favor of the InfoCrystal.

These experiments have helped us to understand better for which types of queries the InfoCrystal might be better suited. For example, the InfoCrystal is ideally suited for "m out of n features" types of queries. Further, the experiments suggest that users could benefit from a hybrid interface, where they could simultaneously use a Boolean and an InfoCrystal interface to formulate queries. This observation is also clearly articulated in the feedback received from the experimental subjects (see Chapter 9).

---



## **8.4.2 Results for the Generation Task**

In the generation task subjects had to create the correct Boolean or InfoCrystal query based on a textual description of the information need. A key issue we had to address is how to determine a score for the generated queries in both modes and how to compute the paired-difference score. We decided to make use of the fact that any valid Boolean query can be visualized in an InfoCrystal, causing the interior icons to be selected in a unique way. Hence, we can represent the correct query and any generated valid Boolean query in the form of InfoCrystals, and we can compute a score that reflects to what degree their associated selection patterns of the interior icons overlap. If they overlap perfectly, then the score is equal to one. If the selection patterns are just the inverse of each other, then the score is equal to zero.

We considered two ways of using these scores to compute the paired-difference score: 1) We assigned a categorical value of one if the InfoCrystal mode had a higher score, zero if both modes had the same score, and minus one if the Boolean mode had the higher score. 2) We simply set it equal to the difference between the scores for the two query languages.

As for the recognition task, we recorded for each query the amount of time it took a subject to perform the generation task. We then computed the difference between the recorded times for the InfoCrystal and the Boolean mode. Hence, a negative difference value implied that a subject took less time to create the query using the InfoCrystal than using the Boolean interface. The subjects had to use a standard command line interface to enter the Boolean queries, which could be a time consuming and tedious task. Hence, we would expect that the Boolean interface would require more time, especially for queries that use less than four features. However, there will come a point where the InfoCrystal is just as time consuming to program, because it contains so many interior icons that a subject has to consider. The time data reflects our expectations [see Table 8.17].

### **8.4.2.1 Generation Task Biased in Favor of Boolean Query Language**

In order to translate a generated Boolean query into the InfoCrystal, we had to ensure that the subjects only submitted valid Boolean queries. We accomplished this by automatically testing the validity of the generated

---

queries and giving the subjects feedback on how to modify currently invalid queries. However, we thereby eliminated a major source of errors that occur when creating Boolean queries [Borgman 1989, Young and Shneiderman 1993]. Young and Shneiderman [1993] found that almost half of the errors they observed in a similar generation task could be attributed to scoping errors or unbalanced parentheses. In essence, we focused only errors in the choice of the Boolean operators, whereas subjects usually also experience great difficulty in applying the brackets appropriately to achieve the desired nesting and to scope the operators correctly.

Hence, the generation task was biased in favor of the Boolean mode, because we only accepted and recorded valid Boolean queries, thereby eliminating a common source of errors. One of the advantages of the InfoCrystal is that any selection pattern of the interior icons corresponds to a valid Boolean query. We made the choice to only accept valid Boolean queries, because we wanted a consistent and fair way of scoring and comparing the generated queries in both modes. We also did not want to bias the experiment against the Boolean mode by assigning a score of zero to queries that are invalid because of mistakes in the placement of parentheses. To bias the experiment in favor of the Boolean mode represented to us a lesser evil, because we wanted to test if the InfoCrystal could be an effective interface to formulate Boolean queries.

The generation experiment was further biased in favor of the Boolean mode for the following reason. For queries that require a large percentage of the interior icons to be selected, it is easier to achieve a decent score by generating a valid Boolean query that is not perfectly coordinated than it is for the InfoCrystal. It is easy to fail to select all the icons that need to be selected, where this problem gets worse, the more icons that need to be selected. Table 8.21 reflects this fact because it shows that the major source of error for the InfoCrystal could be attributed to the fact that the subjects did not select all the necessary icons.

The main conclusion that we can draw from the analysis of the results for the generation task is that there is a statistically significant difference between the two query languages in favor of the Boolean mode. This should come as no surprise based on the above discussion. The analysis of variance for both

---

treatment types was statistically significant. As was the case for the recognition task, there were queries for which one of the two query languages performed much better. We had three such queries for the Boolean mode and one for the InfoCrystal. This fact was yet another reason why the Boolean mode performed better overall. Two of the queries favoring the Boolean mode required the selection of many of the interior icons, implying that it was easy for the subjects to miss selecting some of them in the InfoCrystal mode. For the query that was easier to express in the InfoCrystal, subjects tended to generate incorrect Boolean queries that hardly penalized the subjects in terms of the score, because their common mistake just had the effect that one icon was not selected that should have been. For the queries that favored the Boolean mode, the common mistakes in the InfoCrystal resulted in much lower scores.

A statistically significant difference could be detected in favor of the InfoCrystal for both the T-test and for the two analyses of variance of the time measurements. This result has to be interpreted with caution, because the standard Boolean interface used in this experiment required users to do quite a bit of typing, which was a tedious and time consuming task (the user feedback reflected this as well). Another advantage of the InfoCrystal is that it requires users only to select the appropriate interior icons, where the word "only" needs to be put in context: the larger the number of interior icons, the more time consuming and demanding it becomes for users to select all the correct icons.

#### **8.4.2.2 Categorical Paired-Difference Scores**

In this section we present and analyze the categorical paired-difference scores. Table 8.11 shows the actual scores. There are three queries that have mostly scores of -1, implying that the Boolean mode performed better, and there is one query that has predominately ones. We will examine these queries in more detail in Table 8.20. The T-test of the average of all six categorical paired-difference scores is only significant at the 5% but not at the 1% level (see Table 8.12). The analysis of variance of the scores does reveal a statistically significant difference between the treatments that are either the pairs of queries with the same number of features or the individual queries (see Table 8.13).

---

	1	2	3	4	5	6	7	8	9	10
(A or B)	- 1	0	0	0	0	0	0	0	0	0
(A and (not B))	0	0	0	0	- 1	0	0	0	0	0
(A or (not (B and C)))	0	- 1	- 1	- 1	1	- 1	1	- 1	- 1	- 1
Exactly 2 out of 3	0	1	1	1	1	1	0	1	1	1
(A or (not B)) and C and (not D))	- 1	0	- 1	0	0	- 1	0	- 1	- 1	- 1
((A or B) and (C or D))	0	0	- 1	- 1	- 1	0	0	0	0	0
Total	- 2	0	- 2	- 1	0	- 1	1	- 1	- 1	- 1
Average	-0.33	0	-0.33	-0.17	0	-0.17	0.17	-0.17	-0.17	-0.17

Correct	1	2	3	4	5	6	7	8	9	10
(A or B)										
(A and (not B))										
(A or (not (B and C)))										
Exactly 2 out of 3										
((A or (not B)) and C and (not D))										
((A or B) and (C or D))										

Table 8.11: the top table shows the categorical scores for the generation task, which have been calculated as follows: we assign a categorical value of 1 if the InfoCrystal mode has a higher score, 0 if both modes have the same score, and -1 if the Boolean mode has the higher score. The leftmost column displays the Boolean structure of the queries used in the experiment. We have also grouped the queries that use the same number of features by enclosing them by a thick black border. The bottom table shows in a graphical way, using light gray, for which queries the generated InfoCrystal query was correct and had a better score than the Boolean one. Similarly, black indicates where the Boolean mode was correct and had a better score. If a subject generated the correct query or the wrong query for both languages, then this is visualized in the bottom table using a gray or striped pattern, respectively.

<b>Paired-Difference T-test</b>			-1.833	-2.821
Mean	Est.St.Devi.	to	t(0.05,9)	t(0.01,9)
-0.13	0.15	-2.75	Boolean	no

Table 8.12: shows the mean and the estimated standard deviation of the averages of the categorical scores for the generation task. The resulting t-value of -2.75 is only significant for a t-distribution with 9 degrees of freedom at the 5%, but not at the 1% level, where the corresponding values of the t-distribution are equal to -1.833 and -2.821, respectively.

**ANOVA (treatments = number of features used in a query)** 4.41 6.01

	S.o.S.	D.o.F.	M.S.	Fo	(2,18;0.05)	(2,18;0.01)
Treatments	1.82	2	0.91	10.78	yes	yes
Blocks	0.63	9	0.07			
Error	1.52	18	0.08			
Total	3.97	29				

**ANOVA (treatments = individual queries)** 2.43 3.5

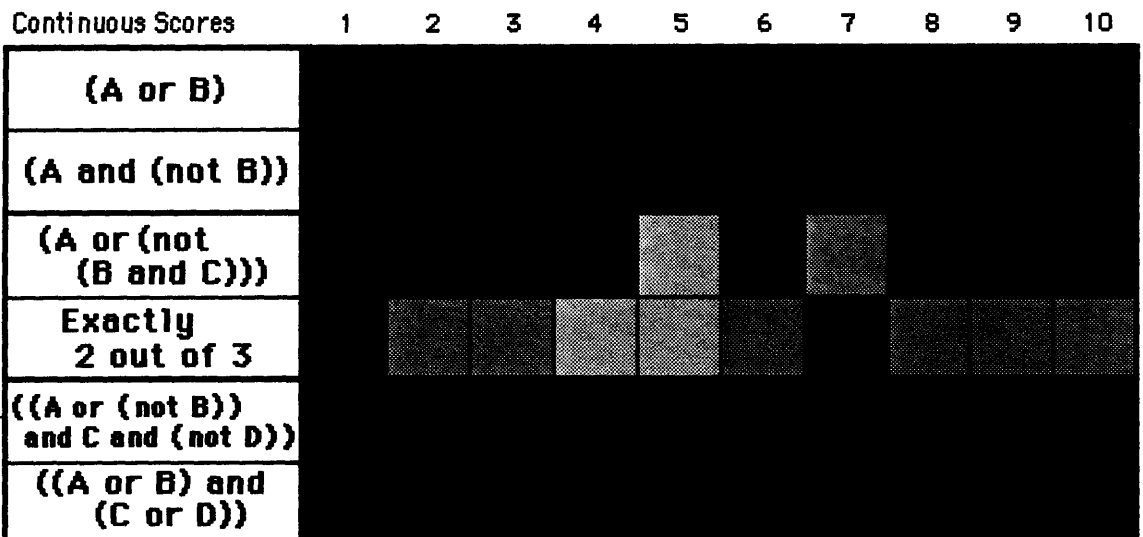
	S.o.S.	D.o.F.	M.S.	Fo	(5,45;0.05)	(5,45;0.01)
Treatments	12.53	5	2.51	8.59	yes	yes
Blocks	1.27	9	0.14			
Error	13.13	45	0.29			
Total	26.93	59				

**Table 8.13:** shows the analysis of variance tables of the categorical scores for the generation task, where the treatments are either the pairs of queries with the same number of features or the individual queries. The resulting F-values are significant at the 5% and the 1% level for either ANOVA.

### 8.4.2.3 Continuous Paired-Difference Scores

In this section we examine the continuous paired-difference scores for the two query languages. They are computed by taking the difference between the scores for the InfoCrystal and the Boolean version of a query. These individual scores reflect the degree of overlap between the selection patterns of the interior icons for the generated and the correct query. We obtained very similar results as for the categorical paired-difference scores. The only major difference is that here the analysis of variance did not detect any difference between the queries that use a different number of features.

	1	2	3	4	5	6	7	8	9	10
(A or B)	-0.67	0	0	0	0	0	0	0	0	0
(A and (not B))	0	0	0	0	-0.67	0	0	0	0	0
(A or (not (B and C)))	0	-0.43	-0.29	-0.43	0.43	-0.71	0.29	-0.71	-0.71	-0.43
Exactly 2 out of 3	0	0.14	0.14	0.57	0.57	0.14	0	0.14	0.14	0.14
(A or (not B)) and C and (not D))	-0.07	0	-0.20	0	0	-0.13	0	-0.13	-0.27	-0.07
((A or B) and (C or D))	0	0	-0.40	-0.47	-0.13	0	0	0	0	0
Total	-0.73	-0.29	-0.74	-0.32	0.20	-0.70	0.29	-0.70	-0.84	-0.35
Average	-0.12	-0.05	-0.12	-0.05	0.03	-0.12	0.05	-0.12	-0.14	-0.06



**Table 8.14:** the top table shows the continuous paired-difference scores for the generation task, which have been calculated by just taking the difference between the score for the InfoCrystal and the Boolean version for the same query. The bottom table displays the same scores in a graphical way, where gray represents zero and a gray tone closer to black / white implies that the Boolean / InfoCrystal interface performed better, respectively.

Paired-Difference T-test			-1.833	-2.821
Mean	Est.St.Devi.	to	t(0.05,9)	t(0.01,9)
-0.07	0.07	-3.30	Boolean	Boolean

**Table 8.15:** shows the mean and the estimated standard deviation of the averages of the scores for the generation task. The resulting t-value of -3.30 is significant for a t-distribution with 9 degrees of freedom at both the 5% and the 1% level in the favor of the Boolean mode.

ANOVA (treatments = number of features used in a query)					4.41	6.01
	S.o.S.	D.o.F.	M.S.	Fo	(2,18;0.05)	(2,18;0.01)
Treatments	0.01	2	0.005	0.125	no	no
Blocks	0.12	9	0.01			
Error	0.69	18	0.04			
Total	0.82	29				

ANOVA (treatments = individual queries)					2.43	3.5
	S.o.S.	D.o.F.	M.S.	Fo	(5,45;0.05)	(5,45;0.01)
Treatments	1.27	5	0.25	4.03	yes	yes
Blocks	0.24	9	0.03			
Error	2.84	45	0.06			
Total	4.35	59				

**Table 8.16:** shows the analysis of variance tables of the scores for the generation task, where the treatments are either the pairs of queries with the same number of features or the individual queries. The resulting F-values are significant at the 5% or the 1% level for only the ANOVA, where the treatments are equal to the individual queries.

#### 8.4.2.4 Time Measurements

In this section we examine the amount of time it took subjects to generate a query. Table 8.17 (a) shows the difference between the time measurements for the InfoCrystal and Boolean version of a query; (c) shows the time measurements for the InfoCrystal; (d) shows the percentile difference in the time measurements between the two query languages. We have already mentioned that these time differences have to be interpreted with caution, because the standard Boolean interface used in this experiment required users to do quite a bit of typing, which can be a tedious and time consuming task.

(a) Time Differences between the InfoCrystal and the Boolean mode

	1	2	3	4	5	6	7	8	9	10		
(A or B)	-43	-16	-13	-17	-19	-25	-49	-5	-13	-6	-206	-442
(A and (not B))	2	-33	-34	-41	-24	-15	-30	-10	-20	-31	-236	
(A or (not (B and C)))	11	5	-25	-107	-290	-15	-110	-38	-2	-15	-586	-1663
Exactly 2 out of 3	-210	-38	-23	-246	-99	-29	-195	-42	-138	-57	-1077	
(A or (not B)) and C and (not D))	36	-5	17	-86	-56	-5	24	6	-24	-23	-116	76
((A or B) and (C or D))	5	-10	2	130	-38	9	5	7	83	-1	192	
Total	-199	-97	-76	-367	-526	-80	-355	-82	-114	-133		
Average	-33.2	-16.2	-12.7	-61.2	-87.7	-13.3	-59.2	-13.7	-19.0	-22.2		

(b) Times difference between the two modes represented visually

Time	1	2	3	4	5	6	7	8	9	10		
(A or B)											-206	-442
(A and (not B))											-236	
(A or (not (B and C)))											-586	-1663
Exactly 2 out of 3											-1077	
((A or (not B)) and C and (not D))											-116	76
((A or B) and (C or D))											192	

Table 8.17: (a) shows the difference in the time measurements, measured in seconds, between the InfoCrystal and the Boolean query language for each query. (b) displays the time differences in a graphical way, where light gray indicates that the InfoCrystal was faster, and black that the Boolean interface took less time.



(c) Times for InfoCrystal

	1	2	3	4	5	6	7	8	9	10
(A or B)	8	10	15	7	16	7	13	11	31	22
(A and (not B))	32	6	12	8	16	14	11	13	20	10
(A or (not (B and C)))	49	59	54	80	75	43	53	15	92	40
Exactly 2 out of 3	16	12	40	62	21	21	37	20	20	18
(A or (not B) and C and (not D))	138	113	127	137	76	119	155	68	129	123
((A or B) and (C or D))	52	37	65	193	64	51	55	47	142	62

(d) Percentile Difference between the two query languages

	1	2	3	4	5	6	7	8	9	10
(A or B)	-84%	-62%	-46%	-71%	-54%	-78%	-79%	-31%	-30%	-21%
(A and (not B))	7%	-85%	-74%	-84%	-60%	-52%	-73%	-43%	-50%	-76%
(A or (not (B and C)))	29%	9%	-32%	-57%	-79%	-26%	-67%	-72%	-2%	-27%
Exactly 2 out of 3	-93%	-76%	-37%	-80%	-83%	-58%	-84%	-68%	-87%	-76%
(A or (not B) and C and (not D))	35%	-4%	15%	-39%	-42%	-4%	18%	10%	-16%	-16%
((A or B) and (C or D))	11%	-21%	3%	206%	-37%	21%	10%	18%	141%	-2%

Table 8.17 (cont.): (c) displays the time measurements for the InfoCrystal. (d) shows the percentile difference between the two query languages by dividing the entries in table (a) by the time measurements for the Boolean mode of a query.

Table 8.18 shows the outcome of performing the T-test on the average of all six time differences between the two query languages. It is significant at both the 5% and 1% level in favor of the InfoCrystal. The analysis of variance of the time differences does also reveal a statistically significant difference in favor of the InfoCrystal between the treatments, where the treatments are either the pairs of queries with the same number of features or the individual queries (see Table 8.19).

<b>Paired-Difference T-test</b>			-1.833	-2.821
Mean	Est.St.Devi.	to	t(0.05,9)	t(0.01,9)
-33.82	26.31	-4.06	InfoCrystal	InfoCrystal

Table 8.18: shows the mean and the estimated standard deviation of the averages of the time difference for the generation task. The resulting t-value of -4.06 is significant at the 5% and the 1% level.

**ANOVA (treatments = number of features used in a query)**                      4.41                      6.01

	S.o.S.	D.o.F.	M.S.	Fo	(2,18;0.05)	(2,18;0.01)
Treatments	39860.7	2	19930.4	12.61	yes	yes
Blocks	18691.7	9	2076.9			
Error	28445.3	18	1580.3			
Total	86997.7	29				

**ANOVA (treatments = individual queries)**                      2.43                      3.5

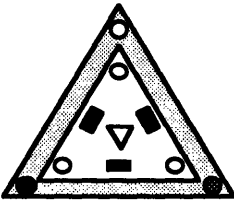
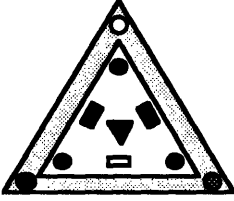
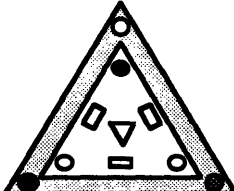
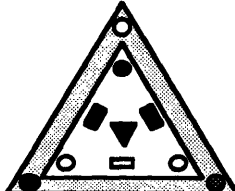
	S.o.S.	D.o.F.	M.S.	Fo	(5,45;0.05)	(5,45;0.01)
Treatments	96564	5	19312.7	6.22	yes	yes
Blocks	37383	9	4153.7			
Error	139744	45	3105.4			
Total	273691	59				

**Table 8.19:** shows the analysis of variance tables of the time measurements for the generation task, where the treatments are either the pairs of queries with the same number of features or the individual queries. The resulting F-values are both significant at the 5% and the 1% level for either ANOVA.

---

8.4.2.5 Discussion

As for the recognition task, it is instructive to examine the queries for which one of the query languages clearly performed better. Table 8.20 shows in its top row the query that subjects found easier to formulate using the InfoCrystal. This query is of the type "m out of n" that can be quite cumbersome to express using the Boolean mode. The other rows show the queries where the subjects consistently performed better using the Boolean mode.

<p><b>InfoCrystal superior</b></p> <p>4th</p> <p>Exactly 2 out of 3</p>		<p>((A or B or C) and (not (A and B and C)))</p> <p>A common mistake in the Boolean mode was that the subjects generated the query ((A and B) or (B and C) or (A and C)) that selects all the three icons of rank two, as desired, but also the unwanted center icon of rank three, a mistake that did not affect the score greatly.</p>
<p><b>Boolean superior</b></p> <p>3rd</p> <p>(A or (not B) or (not C))</p>		<p>(A or (not B) or (not C))</p> <p>A series of OR operators combined with the NOT operator can be challenging for novice users to translate into an InfoCrystal. Subjects either chose the most conservative approach by interpreting the OR's as AND's, resulting in (a); or they just in effect applied A and ignored the rest of the query, resulting in (b).</p> <div style="display: flex; justify-content: space-around; align-items: center;"> <div data-bbox="576 1585 808 1827">  <p>(a)</p> </div> <div data-bbox="1006 1585 1239 1827">  <p>(b)</p> </div> </div>

<p>5th</p> <p>(A or (not B)) and C and (not D))</p>		<p>(A or (not B)) and C and (not D))</p>
<p>6th</p> <p>((A or B) and (C or D))</p>	<p>Many of the subjects were able to infer from the information need statement that many of the icons of rank two had to be selected, see (a). Then, they had to realize that icons with a rank greater than two had to be selected and the icon of rank of four was the next easiest icon to select, see (b).</p> <div style="display: flex; justify-content: space-around;"> <div data-bbox="675 989 873 1247"> <p>(a)</p> </div> <div data-bbox="1089 989 1287 1247"> <p>(b)</p> </div> </div>	<p>((A or B) and (C or D))</p>

**Table 8.20:** The first row shows the query for which the InfoCrystal has a clearly better score than the Boolean mode, because it is easy to represent it in an InfoCrystal, whereas it requires the formulation of a quite complicated query expression in the Boolean mode. The other rows show the query for which the Boolean has a better score. Two of these queries require the selection of many interior icons in the InfoCrystal mode, where it is easy for the subjects to fail to select all the necessary ones.

We examined all the queries generated by the subjects that were not correct. There are two types of errors that occurred: 1) The incorrect query does not select all the necessary relationships among the features (see Misses column in Table 8.21). 2) The incorrect query in effect includes unwanted relationships (see False Alarms column in Table 8.21). We have noted that the number of interior icons increases exponentially as the number of concepts or features increases. Consequently, there will come a point, where it

	Misses	False Alarms
InfoCrystal	56 (28)	13
Boolean	19	14

**Table 8.21:** An incorrect query can be characterized in terms of the relationships between the features that it failed to select, i.e., the misses, and the ones it should not have selected, i.e., the false alarms. This table shows the number of interior icons that were missed or incorrectly selected in all the incorrect queries generated by the subjects for both query languages. The number in the brackets refers to the number of interior icons that were missed by the incorrect queries generated for the third query in the InfoCrystal mode. As we have discussed in Table 8.20, for this particular query the subjects found it difficult to elaborate all the possible relationships as they pertained to this query. Hence, they chose a conservative strategy to select only those icons that clearly satisfied the information need.

---

will be quite demanding for the subjects to explore all the interior icons to generate a query without missing some of the icons that need to be selected. Hence, we expect that the subjects tend to fail to select all the necessary interior icons instead of selecting unwanted interior icons (assuming that initially all the interior icons are not selected).

---

## 8.5 Lessons Learned and General Discussion

In a certain respect, the InfoCrystal is more demanding than the Boolean mode, because it requires users to really understand the structure of the information need, whereas subjects could often "copy" the textual information need to create a textual Boolean query without really having to fully understand the implication of its logical structure. Ideally, we would like to phrase the information needs in a way that required an equivalent effort to translate them into the Boolean query and the InfoCrystal mode. Hopefully, in a future experiment we can create such a set of information needs.

We purposely chose to have the subjects use a version of the InfoCrystal that did not have the enhancing features to be outlined below and elsewhere in this thesis, because we wanted to see how well they could use the InfoCrystal in its most basic form to translate the specific information needs into the appropriate selection pattern of the interior icons. The feedback received from the subjects asked for some of the features outlined below and it can serve as an independent confirmation that these features could make the InfoCrystal a more effective tool.

One of the arguments presented in favor of the InfoCrystal is that it does not require users to think in terms of Boolean algebra to formulate a query. They can think spatially and they need to decide which parts of the space of relationships that they want to explore by selecting the corresponding interior icons. There are, however, instances where users have to be able to translate a specific information need, as was the case in these experiments, or they have a set of preferences and they need to figure out how to program the InfoCrystal accordingly. Hence, it is worth stressing at this point that the InfoCrystal has been or can be easily extended in the following ways to assist users in the task of "programming" it:

First, the InfoCrystal has the built-in capability to show the Boolean query that is equivalent to the current selection pattern of the interior icons. Hence, users can interact with the interior icons and thereby incrementally create the desired Boolean meaning. There are, however, many ways of writing Boolean queries that have equivalent meanings. One of the issues that needs to be further investigated is how to reduce a Boolean query to a form that expresses its meaning in the most concise way. There are methods for

---

performing this reduction process automatically, and we will implement them in our future research.

Second, if users are able to formulate a Boolean query that reflects their information need but they do not know how to represent it in an InfoCrystal, then we have developed a mechanism that can perform the translation automatically. Actually, we can translate any valid Boolean query into an InfoCrystal and vice-versa.

