CPSC 533C Information Visualization Project Update

Exploratory Browsing in Music Space

Heidi Lam

November 17, 2004

Agenda

- Motivation: Exploratory browsing?
- The ideal infovis solution: what should it be?
- Related work: displaying query-based results
- Prototypes: my proposed solution
- Dataset and implementation
- List of ongoing and future work

Project Idea

- How can computer tools/interfaces better support exploratory browsing?

- What is exploratory browsing?

Query Taxonomy

	Specified Target	Uncertain Target
Specified Location	**Navigation:** if a map of the space is present **Exploration:** if not Redundant encoding (target and location) to evaluate if the target is found	**Navigation:** if a map of the space is present **Exploration:** if not Single encoding (location) to evaluate if the target is found
Uncertain Location	**Search/find** with static evaluation (i.e., looking for something defined)	**Browsing** with potentially dynamic evaluation (i.e., target is ill-defined, and its properties may change/be refined along the process).

Query Taxonomy

	Specified Target	Uncertain Target
Specified Location	**Navigation:** if a map of the space is present **Exploration:** if not Redundant encoding (target and location) to evaluate if the target is found	**Navigation:** if a map of the space is present **Exploration:** if not Single encoding (location) to evaluate if the target is found
Uncertain Location	**Search/find** with static evaluation (i.e., looking for something defined)	**Browsing** with potentially dynamic evaluation (i.e., target is ill-defined, and its properties may change/be refined along the process).

Query Taxonomy

	Specified Target	Uncertain Target
Specified Location	**Navigation:** if a map of the space is present **Exploration:** if not Redundant encoding (target and location) to evaluate if the target is found	**Navigation:** if a map of the space is present **Exploration:** if not Single encoding (location) to evaluate if the target is found
Uncertain Location	**Search/find** with static evaluation (i.e., looking for something defined)	**Browsing** with potentially dynamic evaluation (i.e., target is ill-defined, and its properties may change/be refined along the process).

Query Taxonomy

	Specified Target	Uncertain Target
Specified Location	**Navigation:** if a map of the space is present **Exploration:** if not Redundant encoding (target and location) to evaluate if the target is found	**Navigation:** if a map of the space is present **Exploration:** if not Single encoding (location) to evaluate if the target is found
Uncertain Location	**Search/find** with static evaluation (i.e., looking for something defined)	**Browsing** with potentially dynamic evaluation (i.e., target is ill-defined, and its properties may change/be refined along the process).

Query Taxonomy

	Specified Target	Uncertain Target
Specified Location	**Navigation:** if a map of the space is present **Exploration:** if not Redundant encoding (target and location) to evaluate if the target is found	**Navigation:** if a map of the space is present **Exploration:** if not Single encoding (location) to evaluate if the target is found
Uncertain Location	**Search/find** with static evaluation (i.e., looking for something defined)	**Browsing** with potentially dynamic evaluation (i.e., target is ill-defined, and its properties may change/be refined along the process).

Two Scenarios at a Record Store

1. Looking for Ray Charles' "Come Rain or Come Shine"
   - Navigate: Go to "Jazz" à Search under "C" à Find "Ray Charles" à Search among his albums
   - Find/Search: "Do you have Ray Charles' "Come rain or come shine"?

2. Browsing at the "Classical" section à Came across a Jazzified version of Bach à Go to the "Jazz" section à Ray Charles' album is on display

Two Scenarios at a Record Store

The goals of these scenarios are different:

- With find/search/navigation: want to find the target as quickly as possible
- With exploratory browsing: getting there is half of the fun/work?

Project Motivation

- Exploratory browsing is not well-supported by current tools

	Specified Target	**Uncertain Target**
Specified Location	*Navigate/Explore* File explorer Web browser	*Navigate/Explore* File explorer Web browser
Uncertain Location	*Find/Search* File searcher Internet search engine	*Browse* Internet search engine?