Third, if users do not know how to formulate a Boolean query, but they feel comfortable assigning relevance weights to the concepts, then we can use these weights to rank and select the interior icons that are above a certain threshold. The weights could also be computed automatically, using techniques employed by statistical retrieval approaches.

Fourth, users can click on the criterion icon and by holding down certain keys they can formulate a subset of Boolean queries in a similar way that they use calculator to add and subtract numbers by operating on the current value held in the accumulator (as discussed in section 4.2.2)

### **8.5.1 Difference Between the Two Query Languages**

The fact that there are queries that are easier to formulate using one of the two query languages encouraged us to analyze the difference between the Boolean and the InfoCrystal query languages in a little more detail. The following observation can help us to understand some of their differences: On the one hand, the InfoCrystal operates at the lowest possible level of abstraction, because it represents all the possible queries in disjunctive normal form. Its interior icons represent the disjoint constituents that are the necessary and sufficient to create any query. On the other hand, the Boolean query operates at higher level of abstraction. Hence, it makes it easy to express certain high-level statements that will require more work to be pieced together by selecting the appropriate interior icons. However, there are very specific and complex queries that are very cumbersome to formulate using these more general or bulky Boolean constructs.

An alternate, but related way of understanding the difference between the InfoCrystal and Boolean query language is to use a geometrical analogy. On the one hand, the interior icons can be thought of as the atomic shapes out of which any geometric shape can be created. On the other hand, the

---

components in a Boolean query constitute larger shapes that make the creation of certain shapes very straightforward, but are difficult to use and coordinate when a complex shape needs to be created. To help the reader understand this analogy better, we can think of a specific shape as being defined by a particular subset of selected interior icons and vice-versa. In particular, if we include feature "A" in a Boolean statement, then we activate all the interior icons or constituents that contain "A" in a positive way. If the NOT operator precedes "A", then it just reverses the activation pattern and activates all the interior icons or constituents that do not contain "A". Hence, we can imagine that each concept in a Boolean statement has a selection pattern of the interior icons associated with it. In order to determine the final selection pattern, we need to integrate these different selection patterns. If two concepts are connected by the AND operator, then we intersect the selection patterns associated with concepts. If two concepts are connected by the OR operator, then we take the union or merge the selection patterns associated with the concepts.

### **8.5.2 Conclusion**

The recognition and the generation task only tested a specific aspect of the InfoCrystal interface and they did not test all of its valuable or promising features. Still, this user study has produced the following useful results: 1) Although novice users received only a short, fifteen minutes long tutorial, they were able to successfully use the InfoCrystal. This second version of the tutorial made a big difference in terms of how well and quickly users could learn to use the InfoCrystal. Further improvements in the way novice users are instructed to use the InfoCrystal will help them to make full use of its rich set of features and the advantages that it has to offer. 2) The study showed that the InfoCrystal, even at an early stage of development, performed as well as the familiar Boolean interface, although it was biased in favor of the Boolean mode (as discussed in section 8.4.2.1). 3) On the one hand, the user study confirmed that the InfoCrystal is ideally suited for queries of the form "at most, exactly, or at least n out of m features". On the other hand, the study showed that certain Boolean queries are more difficult to formulate using the InfoCrystal than the Boolean interface. However, we believe that users can improve their performance with more practice and if they have access to the

---



enhancing features of the InfoCrystal that have been implemented, but were not made available during the experiments. 4) The user feedback concerning the InfoCrystal interface was very encouraging and it helped to pinpoint possible improvements (see chapter 9).

We plan to conduct further user studies that will examine the ability of users to reformulate queries. We expect that the InfoCrystal should perform well and demonstrate how it supports the query reformulation process. The InfoCrystal makes it easy to broaden or narrow a query, because it represents all the possible queries in a single display. A major advantage of the InfoCrystal is that it uses a simple metaphor to visualize the broadness of a query: the larger the visual area, the broader the query. In other query languages, such as the Boolean one, it is much more demanding and cumbersome to broaden or narrow a query. It can require a deep understanding of these query languages. Furthermore, the InfoCrystal has the attractive quality that enables users to better predict what the consequences of certain changes will be. This is not necessarily the case with other query languages.

Although it is easy to broaden or narrow an InfoCrystal query, there are multiple ways to achieve it. We are again faced with the fact that users have to understand the meaning of the interior icons to be able to modify an InfoCrystal in a precise and desired way. Users can use the quantitative information associated with the interior icons to help them decide how to modify an InfoCrystal, provided their main concern is to change the amount of information that is being retrieved. If, however, they want to change the structure of the retrieved information, then one of the prerequisite is that they understand the meaning of the interior icons. Hence, the recognition task and generation task did address an issue central to the successful use of the InfoCrystal.

To conclude, we hope to implement the InfoCrystal in an environment that enables us to use it as front-end to several and diverse retrieval engines that can rapidly search large information spaces. We believe once we can use the InfoCrystal to explore large information spaces in real-time that some of its advantages and strengths can be more fully demonstrated and that the further improvements will suggest themselves.

---



## CHAPTER 9

# USER FEEDBACK

In this chapter we will present the feedback that we received from the subjects who participated in the user studies. We recorded their spontaneous comments during the experiments and we asked them after each of the four sets of queries<sup>1</sup> if they had any comments or observations. These comments are summarized in the second column of the feedback table, provided they are different from answers found elsewhere in the feedback table. At the end of the experiments we asked a set of three questions, which are displayed in the other column headings, and the answers are summarized in the appropriate columns.

The feedback is instructive in several ways. First, the overall response to the InfoCrystal was positive, ranging from "The InfoCrystal was absolutely clear. ... It was much, much easier with the InfoCrystal. ... I felt confident with the InfoCrystal," to "the InfoCrystal was actually not that bad, even usable." Second, the feedback touched on and clearly articulated features that need to be made available to users. We purposely chose to have the subjects use a version of the InfoCrystal that did not have all the enhancing features that we have implemented or that we plan to develop, because we wanted to see how well they could use the InfoCrystal in its most basic and primitive form. In particular, the following feedback is worth highlighting:

- **Queries Easy to Formulate with the InfoCrystal:** Several subjects observed that the InfoCrystal is better suited for certain queries, such as queries of the type "at least, exactly, or at most M out of N features". The experimental results support this observation.

---

<sup>1</sup> We presented the subjects first with a training set and then with the set of test queries for both the recognition and the generation task.

---

- **Inverse Relationship between Boolean and InfoCrystal Queries:** One of the subjects, in particular, articulated that there is a "confusing" relationship between the cognitive effort required to formulate certain queries. Simple Boolean queries that predominately use the OR operator to coordinate components often require users to select many icons in the InfoCrystal mode. A Boolean query with a simple or repetitive structure can be easy to grasp because it reduces into a simple pattern, whereas it requires users to examine and select many of the interior icons in an InfoCrystal. Once users have more experience with the InfoCrystal representation, they will be able recognize how certain selection patterns are related to Boolean statements. Having a Boolean feedback window will help users to learn this (see also Hybrid Interface bullet). Further, we can assist users so that they can select groups of interior icons by interacting with the border or criterion icons and use the InfoCrystal as a Boolean Calculator (see section 4.2.2).
  - **Combinatorial Explosion:** Many of the subjects commented that InfoCrystal queries with two or three features were straightforward to interpret and use, but that InfoCrystal queries with four features were overwhelming at first. This should come as no surprise because the number of interior icons grows exponentially as the number of features increases.

In a similar vein, a subject articulated that the InfoCrystal in its most basic form requires users to examine all icons, especially if we have a complex information need. This is an expensive and cognitively demanding operation, especially when the number of features considered at same time is more than three.
  - **Translation of Complex Queries in Stages:** Some of the subjects would have liked to have been able to translate complex queries and perform the corresponding selections in the InfoCrystal in stages. This request touches on the issue of being able to save partial results in the InfoCrystal framework and then to combine these partial results to arrive at the final selection pattern. We address this issue in sections 4.2.1 and 4.2.2.
-

- **How to juggle the different possibilities at the same time ?** One subject commented that the OR operator and especially the OR combined with the NOT operator was difficult to translate into the InfoCrystal. The OR implies that there are multiple ways to satisfy the information need. The InfoCrystal in its basic form requires users to think about the different possibilities simultaneously, whereas it would be easier for users to consider each of these possibilities one at a time or in stages. In section 4.2.2 we describe how the InfoCrystal has been or can be extended to address this issue.
  - **Hybrid Interface:** Several subjects concluded that it would be advantageous to have a hybrid retrieval interface, where they could use a textual Boolean and the InfoCrystal interface to formulate queries. Similarly, subjects noted that it would have been helpful if there had been a feedback window that showed the Boolean query that was equivalent to the current selection pattern of the interior icons. We could easily provide this type of feedback, but we wanted to test how well novice users could use the InfoCrystal without it.
  - **List Interface:** One of the subjects suggested that it could be beneficial to see a list of the data items that are retrieved by the selected icons. By examining individual list items users could determine if they are retrieving the information that they are looking for. We have implemented ranked-list interface, but, as we have stated before, we wanted to see how well the subjects could use the InfoCrystal in its most basic and primitive form.
-

	First Response	What was easy? What worked?	What was difficult?	Possible Improvements? Final Response
1 Male 40+ Ph.D. Extensive library searching	Would like both (tradeoff). Complex IC queries require process of elimination.	If Boolean terminology in info. need then Boolean mode at an advantage. IC better suited for certain queries: at least M features out of N.	Typing in Boolean mode. Typing an awkward interface => I am becoming a fan of IC. Need to examine all icons in complex IC queries => expensive operation.	Love both => mixed interface. To be able to do IC selection in stages. Experiment fun and thought provoking.
2 Male 20+ B.Sc. Designed databases	Consistency of experimental interface. IC very nice and easy to get used to.	Writing the Boolean queries	Translating the info. need into the IC.	Possible use of sounds.
3 Male 20+ B.Sc. Knows Boolean logic	Idea of IC beautiful: relationship between number of features and shape. Confusing: simple Boolean query requires you to select many icons in IC mode.	IC easier - liked it.	Typing in Boolean mode.	
4 Female 40+ Ph.D. Familiar with Boolean logic	Difficulty with NOT & OR with IC. The usual issues with text queries: brackets and grammar. Much, much easier with IC. I felt confident with IC.	Simple queries were easier in IC mode than Boolean mode. When asked to find exactly two - easy in IC.	NOT & OR with 3+ features considered at same time in IC.	Have to get used to IC. Useful to get textual feedback when interacting with IC.
5 Female 30+ B.A.		To me, to actually see it in terms of shapes and colors Entering Boolean queries easier than recognizing them.	When to use AND or OR.	IC was absolutely clear.

**Table 9.1:** summarizes the user feedback received during the evaluation experiments and in response to specific questions. The user population consisted of 4 women & 6 men; all had at least a College Education and all had some experience with Boolean Retrieval. Legend: IC = InfoCrystal.

	First Response	What was easy ? What worked?	What was difficult ?	Possible Improvements? Final Response
6 Female 20+ M.Sc. Experience with SQL and Boolean logic	IC neat to play with.	Trying to get the hang of the icons' meaning and how to combine them. Neither of the two query languages is always easier than the other.	IC with four features is difficult to think in terms of Venn Diagrams.	If I reread the instructions for IC again, then this would clear up some things. IC was actually not that bad, even usable.
7 Male 30+ M.Sc. Knows Boolean logic	The two query languages require you to think about task in different ways.	Progressively translate info. need into IC: OR -> select, AND -> deselect.	Translation of info. need into IC is more demanding. Typing queries in Boolean mode. Boolean queries are very wordy.	To get textual feedback on query formulated while interacting with IC. To see list of output of selected icons.
8 Male 20+ B.Sc. Knows Boolean logic	Preferred the IC. Really cool. Thought it would be harder.	Color recognition was easy and helped.	Boolean expressions with all the brackets and operators.	
9 Female 30+ M.Ed. Been exposed to Boolean logic long time ago	IC queries are easier to read by a long shot. Can not read the Boolean queries - what are these brackets ...	IC much easier to use, especially for simpler queries.	All these options to think about in IC, whereas in Boolean mode you formulate a specific query (do this!). Depends on the query.	To get textual feedback on query formulated while interacting with IC. Help with OR in IC.
10 Male 20+ M.Sc. Knows Boolean logic	IC faster.	IC queries with two or three features very straight-forward.	IC queries with four or more features require you to be very careful about the combinations: you might miss one.	Give user textual feedback about work in progress in IC -> will increase user's confidence.

**Table 9.1 (cont.):** summarizes the user feedback received during the evaluation experiments and in response to specific questions. The user population consisted of 4 women & 6 men; all had at least a College Education and all had some experience with Boolean Retrieval. Legend: IC = InfoCrystal.





# CHAPTER 10

## RELEVANT RESEARCH

In this chapter we will describe work by other researchers that is of direct relevance to this thesis. We will indicate how the work mentioned here complements or could be integrated with the InfoCrystal. We will also try to situate the InfoCrystal by comparing it directly with this relevant and previous work. In particular, we will focus on the shortcomings of the existing proposals, as we perceive them, and we will indicate how the InfoCrystal addresses them. We hope this type of exposition will better motivate the approach taken in this thesis as well as help to focus the discussion of what types of information tools will be effective. We will be considering the following approaches: overview maps, ways to visualize hierarchical structures, applications of familiar metaphors for accessing information and ways to display the content of documents, and visual query languages.

### 10.1 Overview Maps

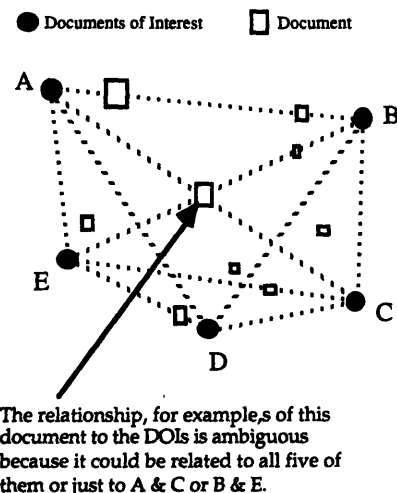
Several researchers have suggested that some sense of the topography of an information or document space would be useful in the retrieval of information. The basic motivation for such an overview map is to enable users to (re)formulate their queries based on a better sense of the document space, to allow them to browse through the document collection and to enable users to use a map as the retrieval interface. We will now describe several of these overview maps that have been developed. The main difference between these approaches is not so much their visualization metaphor but the particular functions used to perform the mapping of higher-dimensional space into a 2D or 3D display:

---

**VIBE (Visualization By Example):** Korfhage et al. (1991) have developed a system where the contents of a document space are displayed based on their similarity to several documents of interest (DOI) that have been selected by the user. The dimensionality of this similarity space is a function of the number of DOIs, which must be greater or equal to three. Hence, Korfhage et al. have to introduce a mapping that reduces the dimensionality to end up with a two-dimensional representation.

They model the similarity space between the user-defined documents of interest (DOI) and the documents by using the ratios of their distances from the DOIs. The resulting loss of information, however, leads to interpretation difficulties because documents with different relationships to the DOIs can superimpose on the same image point. Hence, the user cannot uniquely infer from the display how a document is exactly related to the defined DOIs. The display gives the user only an approximate sense of the relationships between the contents of the document space and the DOIs. The authors claim that any superposition of document points that is not due to equal distance ratios can be resolved by manipulating the image, but the user must be sufficiently versed in use of VIBE to understand this. In VIBE the position of a document depends on the relative distances from the POIs. Hence, a document point can be located close to a certain POI, and still contain relatively little information about that POI. To show the absolute strength of the relationships, VIBE has to use the size coding that is proportional to the strength of the most significant POI. The VIBE suffers from the problem that the display points do not have a unique and straightforward interpretation, because documents with different relationships to the DOIs can superimpose on the same display point. Hence, the VIBE display can be hard to interpret.

- The InfoCrystal shows how the contents of a document space relate to several points of interest, but with the following important difference: The InfoCrystal imposes a structure on how the relationships are visualized so



**Figure 10.1:** VIBE.

that the display points have a unique and straightforward interpretation. This property is exploited to use the display as a visual interface to formulate Boolean as well as weighted queries. Furthermore, the InfoCrystal scales well because it can visualize categorical relationships instead of just individual documents, which will clutter the display as their number increases.

**LyberTree and LyberSphere:** Hemmje et al. (1994) have developed a 3D based visual interface that consists of the LyberTree and LyberSphere visualization modules. Hemmje et al. use the Cone Tree metaphor developed by Card et al. (1991) for visualizing the content spaces, and the VIBE representation to represent the results of a query (see 10.2 for a description of the Cone Tree metaphor). They extend the VIBE representation from a 2D circle to a 3D sphere, where the retrieved documents are mapped onto the sphere in accordance to their relevance to a query. Further, they transform document term networks with two levels of abstraction into hierarchical and directed Cone Trees that use spatial depth to achieve the perception of their topological structure. If the currently selected item is a document, then its new subtree will consist of all its specific terms. This set of terms is automatically generated by a probabilistic information retrieval system. If the selected item is a term, then its new subtree will contain all the documents in which the term is contained. Hemmje et al. hope that users will easily recognize that they have visited certain areas before, because the geometry of the LyberTree looks the same or because the topology requires the same, repeated navigational decisions from the user.

- We have outlined how the InfoCrystal differs from the VIBE approach of visualizing an information space. Hence, the same comments apply for LyberSpheres.

Next we will discuss several methods that use clustering approaches to devise a 2D or 3D representation of a large information space. We will discuss at the end of this section how these clustering approaches are related to the InfoCrystal:

---

**Semantic Maps:** Lin et al. (1991) propose to use a two-dimensional map as a retrieval interface and they use Kohonen nets to cluster and display the contents of a document space. A Kohonen net automatically organizes documents into a two-dimensional array, using a vector representation to drive the net; and it then labels the documents with the most frequent keyword. The resulting map is divided into concept or cluster areas, which is supposed to reveal the frequencies and distributions of the underlying data. When mapping higher-dimensional information into a lower-dimensional space, a loss of information and some distortion is unavoidable. The authors claim that document interrelationships are faithfully maintained by the Kohonen net transformation. This map is intended as a browsing tool to support ordering, linking and browsing information gathered by more traditional filters. The authors suggest that such a map representation could make it easier to identify relevant documents from a large retrieved set and therefore make low precision/high recall results more acceptable. It is, however, not clear how well this approach will work with large data collections.

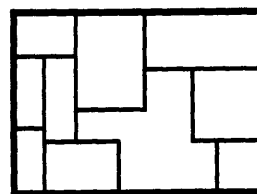


Figure 10.2: Semantic Map.

**Cybermap:** has been developed by Gloor (1991) and is also intended to give a two-dimensional overview of a document space. It is generated based on an index of all words contained in the documents, where the index has been created using automatic indexing techniques.

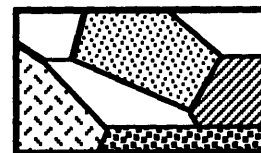


Figure 10.3: Cybermap.

On the basis of on the index, weighted keyword vectors of all documents are computed. A simple clustering technique is used to partition the document space into non-overlapping "hyperdrawers", using a centroid based technique. Initially each document is put in a separate hyperdrawer. Nodes are then added sequentially to the drawers one at the time. The document *d* is added to the hyperdrawer *h* that has the highest similarity value.

**Interactive Clustering:** Faieta and Lumer (1994) have developed a statistical clustering algorithm that is based on collective processing and self-organization. It supports the dynamic visualization and direct manipulation

---

of emergent structures present in multi-dimensional data sets. It uses a standard two-dimensional grid to display its results. The algorithm first places database elements randomly in a grid and then aggregates them in a particular way so that statistical regularities are mapped into spatially structured clusters. The grid is displayed so that users can probe and alter the characteristics of the clusters as they formed. Their clustering algorithm differs from others in that users have the opportunity to intervene at various points of the clustering process. It is the users who decide what they consider is a cluster based on what they see forming on the screen. Commonly, multi-dimensional scaling methods try to conserve distances between points so that points close together in  $n$ -dimensional space are mapped close together in the lower dimensional space. Their algorithm, however, relaxes this requirement to end up with a faster algorithm for visualizing the clustering process.

**BEAD:** Chalmers et al. (1992) have developed a system to represent documents as particles in a three-dimensional space. They have set up rules of physical behavior for these particles that are driven by the documents' characteristics. These rules are chosen to make spatial proximity approximate thematic similarity.

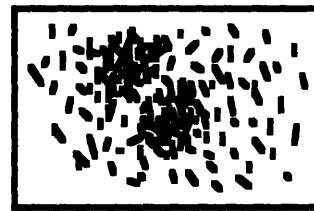


Figure 10.4: BEAD.

By using physically based modeling techniques to take advantage of fast methods for approximating potential fields, they represent the relationships between documents by their relative distance. Inter-particle forces tend to make similar articles move closer to one another and dissimilar ones move apart, resulting in a 3D space that can be used to visualize patterns of a high-dimensional document space. The user can explore this space either by inspecting the three orthogonal plots (in XY, XZ and YZ) or watching an animated perspective view as seen from the point of view rotating around the center of mass. If a query is issued then each particle is given a color according to its document distance from the query. The user can zoom in on an appropriately colored particle in order to see how neighboring particles are related to each other. As mentioned before, the lower dimensionality of the space into which the document space is mapped into, the greater the loss of information and the more approximate the representation of document relationships.

**Scatter/Gather:** Cutting et al. (1991) do not propose an actual visual representation, but they show how clustering techniques could be used as an information access tool. Scatter/Gather enables users to browse in a query-free way large document collections. They take their inspiration from the table-of-contents access method typically provided with a textbook.

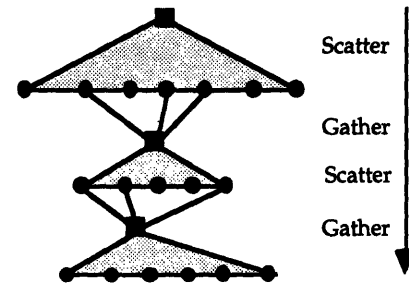


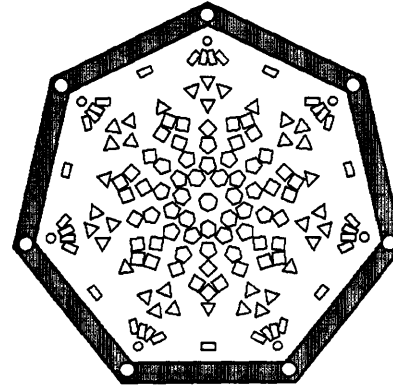
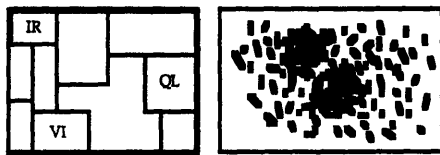
Figure 10.5: Scatter/Gather.

Initially their system scatters the document space into a small number of document groups, or clusters, and presents short summaries of them to the user. On the basis of on these summaries, the user selects one or more of the groups to form a subcollection. The system then applies clustering again to scatter the new subcollection into a small number of document groups, which are again presented to the user. With each successive iteration the groups become smaller, and therefore more detailed. Their technique is directed towards information access with non-specific goals and is intended as a complement to more focused methods. Their technique is meant to be used by users who have non-specific goals or who have difficulty formulating their queries because they are not sure which terms are appropriate. This technique could also be helpful in situations in which is difficult or undesirable to specify a query formally. The system currently does not communicate with the user through a visual interface, but instead uses a command-line interface and prints the results as a textual list.

The work by Cutting et al. (1991) is promising because it has better performance characteristics than other clustering methods. It could be used to let users specify relevant reference documents that could be used to define the inputs of the InfoCrystal (see Chapter 12).

The clustering methods described above, except for Scatter/Gather, provide users with an overview map of the information or document space, where the documents are clustered based on some similarity measure. These efforts make a valuable contribution because users can use them as a starting point in the search process. As we will discuss in chapter 12, these map displays and their clustering techniques could be integrated with the

InfoCrystal, where users could use them to help them identify the inputs that could be used to initialize the InfoCrystal. Further, these overview maps can facilitate information retrieval or database mining because they might reveal hidden regularities in the data that are hard to discover using individual queries. However, these overview have the following shortcomings, as summarized in the left-hand column, and in the right-hand column we indicate how the InfoCrystal addresses them :



- **Approximate representation:**  
The overview maps give users only an approximate sense of the structure of an information space, because higher-dimensional information is mapped into a lower-dimensional space.
- **Difficult to represent relationships among multiple variables:**  
For example, where do users find the documents in the above semantic map that are related to Information Retrieval (IR), Visualization (VI) and Query Languages (QL)?
- **Document-space centered:**  
The documents are displayed based on their similarities and the way they cluster without considering the interests of the user.
- The InfoCrystal imposes a structure on how an information space is visualized so that each icon has a precise and straightforward interpretation.
- The InfoCrystal represents *all* the possible relationships among the inputs.
- **Interest-centered representation:**  
the InfoCrystal represents the document space with respect to the stated interests of a user.

- **Scaling Issue.**  
The overview maps become progressively more difficult to interpret as the size of the document space increases.
- **No visual query language.**  
The overview displays can not be used to formulate queries graphically.
- The discrete version of the InfoCrystal scales well, because it emphasizes relationships instead of individual documents.
- The InfoCrystal is both a *visualization tool* as well as a *visual query language*.

## 10.2 Visualizing Hierarchical Structures

In this section we mention several approaches that address the problem of how large hierarchical structures such as file directories or corporate organizational structures could be visualized.

**Tree-Map:** Johnson and Shneiderman (1991) have developed an approach, called the *Tree-Map* technique, for visualizing hierarchically structured information such as file directories. Their method maps the full hierarchy of directories and files onto a rectangular region in a space-filling manner.

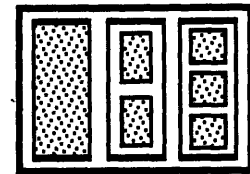


Figure 10.6: Tree-Map.

They make use of Venn diagrams to visualize directories and the files and subdirectories they contain, and they use a "slice and dice" approach to tessellate the rectangular region. This approach shares similarities with a "Russian doll", with the difference that a larger doll can contain more than one smaller doll.

- We can use this approach to visualize the hierarchical structure of a complex InfoCrystal query in compact and space efficient way. We plan to develop a tool, called *query overview*, that provides users with a quick insight into the query structure, thus making it possible for them to quickly locate an InfoCrystal that needs to be reprogrammed. Users can navigate through the structure by clicking on the rectangle representing the InfoCrystal of interest.
-



**Cone-Tree:** Card, Robertson and Mackinlay (1991) at Xerox PARC have developed 3D visualization and interactive animation techniques for exploring large information spaces by shifting some of the user's cognitive load to the human perceptual system. They have developed the *Information Visualizer* that allows a user to explore large information structures. The problem is that when the user moves their attention to another part of the structure, the instantaneous change from the old to new part can be disorienting. Their system animates the change of view, preventing this disorientation and giving the user a sense of embodied navigation with the structure.

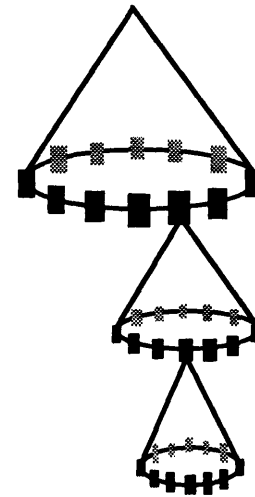


Figure 10.7: Cone Tree.

As part of the *Information Visualizer*, they have developed an information visualization technique, called the *Cone Tree*, which is used for visualizing hierarchical information structures. The hierarchy is presented in 3D to increase effective use of the available screen space and to enable the visualization of the whole structure. The top of the hierarchy is the apex of a cone with its children placed evenly spaced along its base. The next layer of nodes is drawn below the first, with their children in cones. When a node is selected, the *Cone-Tree* rotates so that the selected node and each node in the path from the selected node up to the top are brought to the front and highlighted. The Visualizer presents as much contextual information as possible and provides useful abstractions of the data and its structure. The 3D perspective view of the *Cone-Tree* provides also a fisheye view of the information. A selected object appears brighter, closer and larger than other ones, both because of the 3D perspective view and because of coloring and simulated lighting. Interactive animation is used to shift some of the user's cognitive load to the human perceptual system. The animation allows the user to track rotations, and when it is completed, no time is needed for reassimilation, because the perceptual phenomena of object constancy.

- The *Cone Tree* representation could be used to visualize the hierarchical InfoCrystal query structures in a three-dimensional form. The work by Card, Robertson and Mackinlay at Xerox PARC demonstrates convincingly

that interactive animation techniques are effective for accessing large information spaces, because they shift some of the user's cognitive load to the human perceptual system.

**Fisheye View:** This representation offers users an overview of a large information structure, where the elements that are currently of greatest importance are clearly visible and the less important ones do not clutter the display. The Fisheye representation uses both the distance from the current point of interest and the a priori importance to the users to display hierarchical structures [Furnas 1986]. This approach has been extended to the graphical display of graphs by adding a visual worth variable [Sarkar and Brown 1992].

- We plan to explore the use of a fisheye transformation to emphasize interior icons satisfying certain requirements and to de-emphasize the others. The Network InfoCrystal already possesses a fisheye effect (see section 3.6.2).

### 10.3 Familiar Metaphors for Accessing Information

In this section we describe several familiar metaphors that can be used to help users access large information spaces. We also describe a visual metaphor that can be used to communicate to users how the contents of individual documents are related to the users' interests.

**House/Rooms/Objects Metaphor:** Pejtersen et al. (1993) have developed a retrieval interface for Danish libraries by performing a work domain analysis and employing the familiar metaphor of a House/Rooms/Objects to communicate organization and possible attributes of the documents. The interface is designed to support recognition-based navigation and browsing for information. They provide a number of activity spaces to support different cognitive processes and tasks. A semantic network of the information sources in fictional literature has been designed to match the user's goals and motivations. A multifaceted classification scheme for representing the book content at several levels of abstraction that corresponds to user's ways of asking for information has been developed for indexing the books. This user-oriented indexing results in a very tight relationship between the representational structure of the book content/links and users' queries and categorization of information. The content and structure of the interface were designed to match users' cognitive and perceptual capabilities during shifts among several retrieval strategies.

The icons have been designed to match the user population's cultural background and knowledge (e.g., globe = geographic location, clock = time, etc.). Only icons whose meaning could be perceived by naive users within at most two seconds have been used (using multiple choice association tests). To support signs for actions, icons were chosen in the form of metaphors having functional/action analogies to a familiar context. For example, users communicate their reading needs by interacting with objects on work desk in a room, where these objects represent the various dimensions of the information need (specify the features of interest, where these features belong to different categories). The icons function as command icons which allow the specific dimensions to be specified by direct manipulation. They claim that the navigational metaphor (HOUSE -> ROOMS -> OBJECTS ...) is a very efficient support in providing users with an understanding of the structure of

---

the hypermedia system.

In summary, Pejtersen et al. argue for information or hypermedia systems that support perception and action-based or recognition-based information processing. They propose that hypermedia interfaces should be based on an analysis of users' cognitive and perceptual characteristics. Semantic domain networks should be represented in interface displays as symbolic information referring to the semantic content of the nodes, but, at the same time, they should be represented as signs for action/link selection.

**Piles:** Rose et al. (1993) have designed an interface based the familiar 'pile' metaphor to support users in the casual organization of documents and to provide them with content awareness. Users can loosely organize documents by placing them in a pile via direct manipulation. By moving the cursor over the pile, users can flip through it, quickly viewing small representations of the contents of the documents.

Rose et al. also provide users with a method that automatically creates different piles of related documents, using a non-hierarchical clustering algorithm, where the descriptive keywords have been automatically extracted and are used to form a vector. They use a standard vector space approach as well as a non-hierarchical clustering algorithm to help users become more content and structure aware. Their method creates a certain number of clusters that are visualized using the familiar pile metaphor.

- We could offer an additional style for the interior icons to visualize the contents of the individual interior icons.

**TileBars:** Hearst (1994) argues that term distribution patterns in a retrieved document should be made visible and has developed for that purpose the *TileBars*. The patterns in a column of *TileBars* can

be quickly scanned and deciphered to help users make judgments about the potential relevance of the retrieved documents. The bars for each set of query terms are lined up vertically one next to the other. This produces a representation that simultaneously and compactly indicates relative

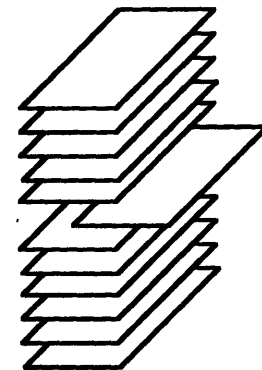


Figure 10.8: Piles.



Figure 10.9: TileBars.

---

document length, query term frequency, and term distribution across the identified distinct text segments. Term overlap and term distribution are both easy to compute and can be displayed in a manner in which both attributes together create recognizable patterns.

Hearst (1994) makes the argument that methods that use a similarity measure to determine the relevance of a document are appropriate for abstracts, because most of the terms in an abstract are salient for retrieval purposes, because they act as placeholders for multiple occurrences of those terms in the original text, and because these terms tend to reflect to the key topics in the text. However, it can be problematic to apply the similarity approach to full-length text documents because their structure is quite different from that of abstracts. Most long texts discuss several key topics simultaneously, hence two texts with one shared key topic will differ in their other key topics. As we have argued in chapter 5 and in Spoerri (1993), Hearst makes the point that a ranked list obscures the role that the query terms played in the ranking of the retrieved documents, whereas the goal should be to provide users with this type information in a form to permit swift interpretation.

#### **10.4 Visual Query Languages**

Many of the problems that users face when formulating queries can be overcome by offering them visual interfaces, where the evidence suggests that graphical ways of specifying a query are preferable for most kinds of queries (Bell and Rowe 1990). Several retrieval systems have recently been developed that use graphical interfaces that support users in the browsing, selection and retrieval of information [Fox et al. 1993, Kahle et al. 1993]. These interfaces use traditional and standard graphical representations, such as forms-based interfaces with structured fields, associated sliders and radio buttons, ranked lists, tables or scatter plots, to help users formulate queries and view the results. These systems attempt to make all the options clearly visible and to supply the syntax for the queries. These systems have been developed using "user-centered" design principles and user studies to guide the creation of the respective interfaces. Although these interfaces represent a great step forward to help users access information, they do not propose very innovative visual interfaces. A goal of this thesis has been to create a novel visual

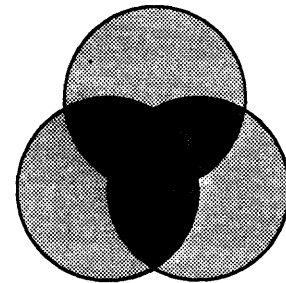
---

representation for accessing information that has been sufficiently mature and tested to warrant further work by a larger community of researchers to develop it further so that it will eventually become part of the mainstream of available visual metaphors.

In this section we mention several visual query languages that have been developed to help users to retrieve data from relational databases:

**Venn Diagrams:** Michard (1982) develops a graphical query language that is based on circular Venn diagrams and avoids the explicit use of parentheses and Boolean operations for set operations (Venn diagrams are widely used in schools to teach basic set operations and Boolean algebra). Each time a user specifies a criterion, a corresponding circle is drawn in a display area with a legend indicating the selection to which it

belongs. The user designates which elementary subsets to be selected by pointing on the desired portion of the Venn diagram of the intersecting circles. At most three criteria can be considered at the same time, because more than three intersecting circles can not represent all possible relationships between more than three criteria. To be able to create more complicated queries the user must use a "Memorize" function that causes the previous selected subsets to be represented by a single circle. This new subset can be then combined with at most two more new criteria. Michard conducted an experiment to compare this graphical query interface with a more traditional design. The results showed that the Venn diagram representation lead to less error-prone queries, where the statistically significant difference was mainly due to parentheses misuses. We eliminated this source of errors in our user studies by only accepting valid Boolean queries (see Chapter 8).



**Figure 10.10:** Venn Diagrams.

- The InfoCrystal moves beyond the Venn diagram approach so that more than three criteria can be represented at the same time. Further, the InfoCrystal is a more versatile and comprehensive interface than the one presented by Michard, because, for example, complex or weighted queries can be formulated more readily.
-

**Cougar:** Hearst (1994) has developed a browsing interface, called *Cougar*, for displaying multiple category information, using the familiar Venn diagram approach as employed by Michard (1982). It is strictly speaking not a full-fledged visual query language. However, it lets users view the retrieved documents based on the intersection of assigned categories. Users can search on either keyword or category information.

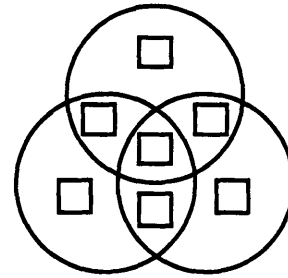


Figure 10.11: Cougar.

In *Cougar*, documents are assigned their three top-scoring categories from a pre-determined set that has been constructed using an automatic categorization algorithm. The documents are then indexed on the category information as well as on all lexical items from the title and the text body. Users issue queries by entering words or selecting categories from an available list. The most frequently occurring categories in the retrieved documents are displayed in a bank of color-coded buttons. The user can select up to three of these categories and see how the documents intersect with respect to those categories. Like the *InfoCrystal*, *Cougar* is designed to provide the needed tools to enable users to browse multi-dimensional spaces, because multiple categories or properties are associated with each document. The contribution of *Cougar* is the way users receive assistance in the selection of the reference points. However, its visual interface suffers from the same limitation as the traditional Venn diagram approach. The *InfoCrystal* was designed precisely to overcome this limitation so as to be able to visualize  $N$  and not just at most three categories at the same time. Furthermore, the *InfoCrystal* can visualize weighted and vector space queries.

**The Cube of Contents:** (Arents and Bogaerts 1993) uses the familiar three-dimensional cube to help users formulate queries by selecting values for up to three mutually exclusive attributes, but only one intersection of two of three attributes is visible at any time. This limitation illustrates one of the drawbacks of a 3D interface, where occlusions are unavoidable.

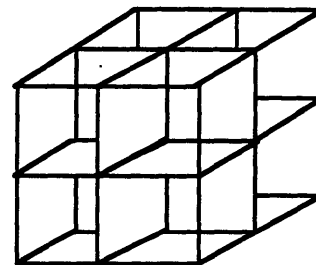


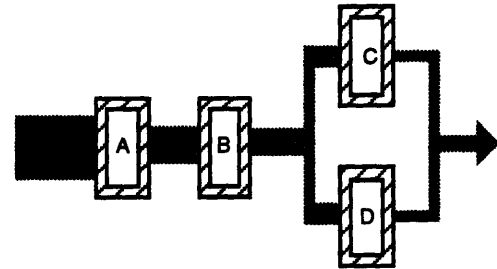
Figure 10.12: Cube of Contents.

This approach suffers from the same limitation as the interface based on the traditional Venn diagram approach because at most three concepts or dimensions can be considered simultaneously.

**Filter/Flow:** Young and Shneiderman (1992) have developed a graphical interface for specifying Boolean queries in a visual form that uses the metaphor of water flowing through filters. A similar representation is used in electrical engineering to depict electrical circuits of capacitors. The interface is designed for accessing a relational database.

The flow is left to right and the logical AND is visualized by requiring that the attribute menus are in same row but in a different column. The logical OR is represented by requiring that the attribute menus are placed in same column but in different rows. This interface is intended to alleviate some of the difficulties users have in specifying Boolean queries and an experiment was conducted to comparing the visual interface with a text-only SQL interface. There was a statistically significant difference in performance favoring the Filter/Flow interface, where the most frequent error type was the incorrect use of parentheses.

Anick et al. (1991) have developed a similar interface, called *tiles*, for specifying Boolean queries, where they use the vertical dimension to signal an OR and the vertical to indicate the AND operator.



A AND B AND ( C OR D )

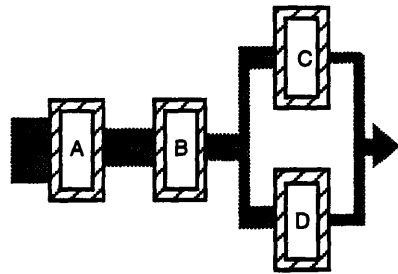
Figure 10.13: Filter/Flow.



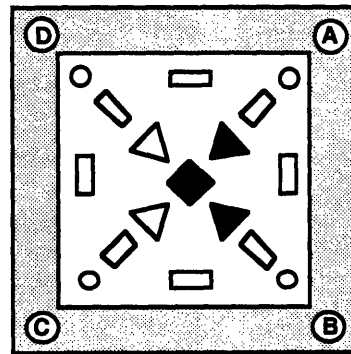
Existing visual query languages, including the ones mentioned above, suffer generally from the following limitation:

- **Can formulate only specific queries.** Existing query languages use visual primitives and a grammar that enables the formulation of a specific query. In order to generate a different query, users have to modify the way the current query has been organized.
- The InfoCrystal is a visual query language that enables users to formulate a whole range of related queries. It represents all the possible Boolean queries involving its inputs in normal disjunctive form.

The tile / filter flow query shown on the left would be formulated as follows in a InfoCrystal.



A AND B AND ( C OR D )



For a crystal with N inputs there are  $2^{2^N-1}$  possible queries that can be formulated by selecting the appropriate interior icons. Hence, a InfoCrystal represents a large universe of queries in a compact and accessible form.



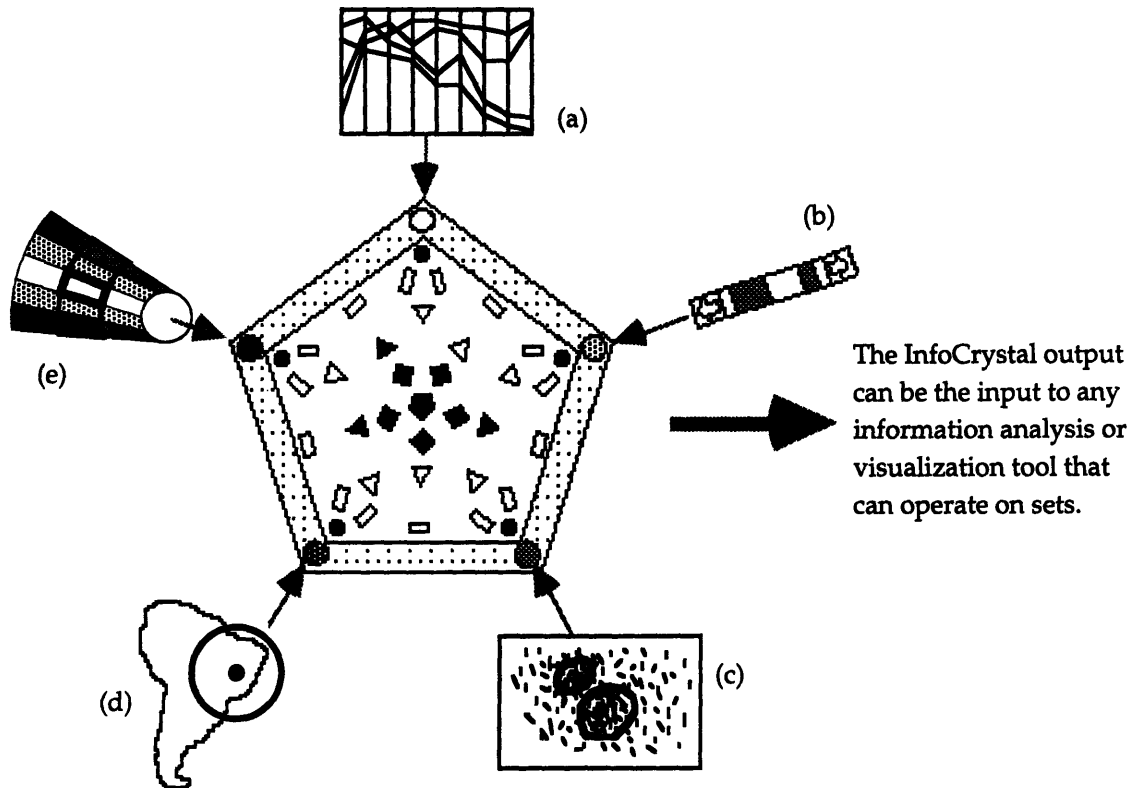
# CHAPTER 11

## APPLICATIONS

In this chapter we present a collection of brief scenarios of how the InfoCrystal representation could be applied in different domains. This collection is not intended to be exhaustive, but instead to stimulate the reader's mind and to demonstrate the versatility of the InfoCrystal.

The InfoCrystal is a flexible tool that enables users to compare arbitrary sets of data items. These sets can be fuzzy sets, where the degree of membership can be used in the visualization. The InfoCrystal can structure and cluster the information into discrete or continuous groupings to visualize how the information is related to several criteria. The InfoCrystal has applications as a visual Boolean Calculator in any domain where users need to coordinate several criteria. The InfoCrystal makes it easy for them to formulate and change their queries. Further, users can specify relevance weights and the InfoCrystal can visualize the resulting ranked output. The question at hand is: In which situations is the type of structure the InfoCrystal can visualize and are the types of operations it makes possible useful to users? We will begin to answer this question by listing some the data generators that could be used to define the inputs to the InfoCrystal. Next we will discuss how the InfoCrystal can be used as a general-purpose *coordinator* or *generator* of arbitrary data streams. Finally, we will provide brief scenarios how the InfoCrystal can be applied in the following ways and domains: Internet Exploration, Document & Information Retrieval, Database Mining, Multimedia Editing, Electronic Mail Filters, Hypertext Browser and Link Builder, Statistical Visualization, Visualizing the Power Set, Boolean Networks, and Neural Networks.

---



**Figure 11.1:** shows how the InfoCrystal can be used to coordinate diverse data generators to create a hybrid or heterogeneous data generator. The input data generators are: (a) Parallel Coordinates; (b) Slider; (c) Clustering Overview; (d) Map; (e) Input/Cone. The interior icons have been selected so that all the data that is generated by only one of the input generators is included in the output as well as data that is generated by at least four of the inputs. We have also selected some of the icons of rank three.

There are multiple ways to generate an InfoCrystal input, as long as the device, which is used, generates an ordinary or fuzzy set as its output:

- 1) *Parallel Coordinates* representation, which maps a point in a higher-dimensional space into a piece-wise linear curve in a two-dimensional display using parallel coordinates so as to not lose any information [Inselberg 1985]. Users can select a subset of these lines to define the output.
- 2) *Slider*, where users can define range(s) of values along a discrete or continuous data dimension.
- 3) *Clustering Overview*, where users can select multiple subsets of items to define the output.
- 4) *Map*, which can be used to specify a subset or a two-dimensional area of data points.
- 5) *Thesaurus or Classification Hierarchy*, where users can select the appropriate concepts at the desired level of specificity by interacting with an Input/Cone object (see Figure 12.1 for explanation).

Figure 11.1 shows the visual objects representing the data generators mentioned above. This list of data generators is not meant

to be exhaustive, but rather to give an indication of the types of generators can be used. Figure 11.1 further shows how the InfoCrystal can be employed as a general purpose *coordinator* of arbitrary information generators, and can act as a *generator* of diverse data streams. For example, we can use the InfoCrystal to combine and coordinate data streams containing diverse data types. If the data sets generated by the inputs do not overlap a great deal, then the InfoCrystal will clearly visualize this by having the data cluster away from the center.

We now will provide brief scenarios how the InfoCrystal can be applied in the following ways and domains:

- **Internet Exploration:** The InfoCrystal could be used to explore the resources available across the Internet. Users could generate a structured list of interests by interacting with an index, a thesaurus, or a semantic net. The leaf nodes of this structured list could be automatically paired with the appropriate retrieval methods and databases. Users could also be asked to specify the degree of coverage they require and the computational resources they are willing to allocate. The InfoCrystal software has been designed so that the retrieved information can be incorporated and propagated through the query structure on a continuous basis. The InfoCrystal could be used to complement the popular Mosaic interface that uses hard coded thematic listings or links to help users navigate the resources on the Internet.

Mosaic exemplifies a search paradigm, where users follow pointers to **go to the information** of potential interest, based on their accumulated knowledge of where they can find what. However, it does not represent a very powerful way of searching for information and it suffers from several major drawbacks: 1) If users do not know where to find a certain topic, then they have to crawl through the hyperlink structure using a hit and miss approach. 2) Links to relevant material may not yet have been established, or the links may be out of date. 3) Different users have different understandings of how topics are related and hence should be linked. 4) It uses a not very powerful, but appealing retrieval model by employing ad-hoc, fixed, but incomplete associations to link relevant data.

The InfoCrystal is an example of a retrieval paradigm where the

---

**information comes to the users.** They specify a list of their interests and are then presented with a global overview of how the information relates in a detailed way to these interests. At any point users can reconfigure the criteria used to **attract the information.** Furthermore, they can use a multitude of retrieval methods to probe a huge information space such as the Internet. As we have outlined in the introduction, the InfoCrystal represents a high-level retrieval interface that is flexible in terms of the retrieval methods and the data types used. It encourages exploration and the creation of abstractions. It supports "what-if" scenarios and it provides users with dynamic visual feedback. The InfoCrystal requires greater computational resources than the current Mosaic version, but it has greater retrieval power. Users are familiar with thematic headings, such as politics, sports, arts, etc., as a way to structure information. These headings can be used in the InfoCrystal to define the concepts of interest.

The Internet holds the promise that users are able to explore complex relationships between different fields of knowledge and information, where some of the interrelationships can not be foreseen in advance. We are moving into an increasingly interdisciplinary world, where diverse, at first unconnected areas of knowledge need to be related. The Internet offers the opportunity to become the repository for these diverse and rich bodies of knowledge. Hence, we need powerful visual tools that enable users to perform complex data explorations to discover meaningful new connections and opportunities. It is our hope that the InfoCrystal and its design principles represent a step in the desired direction.

- **Document & Information Retrieval:** It is important to distinguish between Document Retrieval and Data Retrieval. The retrieval of documents requires different tools than the search for data, because documents contain contextual and structural information that needs to be considered to be effective. Hence, we have developed additional visual tools to formulate and represent stemming, field and proximity specifications, which are of great value in text retrieval, employing a simple metaphor to visualize the resulting broadness of a query.

With respect to Boolean coordination, retrieval specialists often suggest to searchers to generate queries, where quasi-synonymous words for each conceptual factor are ORed and these different synonym lists are

---

then ANDed [Cooper 1988, Marcus 1991 and 1994]. Our default selection of the interior icons generates a query that is equivalent to the one suggested by retrieval specialists. A key advantage of the InfoCrystal is that it not only shows this query, but also other related, potentially useful queries.

In the context of retrieving text documents, the NOT operator does not necessarily have a straightforward interpretation and is not as commonly used, unless to indicate the exclusion of the previously or already retrieved documents. In a certain sense the InfoCrystal makes it possible for searchers to formulate queries that they rarely use in document retrieval. Hence, the InfoCrystal could be further customized by offering explicitly the types of queries that are especially effective in document retrieval (i.e., ANDs of ORs), and "suppressing", at least initially, the remaining possible queries that can overwhelm users.

Searchers often do little or no conceptual analysis of their query, especially of any formal kind. This is where InfoCrystal could be a big help [Marcus (personal communication)]. The InfoCrystal could play a useful role as a tool for teaching the basic concepts of modern Boolean retrieval. In particular, it could also be used by search specialists in libraries to communicate with their clients and device the appropriate search strategy.

One of the advantages of the InfoCrystal is that users can explore many different, but related ways of retrieving the information without having to modify the framework of a query. They can perform a "what-if" analysis by changing and observing how the retrieved information is propagated through the InfoCrystal query structure. Users can use a diverse set of retrieval methods to initialize a query structure. For example, at any point in the search process users can switch from a keyword-based to a full-text retrieval approach by replacing an input criterion with a particular document that better captures a specific interest. The InfoCrystal enables users to explore an information space without having to abandon their sense of overview. It provides users with a compact visual representation of how the retrieved documents relate to their specific interests. Such visual feedback helps users decide how to proceed in the search process.

- **Relational Databases:** The InfoCrystal can be used as a Boolean Calculator, and is therefore ideally suited for specifying the required combinations of conditions that records in a relational database need to satisfy. In the
-

context of relational databases, the use of the NOT operation is more appropriate and frequent than in document retrieval (see Document & Information Retrieval bullet).

- **Database Mining:** The InfoCrystal could become a useful element in the toolbox that is needed to "excavate nuggets of value" possibly contained in large databases. The crystal could perform the function of a radar screen that provides users with a compact and structured overview of how the data is related to a multitude of criteria. The InfoCrystal can be easily integrated with other data analysis tools, as we have indicated in other parts of this thesis.
  - **Financial Portfolio Management:** A portfolio manager can formulate an array of criteria to be used to evaluate stocks. These criteria can then be grouped and arranged in a hierarchical structure. The manager will place the necessary criteria that a stock needs to satisfy at the bottom of the hierarchy. The created hierarchical structure acts like a filter that progressively refines the selection of stocks to be considered. The advantage of the InfoCrystal is that the portfolio manager can easily narrow or widen the selection by interacting with a direct manipulation interface. Furthermore, there is no limit on how the criteria are defined: it could be a simple property that needs to be computed or it could involve a complex computation.
  - **Human Resource & Workteam Management:** If a manager needs to assemble a new workteam, then the InfoCrystal allows the manager to see how the skills of his workforce distribute across the space defined by the needed skills. The interface allows the manager to see the workforce in a new light and to become aware of people with interdisciplinary skills. It could also enable workers within a company to identify and find needed resources within their own organization.
  - **Multimedia Editing:** When editing a film or video sequence, editors often face the problem of how to retrieve the appropriate segment that satisfies a certain combination of requirements at an edit point and also supports the desired overall mood. Editing involves the art of compromise and the juggling of multiple criteria to arrive at a solution that optimizes the different requirements and leads to overall pleasing sequence. Editors
-



often end up using a segment that does not satisfy all the criteria, but using one that satisfies one or two of the needed characteristics especially well and that supports the desired overall mood. Editing can involve a lot of trial and error, where editors need to try different possibilities, where this very fluid process can lead them to change their mind about what is needed to make a particular transition work. The InfoCrystal can provide editors with a versatile **palette** that shows them not only the next best possible shots, but a whole range of segments and how they relate to their requirements that they could use to "paint" the next segment. Editors can specify the degree of importance they assign to the different criteria by interacting with the weight sliders, then the InfoCrystal provides them with a ranked output of the possible shots.

In the multimedia context it seems also appropriate that users could interact with an iconic interface to specify their requirements. The MediaStreamer is an example of a rich visual interface that enables users to annotate video clips by interacting with a visual taxonomy of video events [Davis 1993]. We could easily build on this representation and use it to specify search criteria. In the current InfoCrystal implementation, all the atomic or leaf nodes are represented as circles and hence look all the same, although they represent different content. Hence, it could be beneficial if the criterion icons could reflect their content in a visual way and not just the structure or form of the query. Users could interact with a library of icons that represent the available data generators that they could select-drag-drop in the corners of the InfoCrystal's border area to specify an input. If the data generator has an iconic representation of the appropriate size, then it could be even displayed instead of the generic circular criterion icon. If the data generator operates on a hierarchical structure, such as ACM classification system or another taxonomy, and its elements have iconic representation, as is the case in the MediaStreamer for example, then the criterion icons could be replaced by the icon representing the instance currently selected in the classification hierarchy.

- **Electronic Mail Filters:** The InfoCrystal could be used to filter and organize electronic mail. The crystal shows users how the received mail messages relate to their stated interests, and they could use it to help them decide which mail messages to read first, namely the ones that have at least a
-

certain relevance score. Once users have programmed the InfoCrystal by selecting the relationships of interests or setting the weights and the threshold, they would be presented with a ranked list of the crystal's output. At any point users could view the InfoCrystal to show them which types of messages they are ignoring and how well their rules, represented in terms of the chosen concepts and the selected relationships, are performing.

- **Hypertext Browser and Link Builder:** The InfoCrystal could be used in the following ways in the context of a hypertext system: 1) to select those existing links that satisfy certain combinations of criteria; 2) to generate new links by retrieving those documents or passages that satisfy criteria supplied by the user or that have been generated by performing a cluster analysis on a selection of text (fragments) that are of interest to a user. The InfoCrystal can offer users not just a single link to follow, but a whole structured space of links to explore.
  - **Statistical Visualization:** The InfoCrystal can be used to visualize the higher-level correlations or interactions among different input variables. Further, it can be used to visualize all the effects computed in a complete  $2^N$  factorial design. The statistical model of such design with  $N$  factors each at two levels includes  $N$  main effects,  $N(N-1)$  two-factor interactions, ... , and one  $N$ -factor interaction. Hence, the complete model of the  $2^N$  factorial design contains  $2^N - 1$  effects, which can all be represented in an InfoCrystal.
  - The InfoCrystal represents a general tool for **visualizing the combinatorics** of the possible relationships between several concepts, and users can assign weights to the concepts. This fact has implication in the following domains:
    - **Visualizing the Power Set:** The collection of all the possible disjoint subsets among  $N$  sets is commonly referred to as the power set  $2^N$ . The InfoCrystal visualizes the power set, with the exception of the relationship that does not involve any of the sets, which is equivalent to the empty set in this context. Hence, the InfoCrystal could have applications anywhere where the power set plays an important role
-

and needs to be visualized. For example, there is the Dempster-Shafer theory of evidence that addresses the problem of how to represent and manipulate the degrees of support provided by different sources of evidence to a set of  $N$  propositions. In contrast to a standard Bayesian design, in which degrees of belief are assigned to the  $N$  elements directly, the Dempster-Shafer model assigns degrees of belief to members of the power set  $2^N$  [Schocken & Hummel 1992].

- **Boolean Networks:** The interior icons of an InfoCrystal correspond to the elements used by Boolean Networks to model the learning and evolutionary processes in nature [Kauffman 1993]. This fact has further implications, because it suggests how relevance feedback could be used to select the interior icons. Learning mechanisms could be used to assist users in the selection of the interior icons. This can take the form where learning agents use the received relevance feedback by users to compute the selection status of an interior icon.
  - **Neural Networks:** The InfoCrystal in continuous mode can be used to visualize which combinations of the input values will trigger a cell to fire, based on the current settings of the threshold and the input weights.
-



# CHAPTER 12

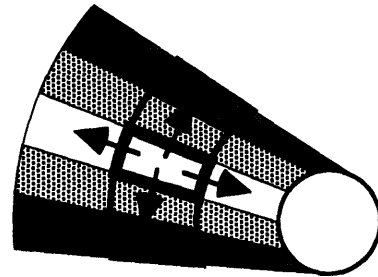
## FUTURE RESEARCH

In this chapter we list the issues concerning the InfoCrystal that we would like to address in the near future. The chapter that discusses some of the possible applications of the InfoCrystal also touches on the question of how the InfoCrystal might evolve in the future.

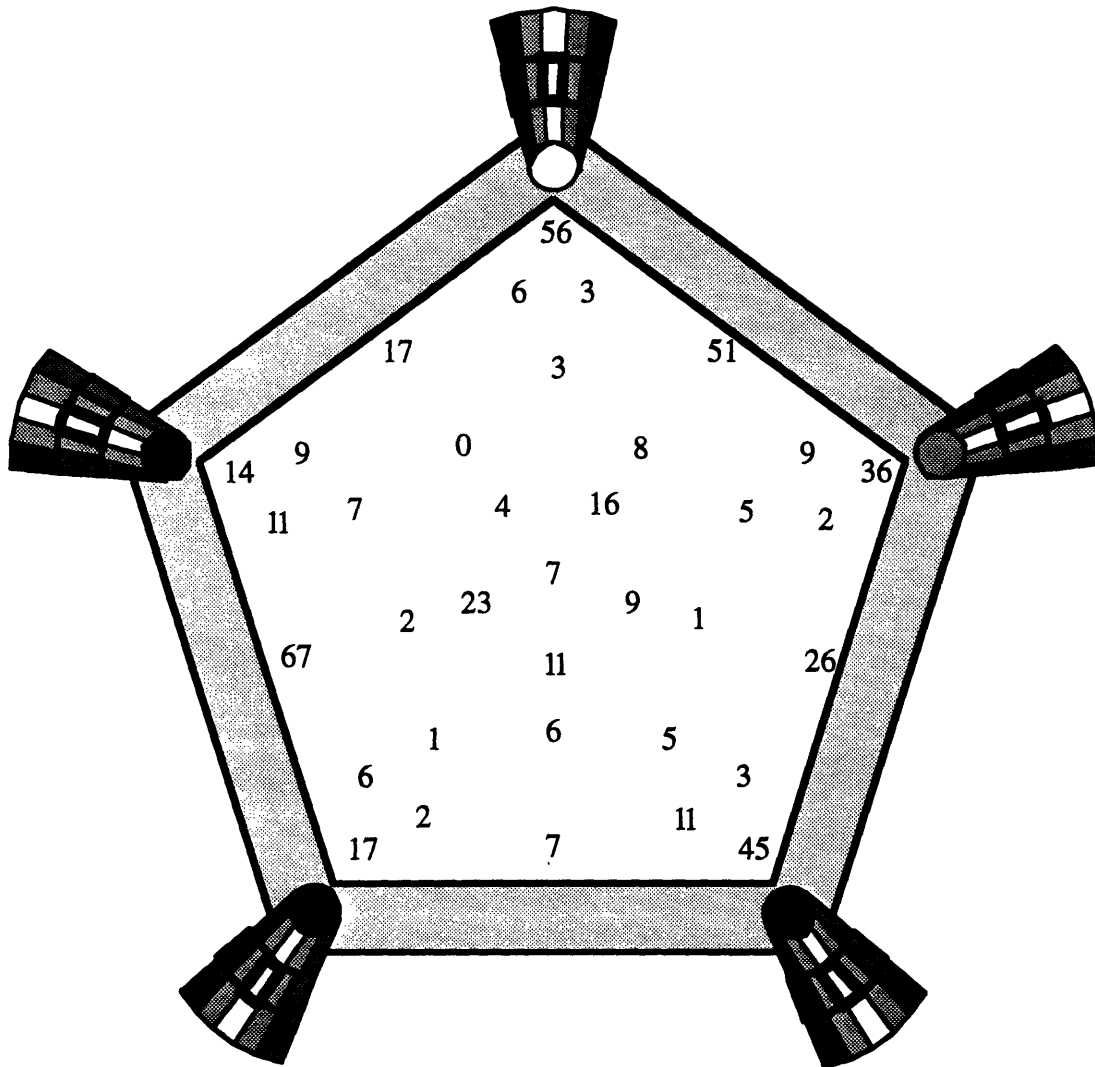
As we have already mentioned, we have implemented the InfoCrystal prototype on the Macintosh using the MacLISP programming language. We intend to reimplement the InfoCrystal using a more common programming language and faster platform to be able to interface more easily with a diverse set of retrieval methods, where the InfoCrystal can act as the common interface. We are planning to address the following issues in the near future:

- **How to generate the search concepts ?** We have mentioned that users are faced with the "vocabulary problem", where this problem could be addressed as follows in a visual way:
    - Users could interact with a list of concepts used to index the documents.
    - We could provide users with a cone shaped visual object, called the *Input/Cone*, that enables them to interact with a hierarchical classification system or a thesaurus to define the concepts of the InfoCrystal. Users navigate through a thesaurus by moving from more general terms to more specific ones and they can get immediate feedback from the way the data distribution changes. The cone shape is intended to reinforce in a visual way that if users move in the direction where the cone gets narrower, then they are narrowing the query by choosing a more specific concept. We are in the process of implementing the Input/Cone visual representation (see Figure 12.1 and 12.2).
-

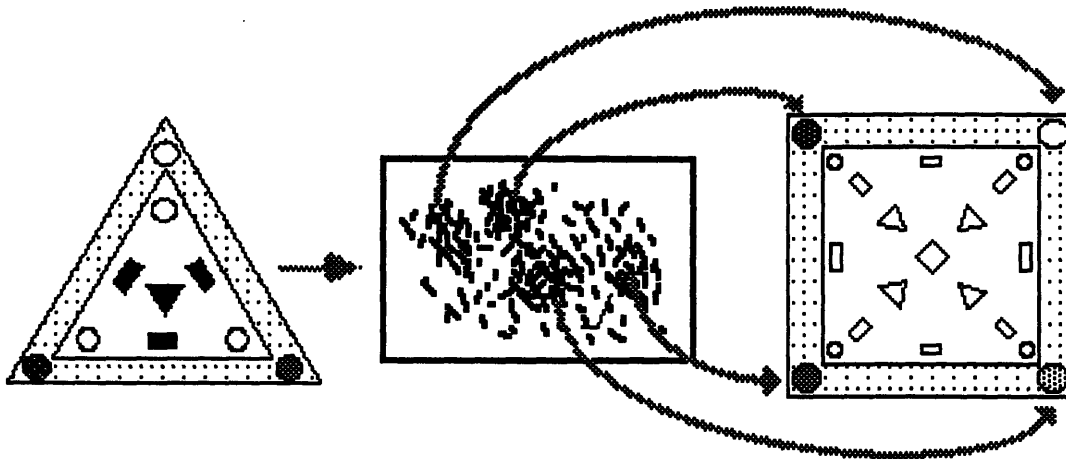
**Figure 12.1:** shows the *Input/Cone* that users can interact with to navigate through a hierarchical classification system to define the concepts of the InfoCrystal. The cone shape is intended to reinforce in a visual way that if users move in the direction where the cone gets narrower, then they are narrowing the query by choosing a more specific concept. The center field in the middle layer, shown with a thick black border, represents the currently selected concept. The center field in the outer layer represents the parent of the selected concept, and the inner layer shows some of the children of the selected concept.



If users click on any of the fields, then they select the concept associated with it and the *Input/Cone* is updated to reflect this change. Although not shown here, there will be buttons at the edges of each cone layer so that users can view and possibly select concepts that are currently not visible. Users can interact with the *Input/Cone* to change the specificity of the input concepts. They can easily zoom in and out and observe how the data distribution changes across the different relationships.



**Figure 12.2:** shows how the *Input/Cone* can be used to define the inputs to a five-concept InfoCrystal. The interior icons are displayed using numerical style to show how the retrieved documents across the 31 different relationships.



**Figure 12.3:** displays how users could use a three-concept InfoCrystal, for example, to generate an initial set by selecting only those relationships that satisfy at least two of the three concepts. Next users could perform a clustering analysis on the output of the InfoCrystal. They can use the centroids of the four identified clusters to define the reference concepts for a new, four-concept InfoCrystal.

- We could also use clustering techniques to help users identify concepts of interest or the dimensions along which to explore an information space. A user could retrieve an initial set, apply a clustering technique, such as the Scatter/Gather method developed by Cutting et al. (1991), to this set to identify  $N$  principal components, where the number  $N$  can be specified by the user. These identified concepts can then be used to create an InfoCrystal with  $N$  inputs (see Figure 12.3). Users could progressively refine their concept selections by using this automatic identification of the reference dimensions.
- **Icon Library of Retrieval Engines and Data Generators:** The leaf or atomic nodes of the query structure represent the locations where we interface with external information sources. The items, which are retrieved based on the instructions specified in the leaf nodes, are propagated through the query structure in a bottom-up fashion. We want users to be able to interact with a library of icons, which represent the available data generators, that they could select-drag-drop in the corners of the InfoCrystal's border area to specify an atomic input. Once the input has been defined, the appropriate window would appear asking the user to specify the needed settings.

- **3-D visualization:** We want to explore how we could use 3D computer graphics techniques to enhance the InfoCrystal: 1) We could place the corners and the criterion icons of an InfoCrystal in the third dimension to reflect the input weights. 2) The interior icons could be placed in the third dimension to reflect their relevance scores. 3) There are already now multiple layers of information associated with the interior of an InfoCrystal, and in the future we could imagine that further information could be added. For example, we could enable users to interact with these different levels of information by varying the degree of transparency between the layers or depth of focus.
- **How to Assist Users in the Programming of the InfoCrystal ?** A single InfoCrystal visualizes a large universe of feasible queries. How we can help users to explore this huge query space ? How can we help users converge quickly on a query that satisfies their information need ? In chapter 4 we have addressed how we can assist users in the translation of Boolean expressions into the InfoCrystal. We also want to develop methods that use other sources of information to determine the selection pattern of the interior icons. For example, we would like to integrate learning mechanisms to assist the user in the selection of the interior icons. This could take the form where learning agents use user relevance feedback to compute the selection status of the interior icons or to recompute the input weights and the threshold setting. Further, we would like to explore the possibility of enabling users to create macros that help them to explore the large query space visualized by an InfoCrystal.
- **Revealing the Complexity Gracefully:** As we have pointed out, the number of possible relationships between  $N$  concepts grows exponentially. The InfoCrystal offers users a structured overview of all these relationships, but their sheer number can be overwhelming. Users currently have the possibility to display the interior icons in different styles to emphasize the icons of interest. For example, users can choose from a list of common Boolean expressions to select a subset of the interior icons, where the not selected ones are displayed in the point style to not crowd the display.

We are interested in developing alternative ways of enabling users to

---



juggle many different concepts without becoming too overwhelmed by the resulting complexity. 1) We are considering a *fish-eye* transformation that emphasizes interior icons satisfying certain requirements and de-emphasizes the others. 2) We are interested in developing methods for displaying the different interior icons in stages and as the need arises ("just-in-time-display"). These methods could consider the way the user interacts with the InfoCrystal to determine how to display and "roll out" the interior icons. We have already implemented a method along these lines in the context of applying Boolean operations (see section 4.2.2 and Figure 4.2).

- **Support multiple data-visualization methods:** We want to build a software environment, where we can view the information using different visualization techniques, and where the output of one visualization can be the input to another one. For example, we would like to be able to visualize the InfoCrystal's output using Parallel Coordinates (PCs), select a subset of the items displayed in PCs and pass them on to another visual analysis tool.
  - We want to continue the development of our object-centered software environment to be able to use any data type as an input to the InfoCrystal and then have the appropriate retrieval method automatically called. For example, at any point in the search process we want users to be able to switch from a keyword-based to a full-text, Partial Matching approach, where they do not have to concern themselves that the appropriate retrieval engine is called. They could select a document from the ranked-list window of the InfoCrystal and drag-drop it in the location of the criterion icon that they wish to replace with this particular document, because it better captures a particular aspect of their information need.
  - One of our ultimate goals is to be able to test if the InfoCrystal interface leads to more effective retrieval as measured in terms of precision and recall. This one of the reasons why we want to reimplement the InfoCrystal on a more powerful platform than the Macintosh so that we can more easily tap into a rich array of retrieval engines that enable us to search large databases more quickly. We believe that one of the InfoCrystal's key advantages is that it is flexible in terms of the retrieval
-

methods that can be used and that it enables users to move seamlessly and quickly between them. Information retrieval is a highly interactive process, where users start out with one translation of their information need, which they modify as their search progresses and they are responding to the intermediary results. The InfoCrystal could be well suited to support users in the search process. We believe that the ability to explore large information spaces with a variety of powerful retrieval engines will show us how the InfoCrystal needs to evolve to become a truly effective retrieval interface.

# CHAPTER 13

## CONCLUSION

The *InfoCrystal*<sup>™</sup> is a powerful visual representation that uses a simple visual analogy to enable users to deal with some of the complexities involved in information retrieval. It is both a *visualization tool* and a *visual query language*. The InfoCrystal can visualize all the possible binary as well as continuous relationships among N concepts. In the binary case, it uses location, rank, shape, color and size coding to enable users to see in a single display how a large information space relates to their interests. In the continuous case, a novel polar representation has been presented that visualizes the relevance scores of the retrieved documents in a ranked order. The InfoCrystal represents all the possible Boolean queries involving its inputs in *disjunctive normal form*, which makes it very easy for users to modify a query. The InfoCrystal acts like a *Boolean Calculator* and users can use it to employ the expressive power of the Boolean retrieval approach and its broadening / narrowing techniques in a visual way. Users can assign relevance weights to the concepts of an InfoCrystal and formulate weighted queries by interacting with a threshold slider. The InfoCrystal has the added advantage that users can control in a visual way how to translate weighted queries into Boolean queries. Complex queries can be created by using the InfoCrystals as building blocks and organizing them in a hierarchical structure. Finally, the InfoCrystal has been generalized to visualize and formulate vector space queries. Hence, the InfoCrystal provides a visual framework that unifies the Exact and the Partial Matching approaches and enables users to take advantage of their respective strengths.

The results of a user study have been presented that compared the standard, text-based Boolean query language with the InfoCrystal in its most basic form. Subjects had to perform a recognition and generation task. The former asked users to recognize either the correct Boolean or InfoCrystal query from among three possible queries. The latter required subjects to

---

generate a Boolean or InfoCrystal query that captured a given information need. These two tasks only tested a specific aspect of the InfoCrystal interface. Hence, this user study did not have the scope to fully evaluate the effectiveness of the complete InfoCrystal representation, but it produced the following useful results: 1) Novice users were able to successfully use the InfoCrystal, although they received only a short, fifteen minutes long tutorial. This second version of the tutorial made a big difference how well and quickly users could learn to use the InfoCrystal. Further improvements in the way novice users are instructed to use the InfoCrystal will help them to make full use of its rich set of features and the advantages that it has to offer. 2) The user study showed that the InfoCrystal, even at an early stage of development, performed as well as the familiar Boolean interface, although the study was biased in favor of the Boolean mode. 3) On the one hand, the study confirmed that the InfoCrystal is ideally suited for queries of the form "at most, exactly, or at least  $n$  out of  $m$  features". On the other hand, it showed that certain Boolean queries are more difficult to formulate using the InfoCrystal than the Boolean interface. However, we believe that users can improve their performance with more practice and if they have access to the enhancing features of the InfoCrystal that have been implemented, but were not made available during the experiments. 4) The user feedback concerning the InfoCrystal interface was very encouraging and it helped to pinpoint possible improvements. The user study shows that the InfoCrystal, even in its most basic form, can be successfully used by novice users and hence warrants further development.

This thesis has addressed the difficult problem of how to visualize information spaces that are abstract and do not have explicit spatial properties that can be exploited. The InfoCrystal provides a *spatial overview* of the data elements in an large information space and *simultaneously* provides *visual cues about the content* of the data elements. These opposing requirements have been resolved by exploiting the grouping principles used by the human visual system to make relationships between different, but related data elements visible and immediate. The InfoCrystal does not lock users into just one way of viewing the data. It helps them to decide how to proceed in the search process and how to control the output because the quantitative information associated with an interior icon.

---

Further, the InfoCrystal is a visual representation that not only has *descriptive* power, because it enables users to see large amounts of information in a compact way, but that also has *expressive* power that enables users, for example, to interact with the data to issue retrieval commands. The InfoCrystal is a high-level retrieval interface because it encourages complex explorations and the creation of abstractions. It is flexible in terms of the retrieval methods and the data types used. The InfoCrystal supports "what-if" scenarios and it provides users with dynamic visual feedback.

The contribution of this thesis to the emerging field of Information Visualization consists of two parts. First, this thesis has demonstrated how information visualization offers ways to accomplish some of the needed improvements in information retrieval. It has provided a constructive proof that it is possible to visualize both Exact and Partial Matching methods in the same visual framework. Second, this thesis suggests directions for further research to develop the InfoCrystal into a tool that enables general users to make full use of its expressive power. The InfoCrystal has broad applications, because it offers a "visual machinery" to compare and relate any number of ordinary or fuzzy sets of arbitrary data items. It opens up new possibilities for complex data explorations. The InfoCrystal can be used as a general-purpose coordinator or generator of arbitrary data streams. It can be used as a Boolean Calculator in any domain where several criteria or requirements need to be coordinated. This thesis has briefly discussed how the InfoCrystal could be applied in domains such as: Internet Exploration, Document and Information Retrieval, Database Mining, Multimedia Editing. The Internet offers the opportunity to become the repository for diverse and rich bodies of knowledge. Hence, users need powerful visual tools that enable them to perform complex data explorations to discover meaningful new connections and opportunities. It is our hope that the InfoCrystal and its design principles represent a step in the desired direction. The overall goal of this thesis has been to contribute to the development of a Visual Retrieval Interface, where users can choose from a diverse set of visual tools to filter and visualize information.

---



## CHAPTER 14

### EPILOGUE

One of the key challenges facing the emerging field of information visualization is to provide users with a variety of visual abstractions and powerful ways of linking these representations. It also needs to enable users to visualize how they have interacted with the information. Each of these abstractions will have a semantics associated with it, because it enables users to create meaning or gain understanding that has certain defining features in terms how users interact with the data. The way users choose to visualize the information and the actions they perform are valuable sources of information that also need to be understood by the users. Hence, we need to find ways to abstract and visualize this meta-information. We envision a future where users will be able to create and interact with a hierarchy of visual abstractions, where a higher level reflects and summarizes the actions and semantics of the lower level. This vision is still very vague, but it could provide us with a "golden carrot" that will eventually lead us to productive discoveries.

The work presented in this thesis grew out of such a vague vision: we wanted to develop a visualization that would give us a compact view of how the contents of a library are related to our specified interests. We wanted to be able to juggle as many different interests as possible at the same and use visual grouping principles to display all their relationships in a compact way. We wanted a representation that allowed us to focus on specific relationships without forcing us to abandon our sense of overview. We wanted to develop a representation that would provide us with a global, yet locally detailed overview of the contents of the database.

What has been the outcome of this vague vision? What did we stumble over as we were busy chasing after this vision dangling in front of our mind's eye? We have developed a powerful visual representation that uses a simple visual analogy to enable users to deal with some of the complexities involved in information retrieval. We have presented a novel representation that not

---

only can be used to visualize arbitrarily complex Boolean queries or weighted queries, but also the vector-space retrieval approach and its related partial matching methods. We have developed a visual framework that can accommodate the major retrieval methods. Further, we have outlined how the InfoCrystal allows users to move effortlessly between different ways of accessing and viewing information.

We understand the InfoCrystal as a possible piece in the mosaic of emerging information visualization tools. The purpose of this thesis has been to demonstrate these capabilities in the form of a prototype and hopefully to have presented it in such a way to inspire others to build on the presented work.

We have stated that the way we process visual information has been an inspiration for creating the InfoCrystal. On the one hand, it only makes sense to turn to our understanding and evolving theories of the human visual system to give us insights into how to create effective visualization. It is precisely the human visual system that we are trying to engage in the process of gaining an understanding of large information spaces. This is the simple and straightforward motivation. On the other hand, there is a deeper reason to try to learn from the way we process visual information. The human visual system uses a multitude of representations to arrive at the perceptions of the physical world that we take for granted. There are bottom-up as well as top-down processes involved in this sense-making process, where these processes exploit the regularities of the physical world [Marr 1982]. It is our intention to suggest that there could be a parallel between the way the human visual system processes information and the type of information visualization environment we need to strive to create. Like our visual system, we need to provide users with a multitude of diverse representations and visual abstractions and with ways to link them that are powerful. The InfoCrystal represents one of these needed abstractions and it could be used to moderate the communication between different representations. To use an analogy, we want these links to be able to speak a language that goes beyond revealing correspondence (i.e., we want to move beyond highlighting corresponding points in the linked representations), but that is able to issue more complex commands that have the effect of transforming the way the linked representations are visualized.

---



There is a growing trend to hide from users the complexity of the methods used to accomplish a task. We have to strive to create visualization systems that find a balance between the ease with which the information can be consumed by our visual system and how to bring analytical and reflective cognitive processes to bear on the sense making process. Information visualization is meant to stimulate and guide our sense-making processes, without however to completely succumb to our established ways of perceiving the world. We are not in a habit to question what we see. On a deeper level, we rarely question what we believe. We are entering an age where our ability to make sense of large and diverse information spaces will become a key competitive advantage. The whole notion of database mining is to find the "eternal needle in the haystack". If we only knew what to look for or where to look ! Our expectations and beliefs play such an important role in how we perceive the world. How do we have to frame the information to be able to perceive potential valuable patterns ? The future seems to require from us to be able to shift our point of view at an increasingly rapid rate. We need to be able to change the context within which we explore information not just occasionally but on a constant basis. The constancy is change. The resulting challenges for information visualization are many-fold to support complex data explorations. First, we need to create visual abstractions that are easy for users to learn how to use. Second, we need to provide users with tools that enable them to create new visual abstractions that capture what they have learned. Third, we need to create visualizations that are able to learn from how users interact with them. We envision a self-organizing visualization environment, where users are integrated in the loop both as a source for learning and as participants that direct the learning process.

Information visualization faces the challenge of having to invent visual abstractions that capture the complexity of the information spaces that users are increasingly required to understand, using design principles that lead to simple and consistent tools. In short, one of the key tasks of information visualization is to increase the simplicity with which users can deal with increased complexity. This challenge is a reflection of the exciting opportunities for invention that the field of information visualization offers. I will be back.

---



## BIBLIOGRAPHY

- Anderson, O.** (1988) "A Universal Venn Diagram," *Mathematics and Computer Education*, 1988.
- Anick, P.; Brennan, J.; Flynn, R.; Hanssen, D.; Alvey, B. & Robbins, J.** (1990) "A Direct Manipulation Interface for Boolean Information Retrieval via Natural Language Query," *Proc. ACM SIGIR '90*.
- Anthony, M. & Biggs, N.** (1992) *Computational Learning Theory*, Cambridge Tracts in Theo. Computer Science 30, 1992.
- Arents, H. & Bogaerts, W.** (1993) "Concept-based Retrieval of Hypermedia Information from Term Indexing to Semantic Hyperindexing," *Information Processing & Management*, 29:3, 1993.
- Bell, J. & Rowe, L.** (1990) "Human Factors Evaluation of a Textual, Graphical, and Natural Language Query Interfaces," *Technical Report M90/12*, UC Berkeley ERL, 1990.
- Belkin, N. & Croft, B.** (1992) "Information Filtering and Information Retrieval: Two Sides of the Same Coin," *Communication of the ACM*, Dec., 1992.
- Belkin, N.; Cool, C.; Croft, W. & Callan, J.** (1993) "The Effect of Multiple Query Representations on Information Retrieval Systems Performance," *ACM-SIGIR '93*, Pittsburgh, June, 1993..
- Belkin, N.; Marchetti, P & Cool, C.** (1993) "BRAQUE: Design of an Interface to Support User Interaction in Information Retrieval," *Information Processing & Management*, 29:3, 1993.
- Bookstein, A.** (1985) "Probability and Fuzzy-Set Applications to Information Retrieval," in Williams, M, ed., *Annual Review of Information Science and Technology*, Vol. 20, 1985.
- Borgman, C.** (1989) "All Users of Information Retrieval Systems Are Not Created Equal: An Exploration of Individual Differences," *Information Processing & Management*, 25:3, 1989.
- Campagnoni, F. & Ehrlich, K.** (1989) "Information Retrieval using Hypertext-based Help System," *ACM Trans. on Inf. Systems*, 7:3, 1989.
-

- Card, S.; Robertson, G. & Mackinlay, J.** (1991) "The Information Visualizer, an Information Workspace," *Proc. CHI'91 Human Factors in Comp. Systems*, 1991.
- Chalmers, M. & Chitson, P.** (1992) "BEAD: Exploration in Information Visualization," *Proc. ACM Int. SIGIR '92 (Information Retrieval)*, 1992.
- Cooper, W.** (1988) "Getting Beyond Boole," *Information Processing & Management*, 24:3, 1988.
- Cutting, D.; Karger D., Pedersen J. & Tukey J.** (1992) "Scatter/Gather: A Cluster-based Approach to Browsing Large Document Collections," *Proc. ACM Int. SIGIR '92 (Information Retrieval)*, 1992.
- Davis, M.** (1993) "Media Streams: An Iconic Visual Language for Video Annotation," *Proc. IEEE Workshop on Visual Languages*, Aug., 1993.
- Davis, J.** (1994) "A Server for a Distributed Digital Technical Report Library," *Technical Report Computer Science Report 94-1418*, Cornell University.
- Faieta, B. & Lumer, E.** (1994) "Exploratory Database Analysis via Self-Organization," *Proc. RIAO'94: Intelligent Multimedia Information Retrieval Systems and Management*, Oct., 1994.
- Foltz, P. & Dumais, S.** (1992) "Personalized Information Delivery: An Analysis of Information Filtering Methods," *Communications of the ACM*, December, 1992.
- Fox, E.** (1983); *Extending the Boolean and Vector Space Models of Information Retrieval with P-Norm Queries and Multiple Concept Types*, Ph.D. thesis, Cornell University, 1983.
- Fox, E. & Sharat, S.** (1986); "A Comparison of Two Methods for Soft Boolean Interpretation in Information Retrieval," *Technical Report TR-86-1*, Virginia Tech, Department of Computer Science, 1986.
- Fox, E. & Koll, M.** (1988); "Partial Enhanced Boolean Retrieval: Experiments with the SMART and SIRE Systems," *Information Processing & Management*, 24:3, 1988.
- Fox, E.; Betrabet, S.; Koushik, M. & Lee, W.** (1992) "Extended Boolean Models," in *Information Retrieval: Data Structures & Algorithms* (Ed.: Frakes & Baeza-Yates) 1992.
- Fox, E.; Hix, D.; Nowell, L.; Brueni, D., Wake, W. & Heath, L.** (1993); "Users, User Interfaces, and Objects: Envision, a Digital Library"; *Journal of American Society for Information Science*, 44 (8), p. 480 - 491, 1993.
-

- 
- Frakes, W. & Baeza-Yates, R. (ed.), (1992) *Information Retrieval: Data Structures & Algorithms*. Prentice Hall, 1992
- Frei, H-P. & Qiu, Y. (1993) "Effectiveness of Weighted Searching in an Operational IR Environment" *Proc. der GI-Fachtagung Information Retrieval*, 1993.
- Furnas, G.; Landauer, T.; Gomez, L. & Dumais, S. (1983) "Statistical Semantics: Analysis of the Potential Performance of Keyword Information Systems," *Bell Syst. Tech. J.*, 1983, 62, 1753-1806.
- Furnas, G.; Landauer, T.; Gomez, L. & Dumais, S. (1987) "The Vocabulary Problem in Human-System Communication," *Communications of the ACM*, 30(11), 964-971, 1987.
- Furnas, G. (1986) "Generalized Fisheye Views," *Proc. CHI'86 Human Factors in Comp. Systems*, 1986.
- Gloor, P. (1991) "Cybermap, Yet Another Way of Navigation in Hyperspace," *Proc. ACM Hypertext '91*.
- Harman, D. (1992) "Ranking Algorithms," in *Information Retrieval: Data Structures & Algorithms* (Ed.: Frakes & Baeza-Yates) 1992.
- Hearst, M. (1994) *Context and Structure in Automatic Fulltext Information Access*, Ph.D. thesis, University of California at Berkeley, 1994. (Computer Science Division Technical Report)
- Hearst, M. (1994) "Using Categories to Provide Context for Fulltext Retrieval Results," *Proc. RIAO'94: Intelligent Multimedia Information Retrieval Systems and Management*, Oct., 1994.
- Hemmje, M.; Kunkel, C. & Willett, A. (1994) "LyberWorld - A Visualization User Interface Supporting Fulltext Retrieval," *Proc. ACM SIGIR '94*.
- Humphries, M. (1987) "Venn Diagrams Using Convex Sets," *Mathematical Gazette*, 71, March 1987.
- Inselberg, A. (1985) "The Plane with Parallel Coordinates," Special Issue on Computational Geometry, *The Visual Computer*, 1, 1985.
- Kahle, B.; Morris, M.; Goldman, J.; Erickson, T. & Curran, J. (1993), "Interfaces for Distributed Systems of Information Servers", *Journal of American Society for Information Science*, 44 (8), p. 453 - 467, 1993.
- Kauffman, S. (1993), *The Origins of Order*, Oxford University Press, 1993.
-

- Korfhage, R. (1991) "To See, or Not to See - Is that the Query?," *Proc. ACM Int. SIGIR '91 (Information Retrieval)*, 1991.
- Korfhage, R. & Olson, K. (1991) "Information display: Control of visual representations," *Proc. IEEE Workshop on Visual Languages*, Oct., 1991.
- Kuhlthau, C.; Turock, B.; George, M. & Belvin, R. (1990) "Validating a model of the search process: a comparison of academic, public and school library users," *Library & Information Science Research*, 12:1, 1990.
- Kuratowski, K. & Mostowski, A. (1976) *Set Theory: with an introduction to descriptive set theory*, North-Holland Publishing Company, 1976.
- Lancaster, F. & Warner, A. (1993) *Information Retrieval Today*. Information Resources, Arlington VA, 1993.
- Liddy, E.; Paik, W.; Yu, E. & McKenna, M. (1994) "Document Retrieval Using Linguistic Knowledge," *Proc. RIAO'94: Intelligent Multimedia Information Retrieval Systems and Management*, Oct., 1994.
- Lin, X.; Soergel, D. & Marchionini, G. (1991) "A Self-organizing Semantic Map for Information Retrieval," *Proc. ACM SIGIR '91*.
- Marchionini, G. (1992); "Interfaces of End-User Information Seeking", *Journal of American Society for Information Science*, 43 (2), p. 156-163, 1992.
- Marcus, R. (1983); "An Experimental Comparison of the Effectiveness of Computers and Humans as Search Intermediaries", *Journal of American Society for Information Science*, 34, p. 381-404, 1983.
- Marcus, R. (1991) "Computer and Human Understanding in Intelligent Retrieval Assistance," *American Society for Information Science*, 28, 1991.
- Marcus, R. (1994) "Intelligent Assistance for Document Retrieval Based on Contextual, Structural, Interactive Boolean Models," *Proc. RIAO'94: Intelligent Multimedia Information Retrieval Systems and Management*, Oct., 1994.
- Marr, D. (1982) *Vision*, W. H. Freeman Company, 1991.
- Michard, A. (1982) "Graphical Presentation of Boolean Expressions in a Database Query Language: Design Notes and an Ergonomic Evaluation," *Behavior and Information Technology*, 1:3, 1982.
- Montgomery, D. (1991) *Design and Analysis of Experiments*, John Wiley & Sons, 1991.
-

- Pejtersen, A.** (1993) "Designing Hypermedia Representations from Work Domain Analysis," in *Hypermedia* (Ed.: Frei & Schäuble) 1993.
- Pinker, S.** (1990), "A theory of graph comprehension," in Freedle, R. (ed.), *Artificial Intelligence and the Future of Testing*, Lawrence Erlbaum Associates, 1990.
- Radecki, T.** (1988) "Trends in Research on Information Retrieval - The Potential for Improvements in Conventional Boolean Retrieval Systems," *Information Processing & Management*, 24:3, 1988.
- Robertson, G.; Mackinlay, J. & Card, S.** (1991) "Cone Trees: Animated 3D Visualizations of Hierarchical Information," *Proc. CHI'91 Human Factors in Comp. Systems*, 1991.
- Rose, D.; Mander, R.; Oren, T., Ponceleon, D., Salomon, G. & Wong, Y.** (1993) "Content Awareness in a File System Interface: Implementing the 'Pile' Metaphor for Organizing Information", *ACM-SIGIR'93*, Pittsburgh, June, 1993.
- Rosenblum, L.** (editor) (1994) *Scientific Visualization: Advances and Challenges*, Academic Press, 1994.
- Salton, G. & McGill, M.** (1983) *Introduction to Modern Information Retrieval*. McGraw-Hill, 1983.
- Salton, G.** (1988) "A Simple Blueprint for Automatic Boolean Query Processing," *Information Processing & Management*, 24:3, 1988.
- Salton, G. & Buckley, C.** (1988) "Term-Weighting Approaches in Automatic Text Retrieval," *Information Processing & Management*, 24:5, 1988.
- Salton, G. & Buckley, C.** (1990) "Improving Retrieval Performance by Relevance Feedback," *Journal of American Society for Information Science*, 41:4, 1990.
- Sarkar, M. & Brown, M.** (1992) "Graphical Fisheye View Graphs," *Proc. CHI'92 Human Factors in Comp. Systems*, 1992.
- Schocken, S. & Hummel, R.** (1992) "On the Use of the Dempster Shafer Model in Information Indexing and Retrieval Applications, " *Working Paper Series, Stern School of Business IS-92-27*, New York University, 1992.
- Spoerri, A. & Ullman, S.** (1987) "The Early Detection of Motion Boundaries," *Proc. Int. Conf. on Computer Vision*, 1987.
-

- Spoerri, A.** (1991) *The Early Detection of Motion Boundaries*. Master's Thesis, MIT Department of Brain & Cognitive Sciences, and MIT Artificial Intelligence Lab Technical Report 1275.
- Spoerri, A.** (1993a) "Visual Tools for Information Retrieval," *Proc. IEEE Workshop on Visual Languages*, 1993.
- Spoerri, A.** (1993b) "InfoCrystal: A Visual Tool for Information Retrieval & Management," *Proc. IEEE Visualization '91, and Proc. ACM Information & Knowledge Management*, 1993.
- Spoerri, A.** (1994) "InfoCrystal: Integrating Exact and Partial Matching Approaches Through Visualization," *Proc. RIAO'94: Intelligent Multimedia Information Retrieval Systems and Management*, Oct., 1994.
- Tufte, E.** (1983) *The Visual Display of Quantitative Information*. Graphics Press, Cheshire, 1983.
- Tufte, E.** (1990) *Envisioning Information*. Graphics Press, Cheshire, 1990.
- Turtle, H. & Croft, W.** (1991) "Efficient Probabilistic Inference for Text Retrieval," *Proc. RIAO'91*.
- Willet, P.** (1988) "Recent Trends in Hierarchical Document Clustering: a Critical Review", *Information Processing and Management*, p 577-597.
- Young, D. & Shneiderman, B.** (1993) "A Graphical Filter/Flow Representation of Boolean Queries: A Prototype Implementation and Evaluation, " *Journal of American Society for Information Science*, 44:6, 1993.
-



# APPENDIX 1

## TUTORIAL

In this appendix we show in detail the tutorial used to familiarize the subjects, who participated in the experimental evaluation, with the InfoCrystal. This tutorial can also serve as a way to show how Boolean queries can be visualized or formulated using the InfoCrystal. We assume for this tutorial that we have a computer database that stores records about a collection of paintings. Each painting's record stores the features of that painting, including what colors are present. We further assume that we are interested in finding records on paintings with certain combinations of the colors green, red and blue.

The tutorial starts with a basic explanation of the key visual components of the InfoCrystal that visualizes all the possible relationships among the three color concepts. Subjects are introduced to the Boolean meaning of the individual interior icons. Next subjects are shown what the resulting Boolean query will be if different collections of interior icons are selected. At several points during the tutorial, subjects are asked to answer questions to engage them and to test their understanding so far. The order of the examples has been chosen to help subjects build up an understanding of what the Boolean implications are as the number of selected interior icons increases. At any point subjects can return to a previously encountered example to help them understand what the exact implications are when a particular interior icon is selected or deselected.

The tutorial introduces the subjects to the most common Boolean queries and how they can be formulated with the InfoCrystal. After the subjects have viewed these specific examples, they are given the opportunity to interact with the InfoCrystal and select any subset of the interior icons. For an InfoCrystal with three concepts, there are 124 different Boolean queries that the subjects can specify. We will now show the screen designs used to introduce the experimental subjects to the Boolean and the InfoCrystal query language.

---

## The Purpose of this Experiment

To find records in a computer database you have to specify what features you want the records to have.

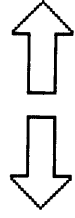
This experiment compares two ways of specifying the desired features or your information need:

- The **Boolean** query language.
- The **InfoCrystal** query language.



Let us suppose we have a computer database with listings or records about a collection of paintings. Each painting's record stores the features of that painting, including what colors are present.

Let us further suppose that we are interested in finding records on paintings with certain combinations of the colors green, red and blue.



## Boolean Query Language

uses the Boolean operators AND, OR, NOT to combine the features of interest to form a query.

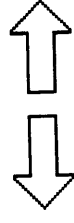
If we form a query (*Green AND Red*) then we are only interested in paintings where **both** these colors appear.

If (*Green OR Red*) then we require that **at least one** of these colors occurs in a painting.

If (*NOT Green*) then we only want paintings where this color is **not** present.

More complex queries can be formed by nesting (combining with parentheses) simple Boolean queries:

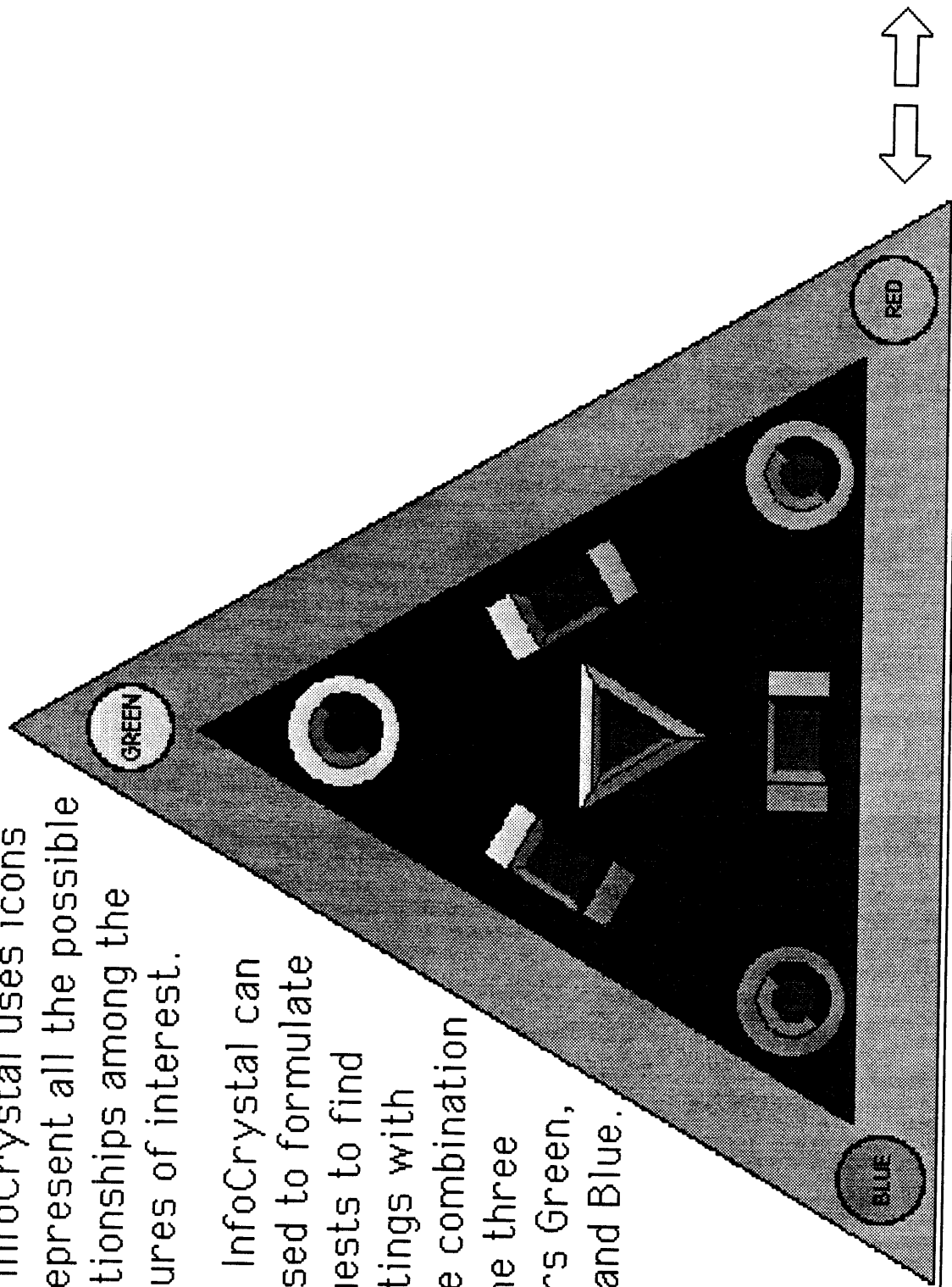
*(Green AND (Red OR Blue))*



## InfoCrystal Query Language

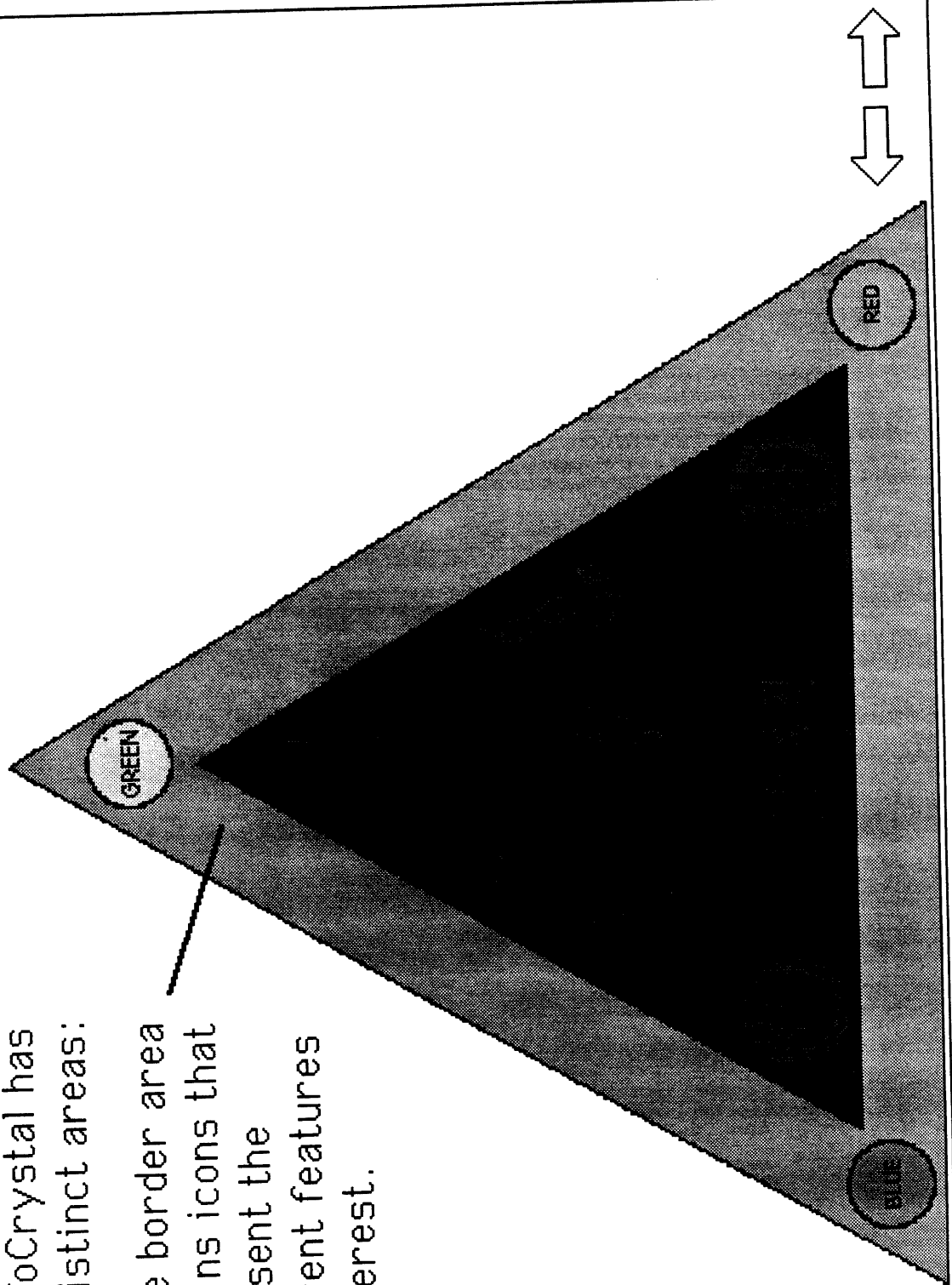
The InfoCrystal uses icons to represent all the possible relationships among the features of interest.

This InfoCrystal can be used to formulate requests to find paintings with some combination of the three colors Green, Red and Blue.



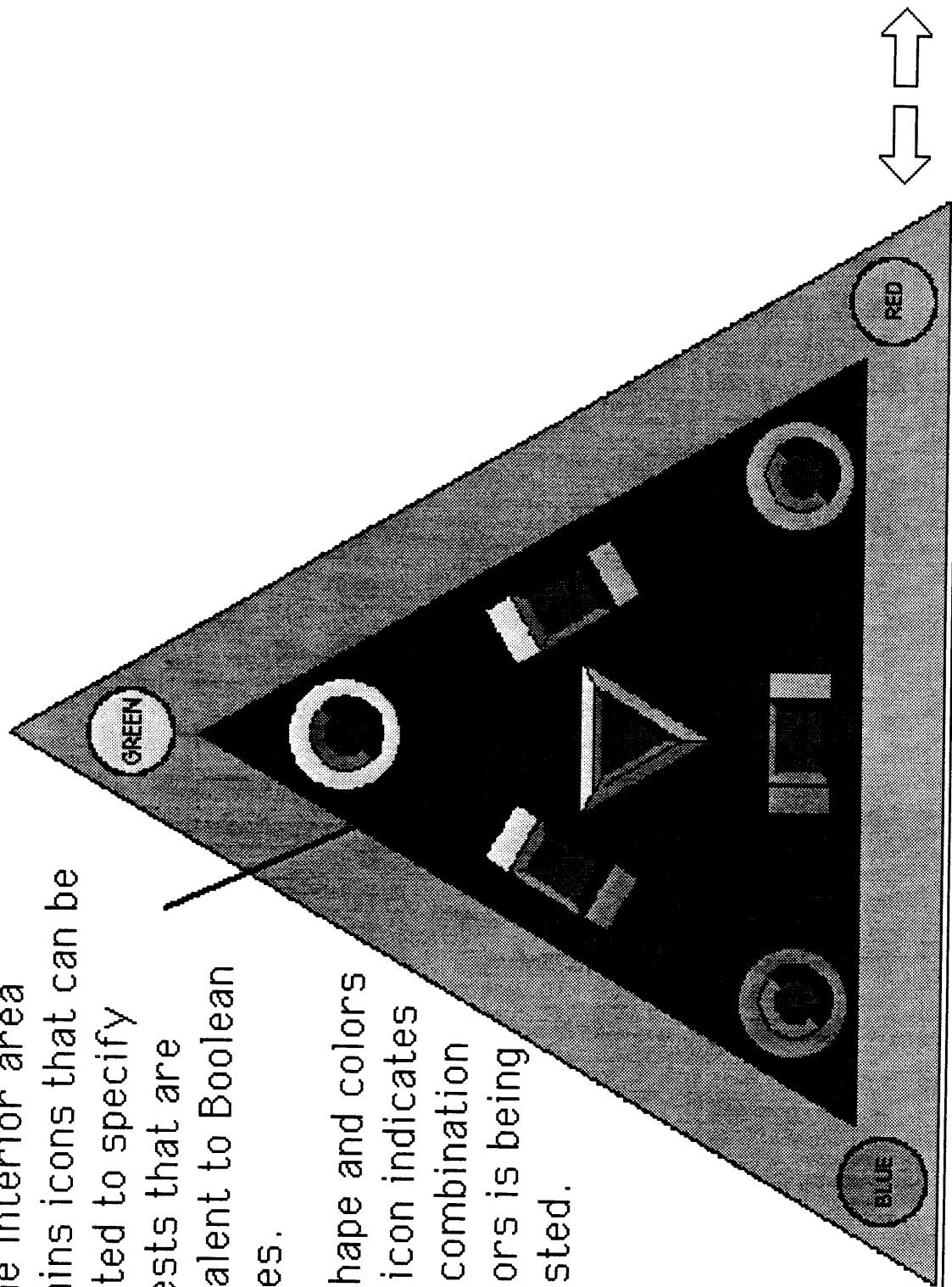
An InfoCrystal has two distinct areas:

- 1) The border area contains icons that represent the different features of interest.



2) The interior area contains icons that can be selected to specify requests that are equivalent to Boolean queries.

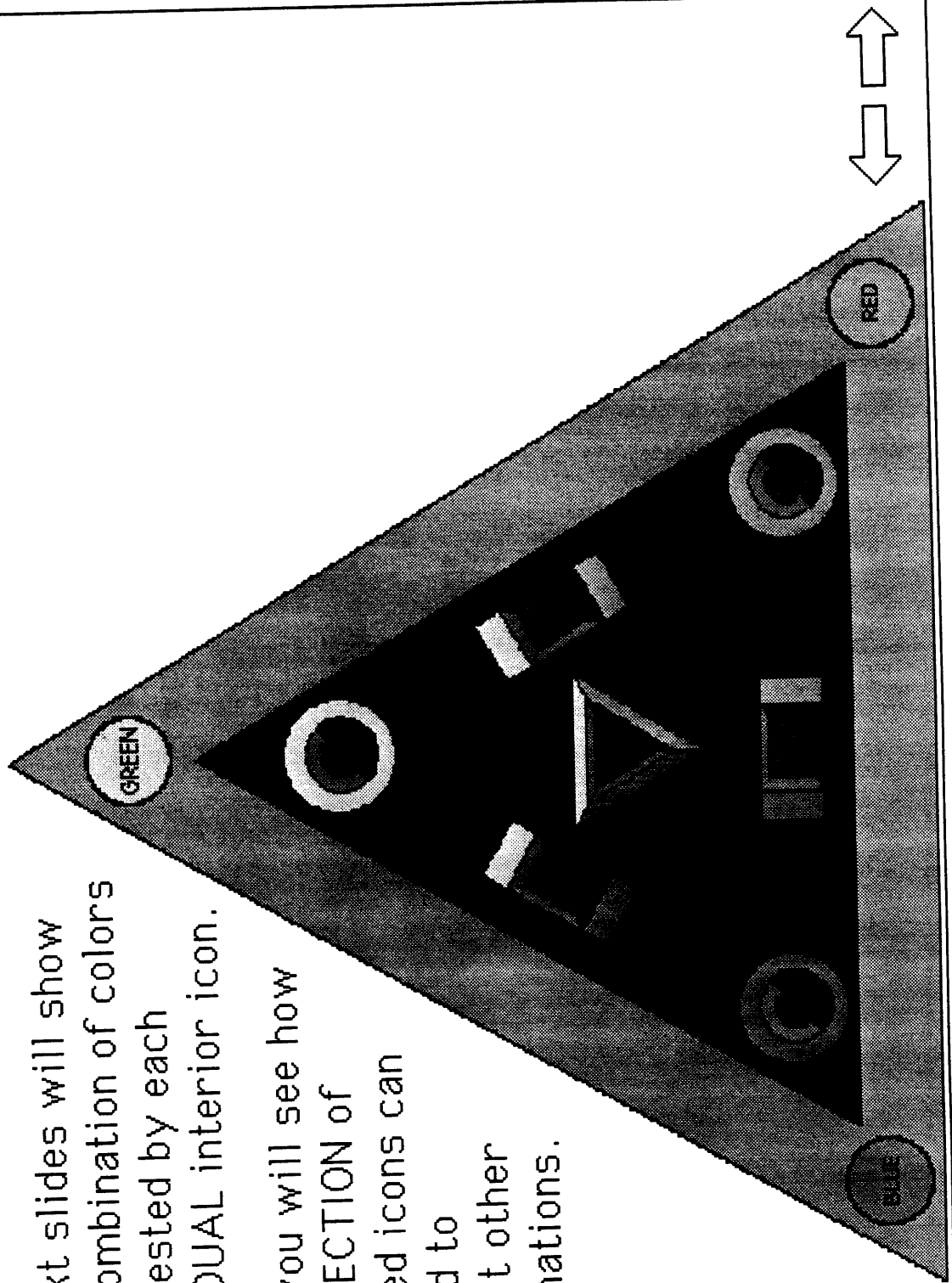
The shape and colors of an icon indicates what combination of colors is being requested.



## How to Represent an Information Need ?

The next slides will show what combination of colors is requested by each INDIVIDUAL interior icon.

Also, you will see how a COLLECTION of selected icons can be used to request other combinations.





**Shape** = CIRCLE indicates that only ONE of the features is present.

**Color** = Green indicates that the Green feature is present and that the other two features are not displayed because their colors are not displayed.

*Selected icons are highlighted and their center area is white.*

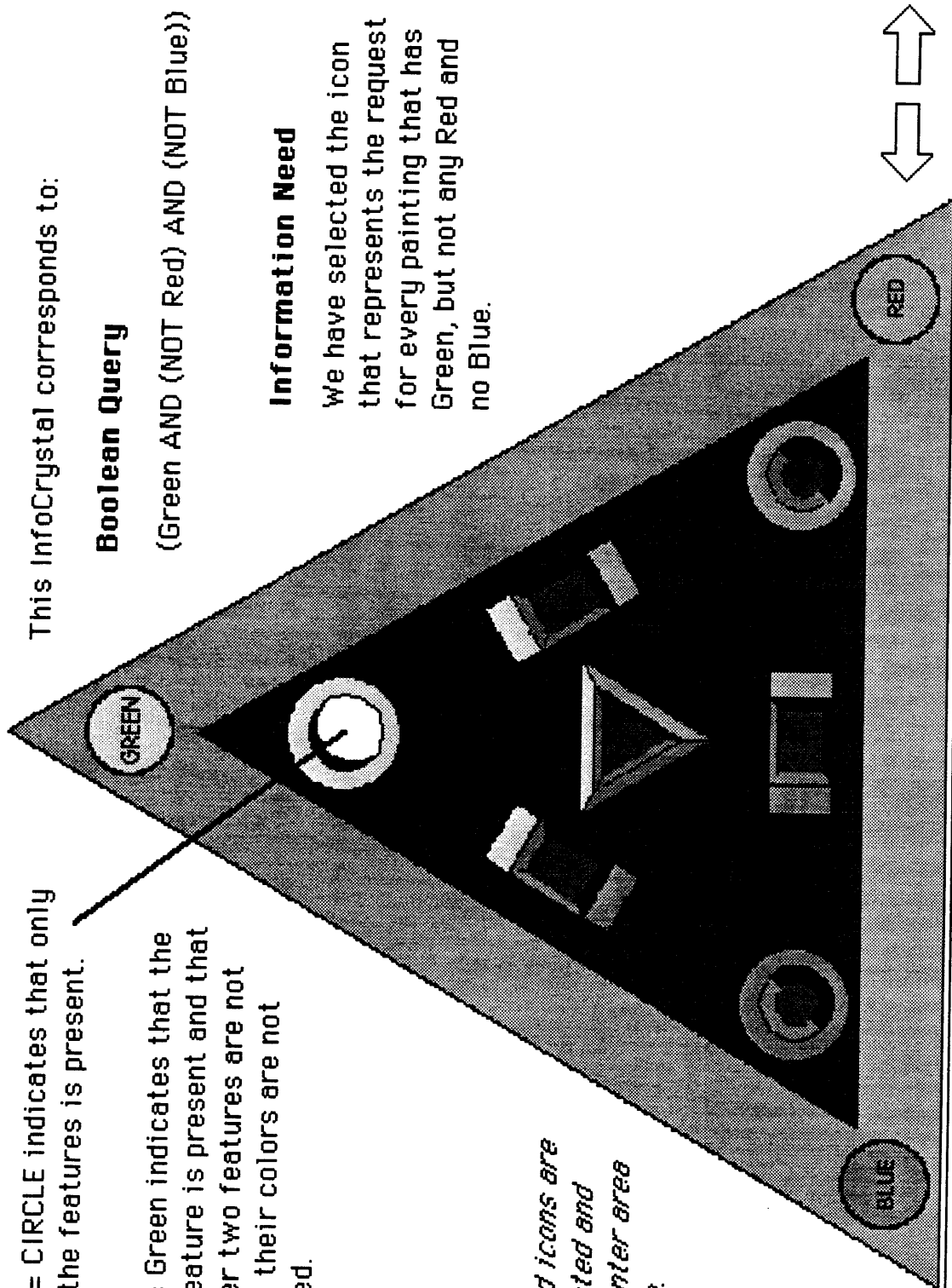
This InfoCrystal corresponds to:

**Boolean Query**

(Green AND (NOT Red) AND (NOT Blue))

**Information Need**

We have selected the icon that represents the request for every painting that has Green, but not any Red and no Blue.



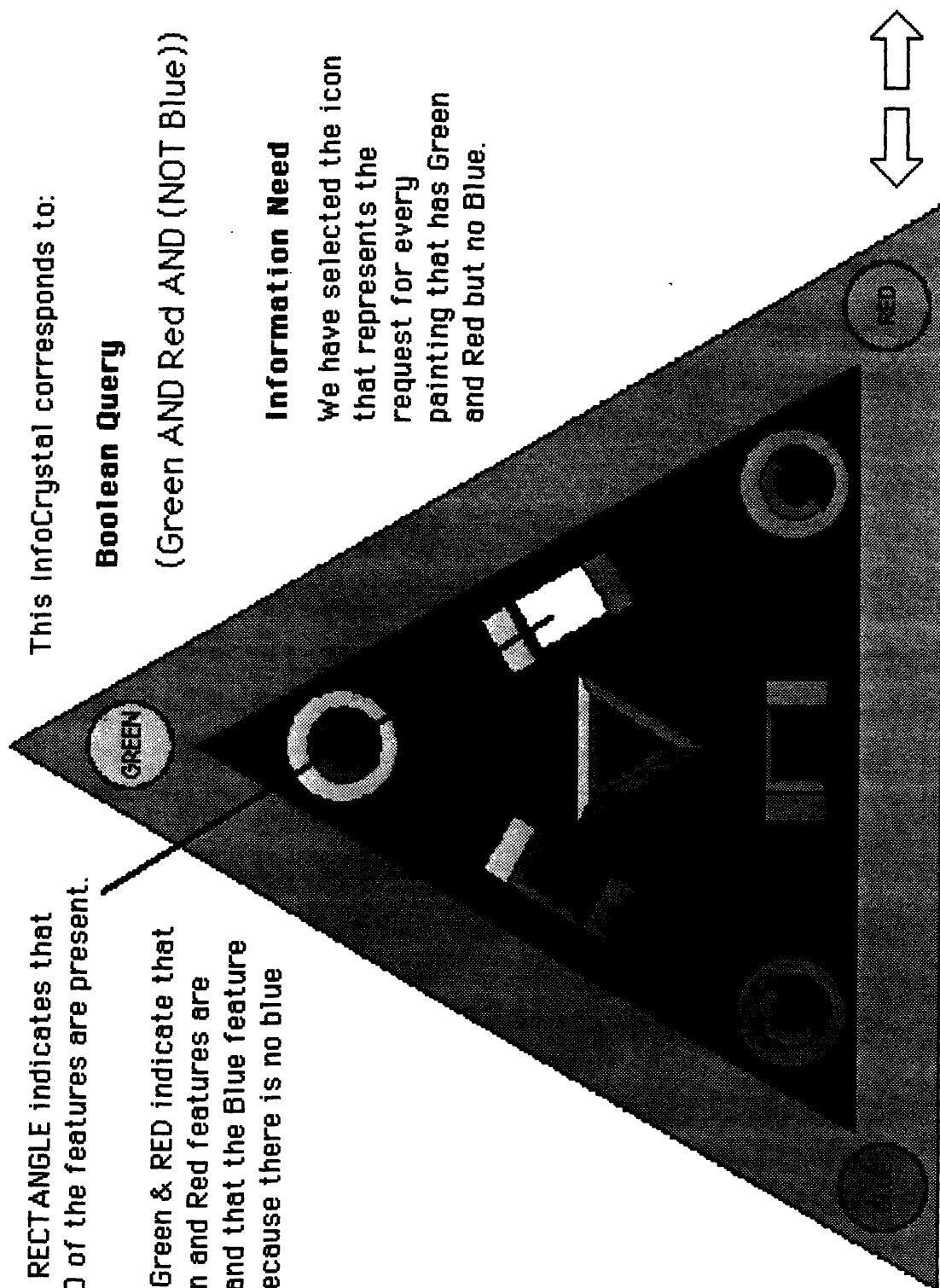
**Shape** = RECTANGLE indicates that just TWO of the features are present.

**Color** = Green & RED indicate that the Green and Red features are present and that the Blue feature is not (because there is no blue color).

This InfoCrystal corresponds to:

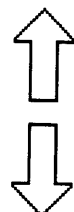
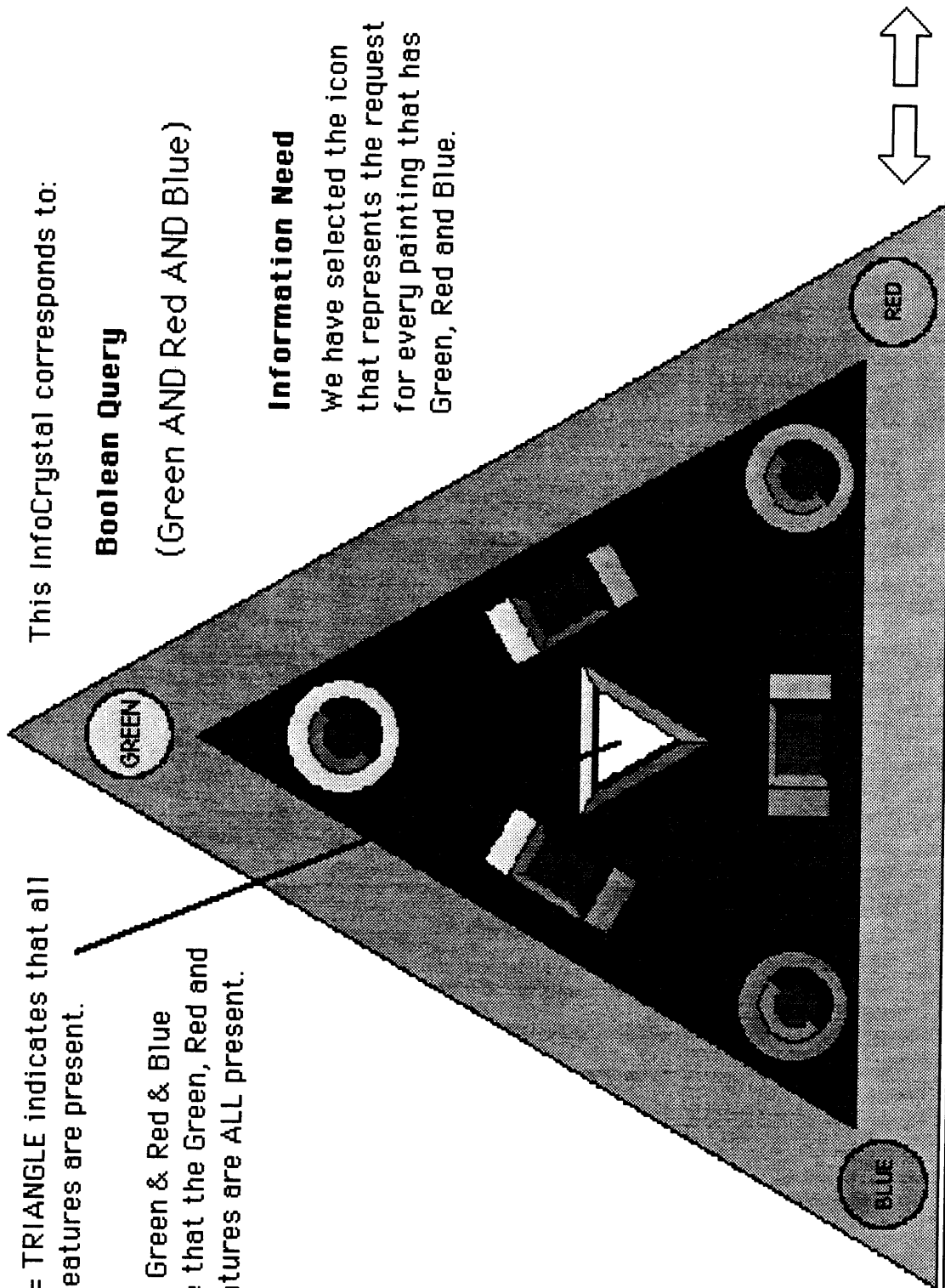
**Boolean Query**  
(Green AND Red AND (NOT Blue))

**Information Need**  
We have selected the icon that represents the request for every painting that has Green and Red but no Blue.



**Shape** = TRIANGLE indicates that all THREE features are present.

**Color** = Green & Red & Blue indicate that the Green, Red and Blue features are ALL present.



If two icons are selected at the same time, then it means that we are requesting all records requested by the first icon PLUS all the ones requested by the second one.

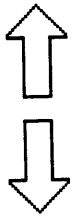
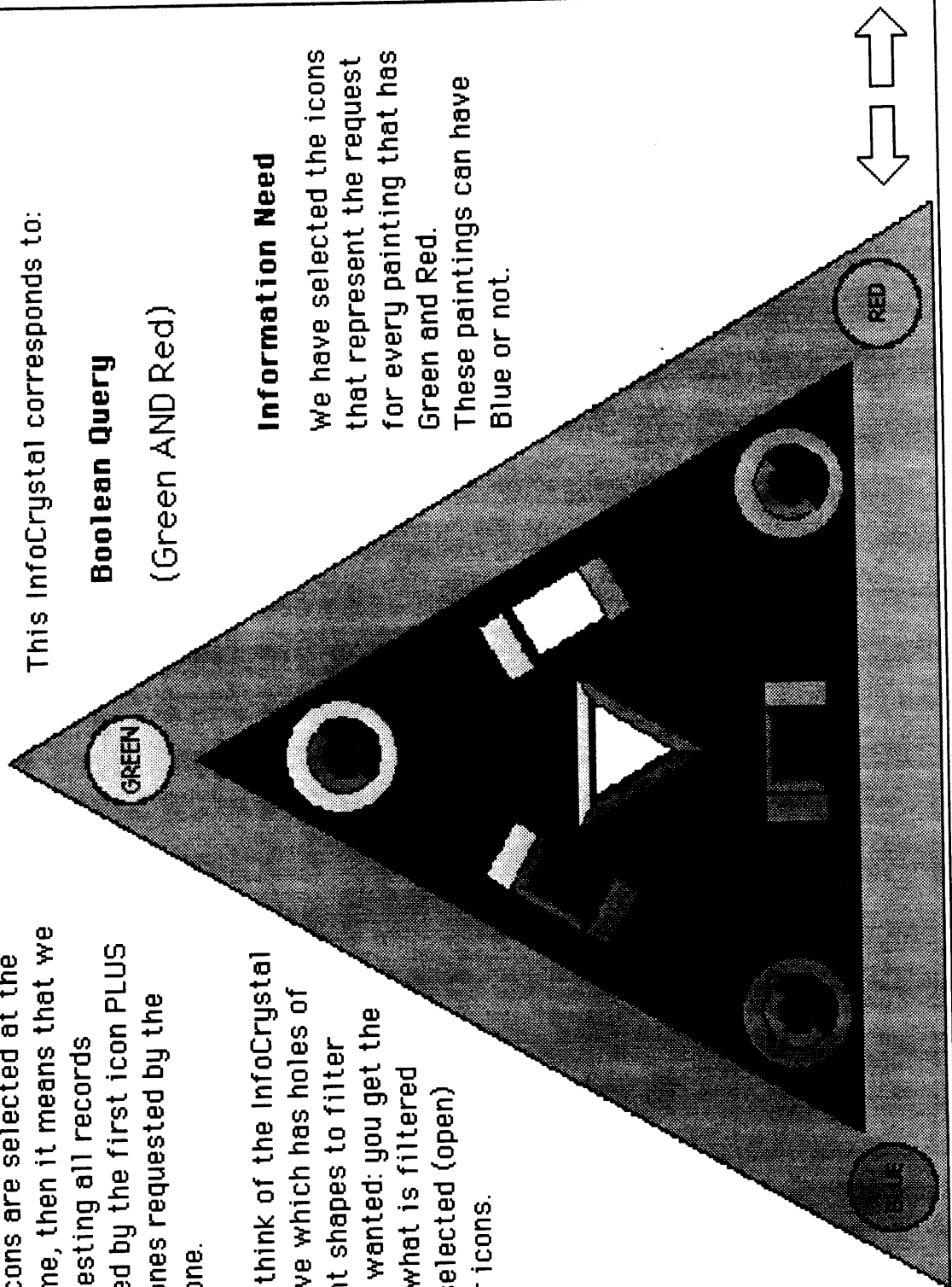
You can think of the InfoCrystal as a sieve which has holes of different shapes to filter what is wanted: you get the SUM of what is filtered by the selected (open) interior icons.

This InfoCrystal corresponds to:

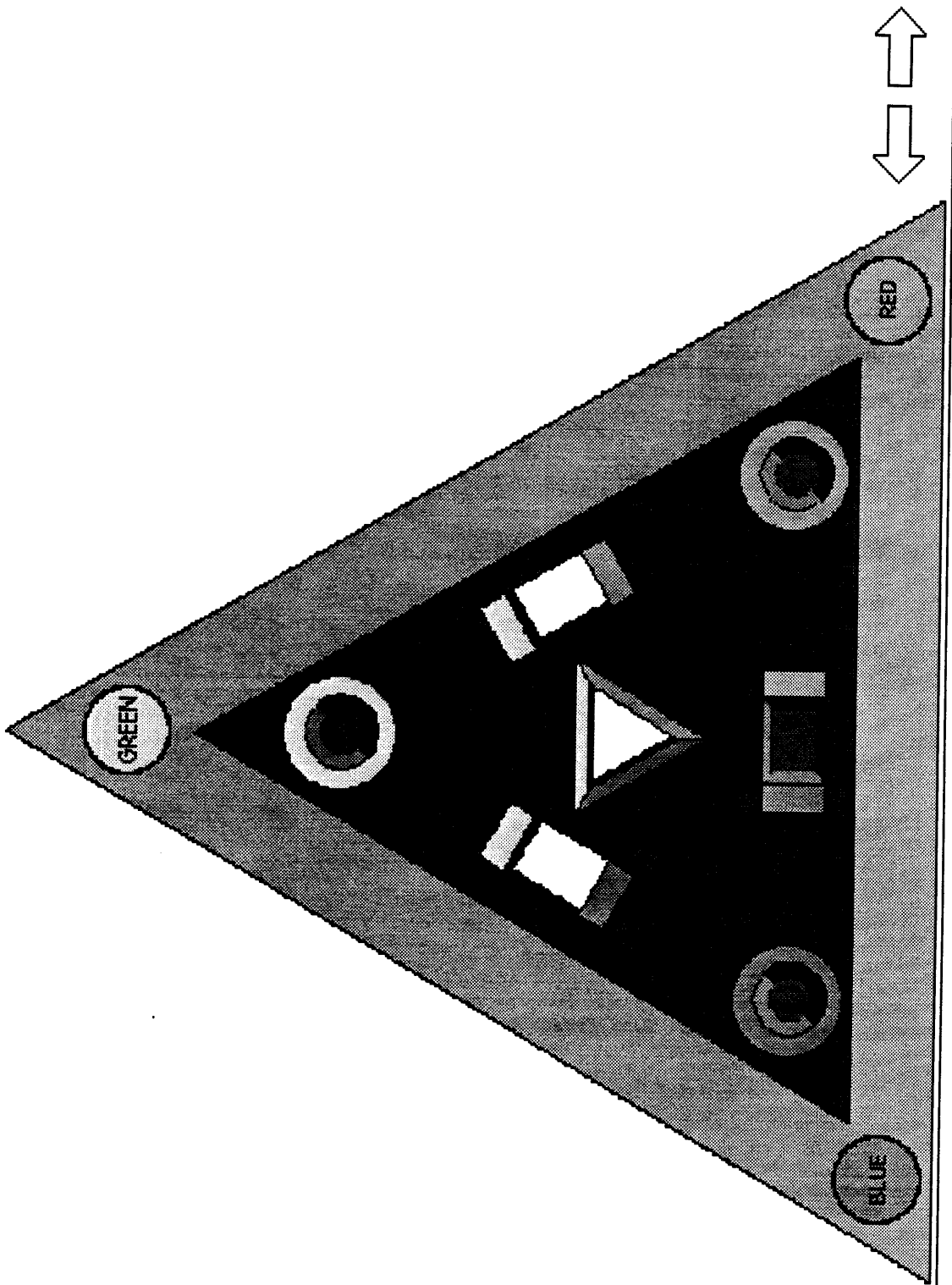
**Boolean Query**  
(Green AND Red)

**Information Need**

We have selected the icons that represent the request for every painting that has Green and Red. These paintings can have Blue or not.



Which Boolean query does this InfoCrystal represent ?

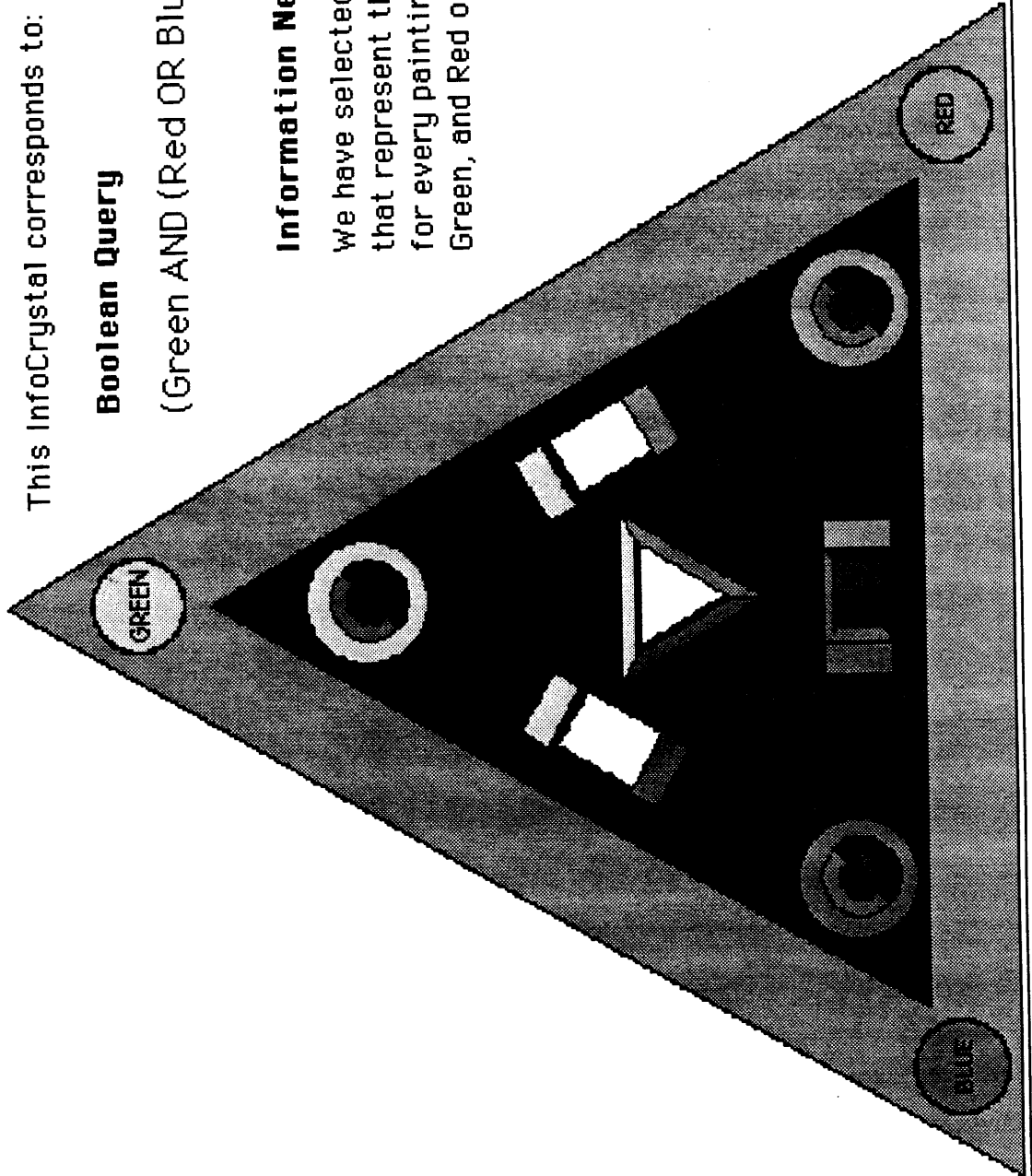


This InfoCrystal corresponds to:

**Boolean Query**  
(Green AND (Red OR Blue))

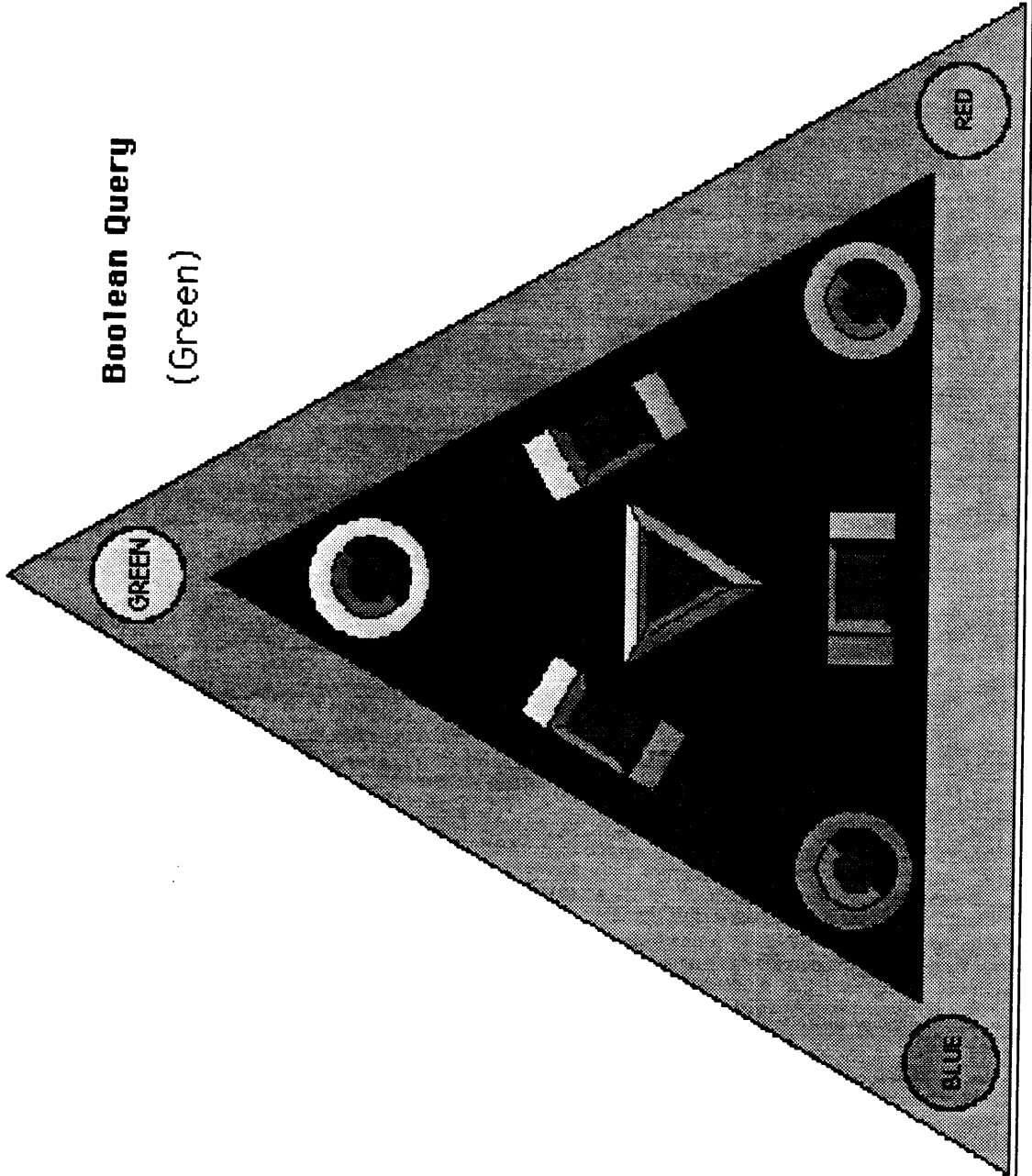
**Information Need**

We have selected the icons that represent the request for every painting that has Green, and Red or Blue.



Which interior icons do we have to select to represent this Boolean query ?

**Boolean Query**  
(Green)

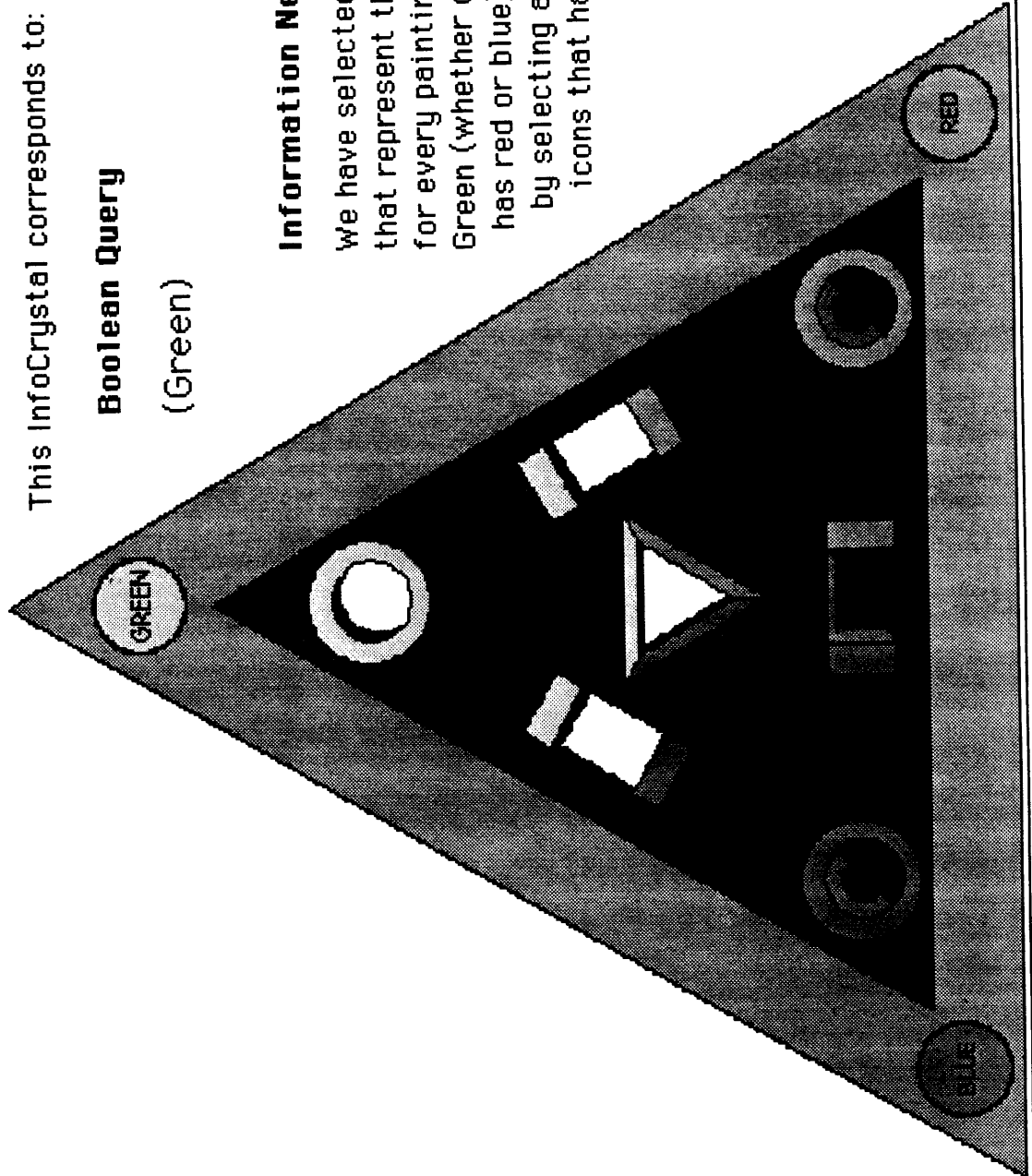


This InfoCrystal corresponds to:

**Boolean Query**  
(Green)

**Information Need**

We have selected the icons that represent the request for every painting that has Green (whether or not it has red or blue) by selecting all those icons that have green.



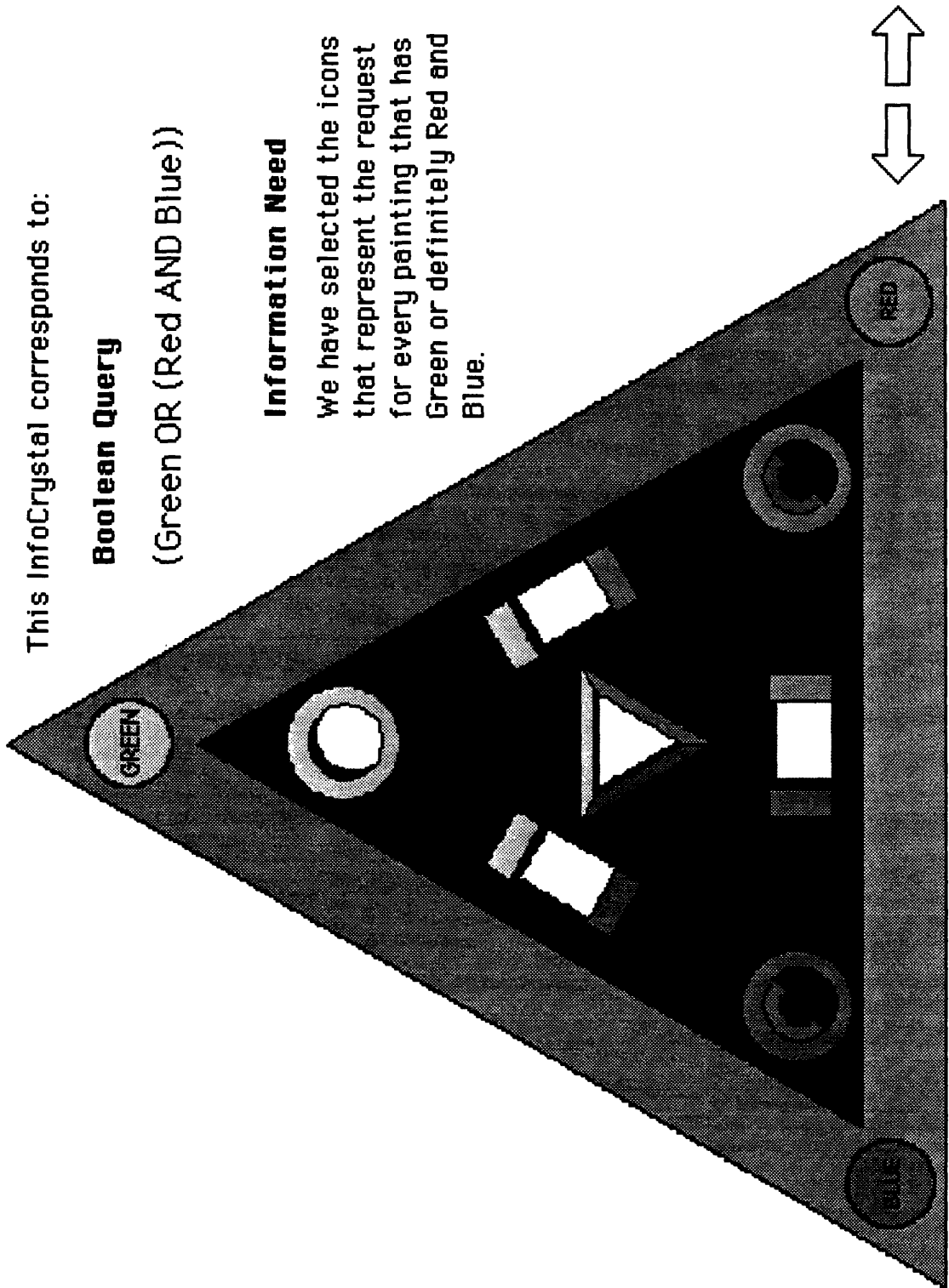


This InfoCrystal corresponds to:

**Boolean Query**  
(Green OR (Red AND Blue))

**Information Need**

We have selected the icons that represent the request for every painting that has Green or definitely Red and Blue.

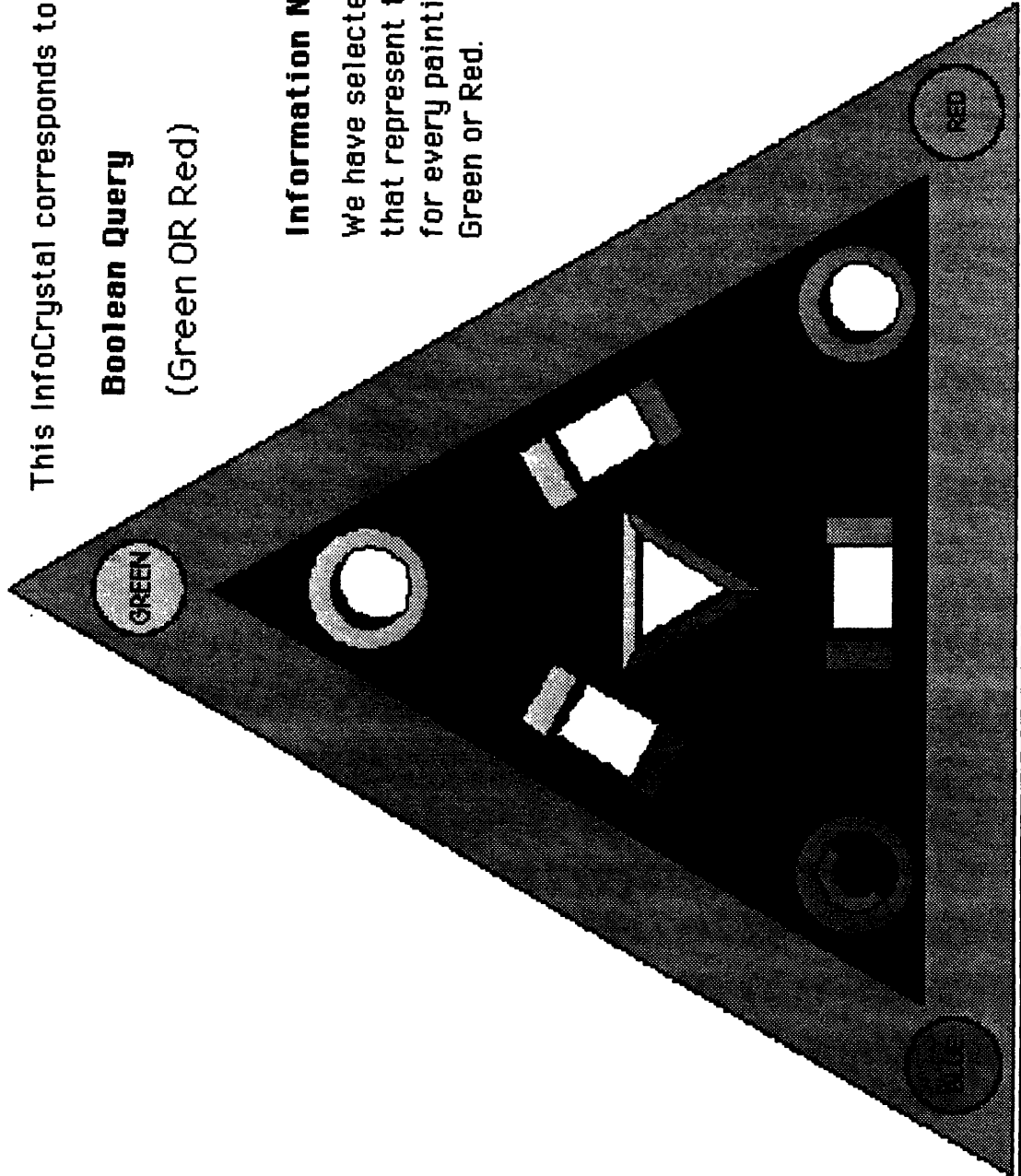


This InfoCrystal corresponds to:

**Boolean Query**  
(Green OR Red)

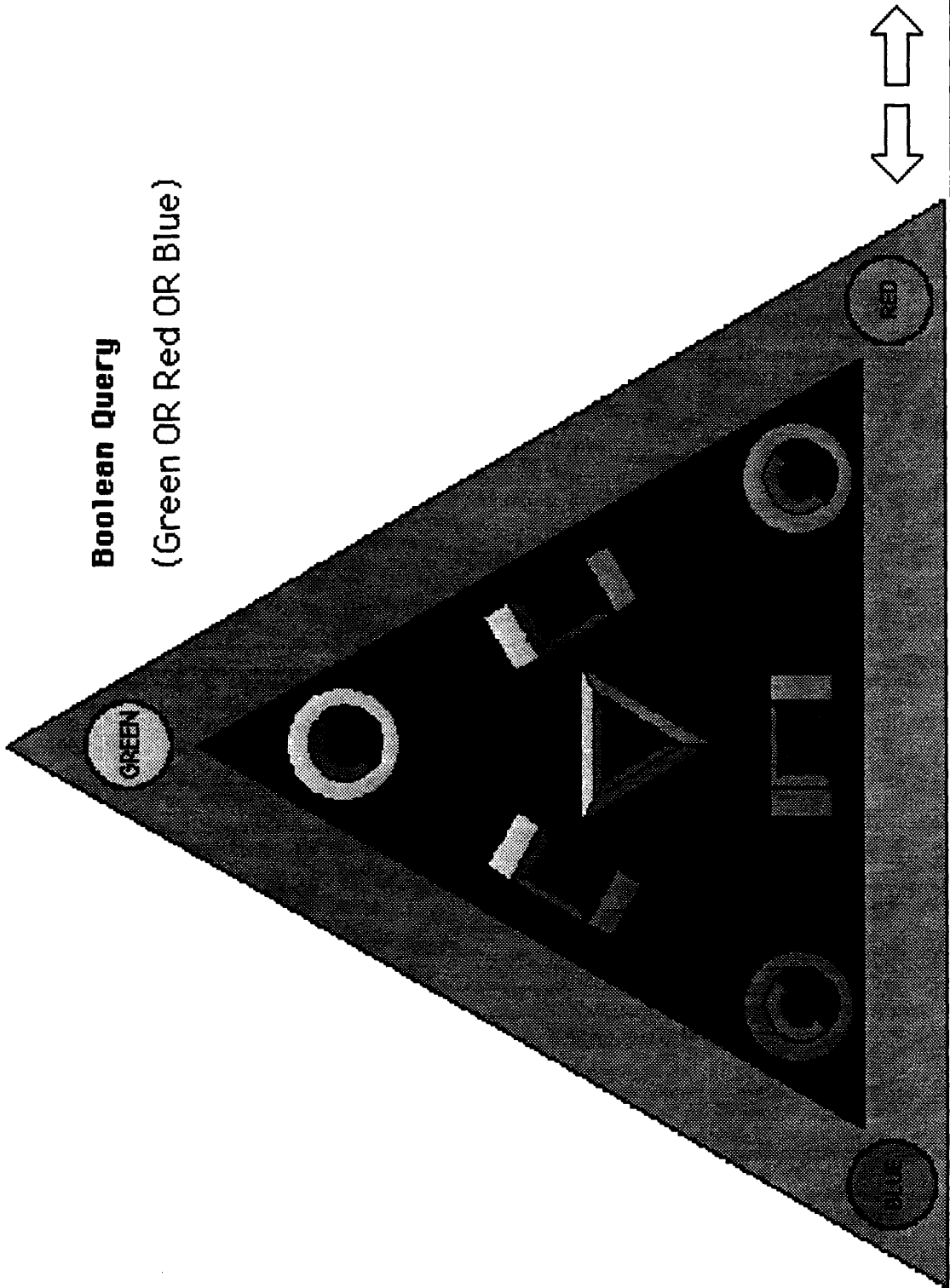
**Information Need**

We have selected the icons that represent the request for every painting that has Green or Red.



Which interior icons do we have to select to represent this Boolean query ?

**Boolean Query**  
(Green OR Red OR Blue)

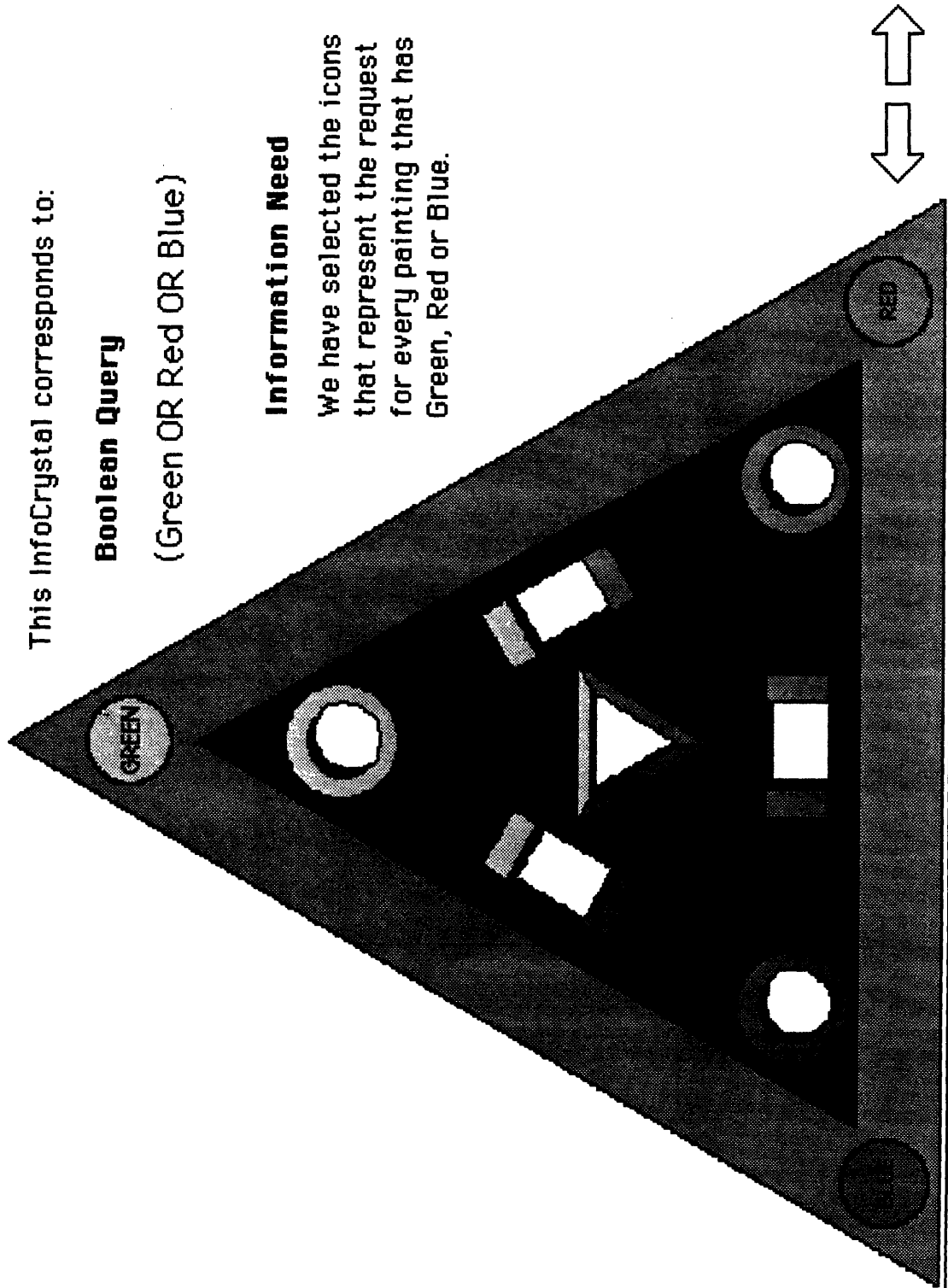


This InfoCrystal corresponds to:

**Boolean Query**  
(Green OR Red OR Blue)

**Information Need**

We have selected the icons that represent the request for every painting that has Green, Red or Blue.



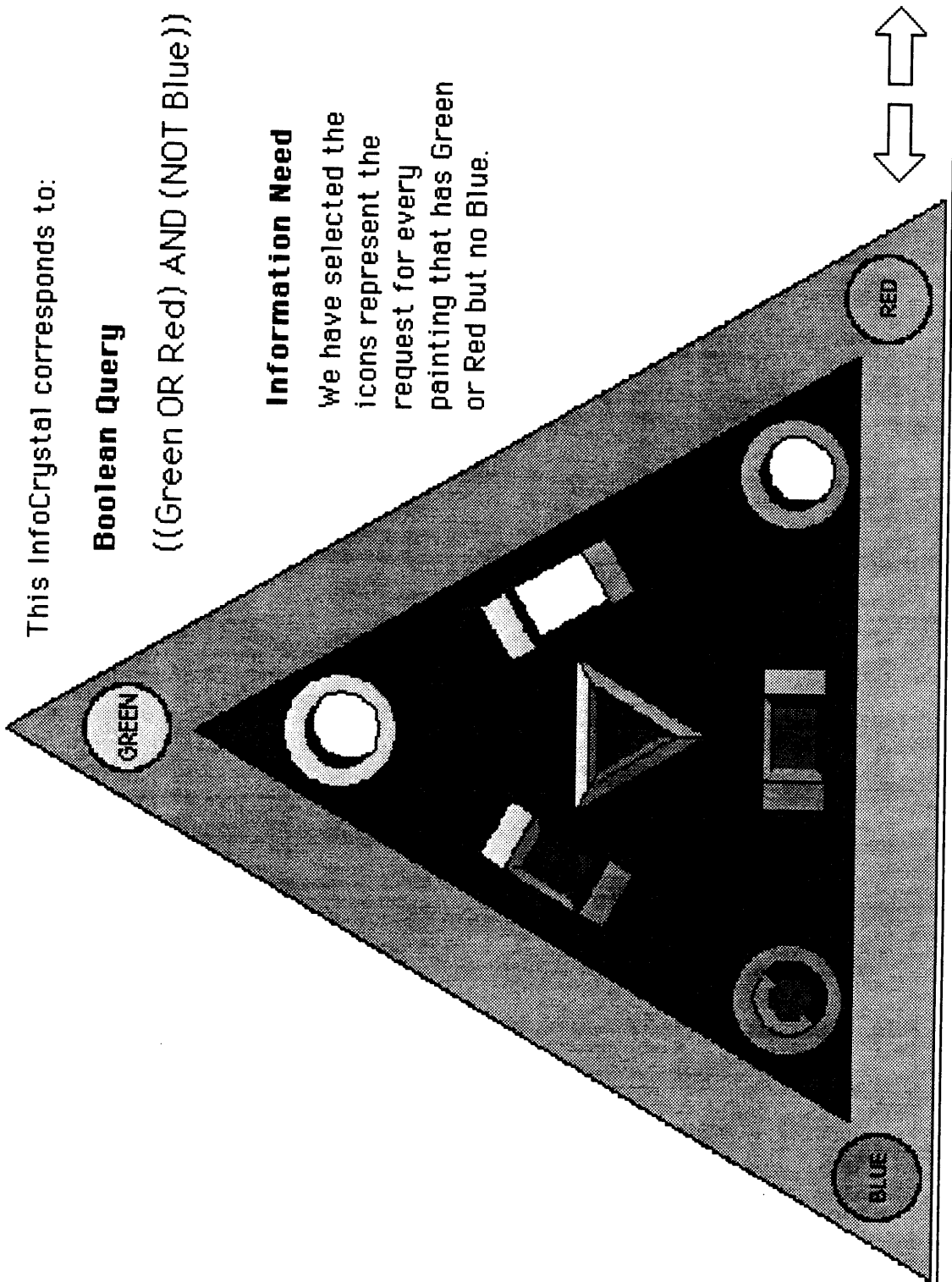
This InfoCrystal corresponds to:

**Boolean Query**

**((Green OR Red) AND (NOT Blue))**

**Information Need**

We have selected the icons represent the request for every painting that has Green or Red but no Blue.

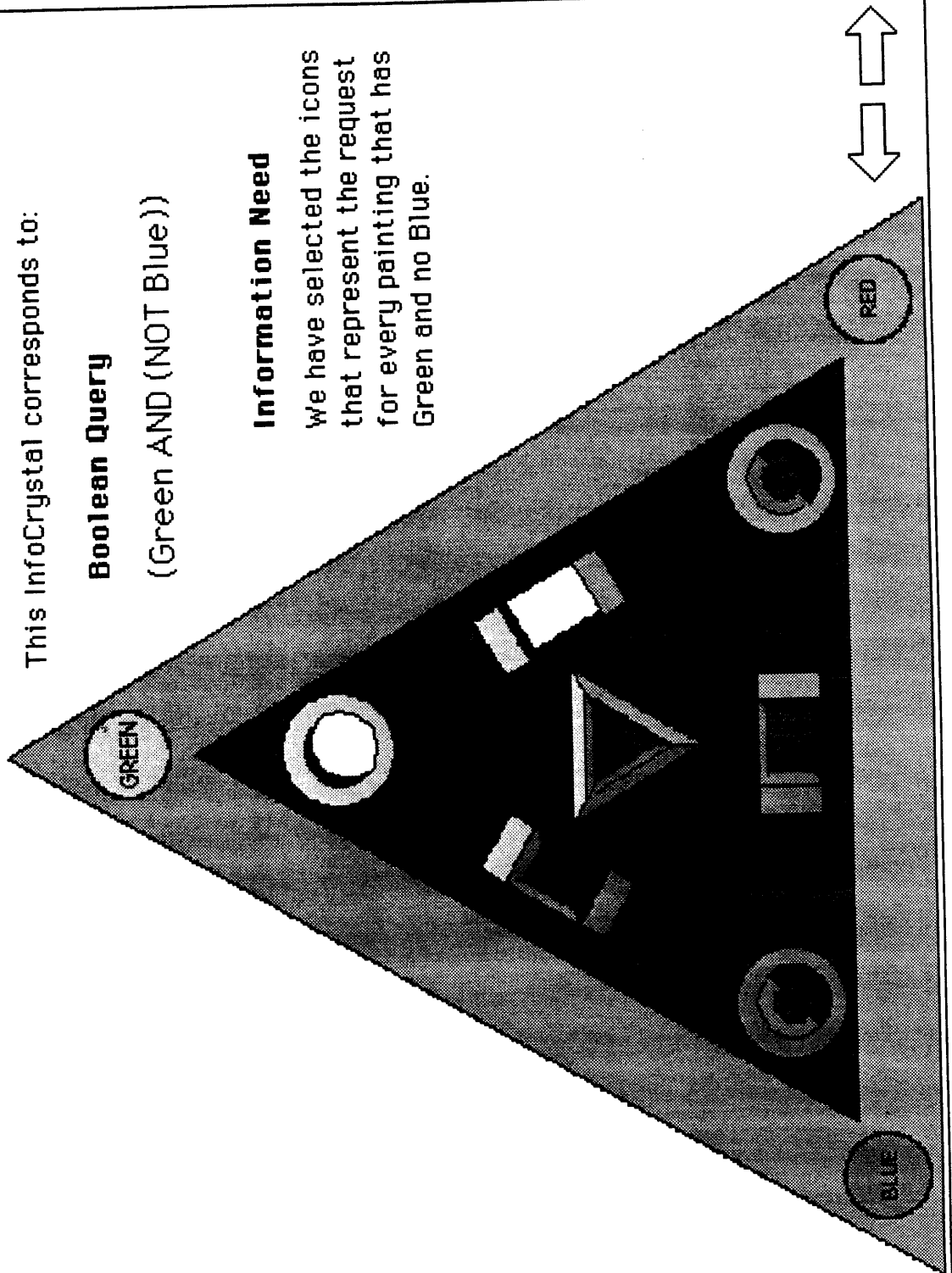


This InfoCrystal corresponds to:

**Boolean Query**  
(Green AND (NOT Blue))

**Information Need**

We have selected the icons that represent the request for every painting that has Green and no Blue.



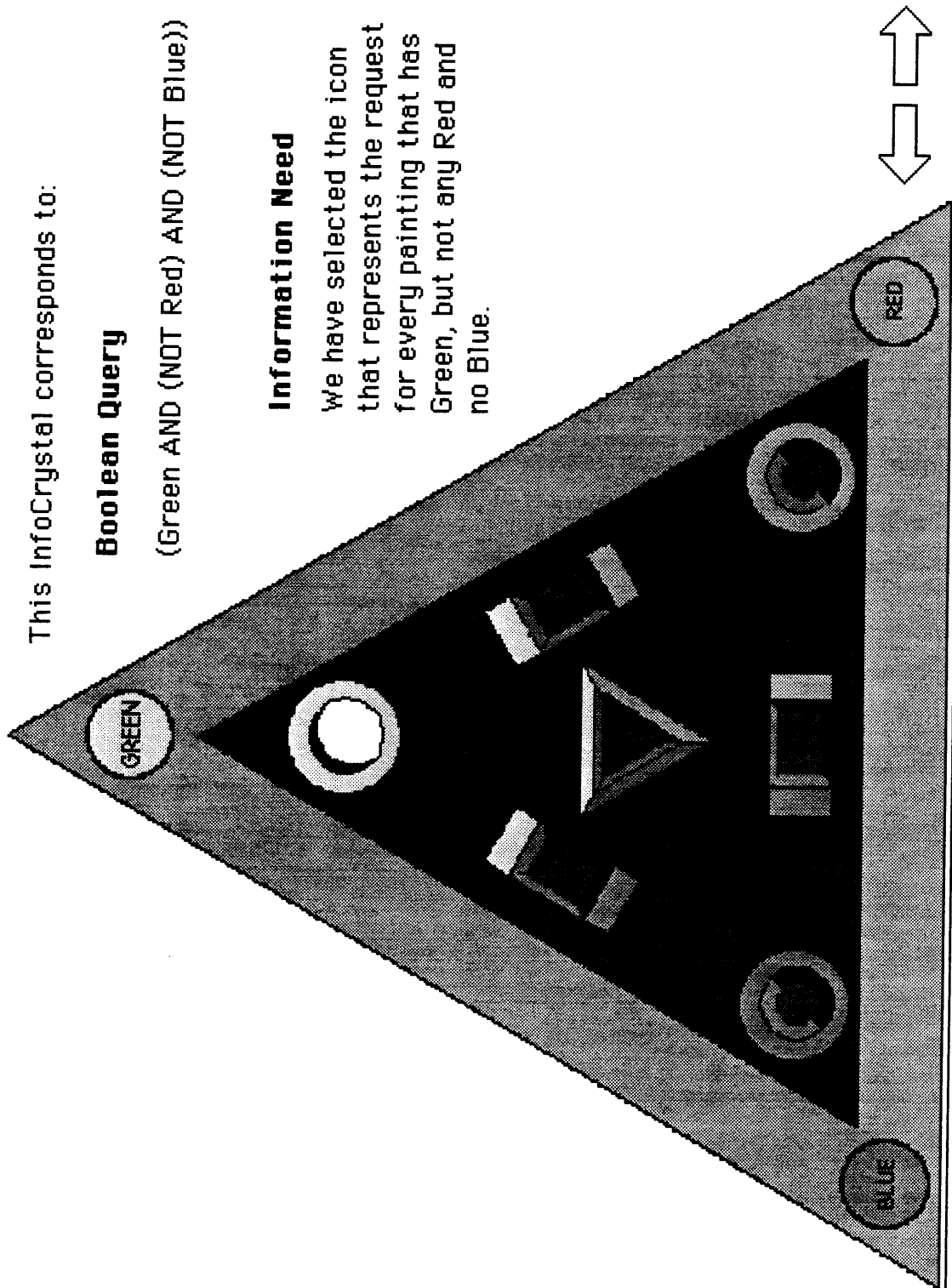
This InfoCrystal corresponds to:

**Boolean Query**

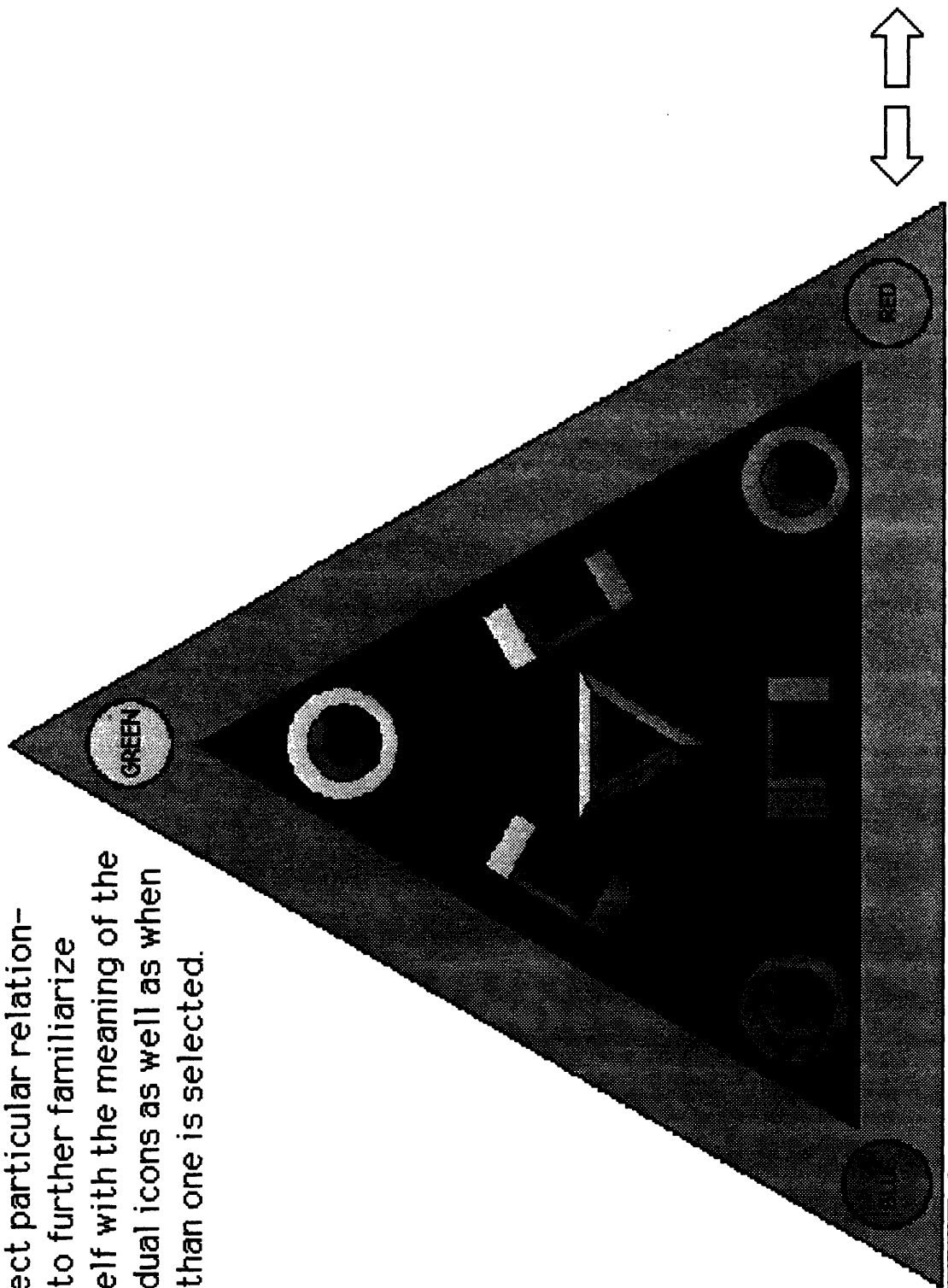
(Green AND (NOT Red) AND (NOT Blue))

**Information Need**

We have selected the icon that represents the request for every painting that has Green, but not any Red and no Blue.



Please **click now** on any of the **interior icons** to select or deselect particular relationships to further familiarize yourself with the meaning of the individual icons as well as when more than one is selected.





## InfoCrystal Summary

The **shape** of an interior icon indicates how many of the features are present.

The **colored sides** of an interior icon reflect which ones of the features that we are considering at the same time are present, and the absent colors for an icon indicate which features are not present.

By selecting one or several of the interior icons we can represent any Boolean query and its associated information need.



Main Menu