The Ideal InfoVis Solution

To better support exploratory browsing, the interface should …

1. Provide context: to allow users to interpret the query results based on their input terms
   à where am I? what am I looking at?

2. Guide navigation: going from the familiar to the unfamiliar à where did I come from? where should I go next?

3. Assist refinement of target: based on retrieved results and query terms à what am I looking for?

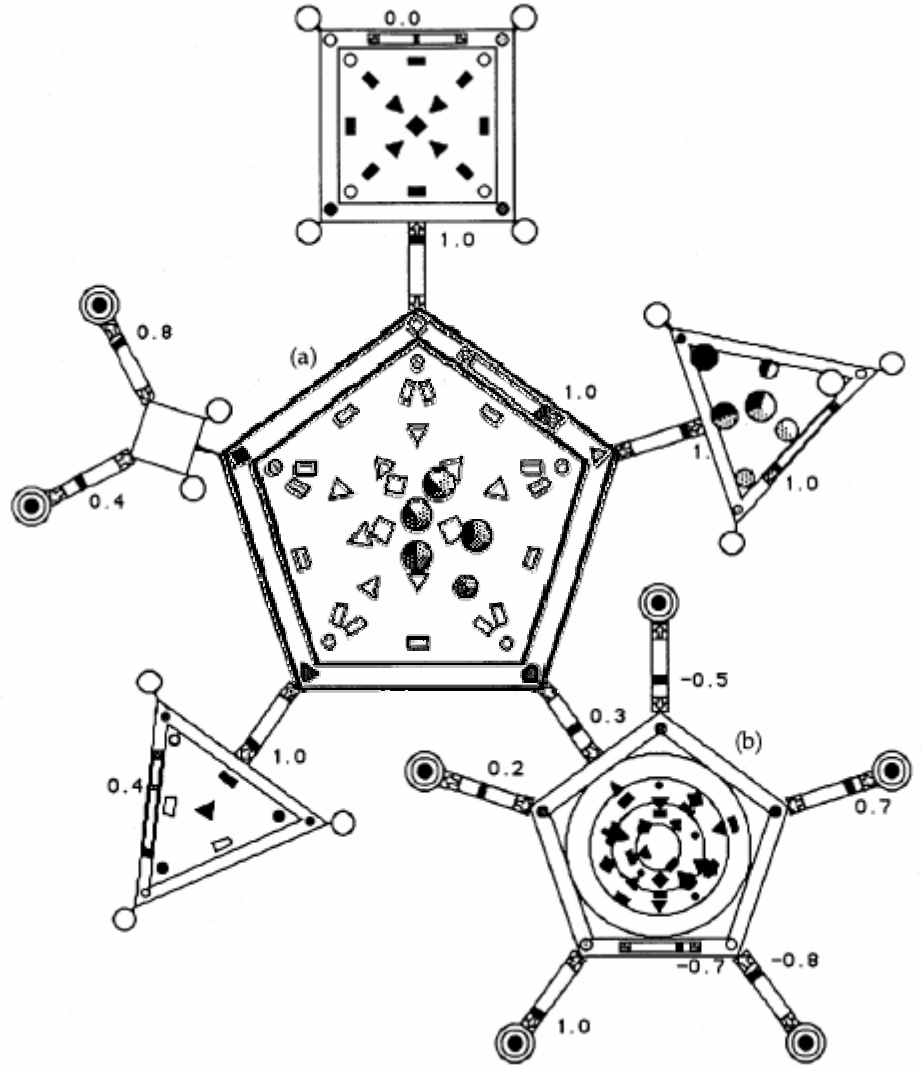
Related Work: Overview

- Focus on query-criteria based from a single search mechanism
- 4 approaches:

1. Spatial: retrieved results are clusters into groups based on query terms, and displayed spatially
2. List: retrieved results are displayed as a linear list
3. Temporal: retrieved results in the context of timelines
4. Integrated: multi-view with combinations of the above approaches

Related Work: Spatial

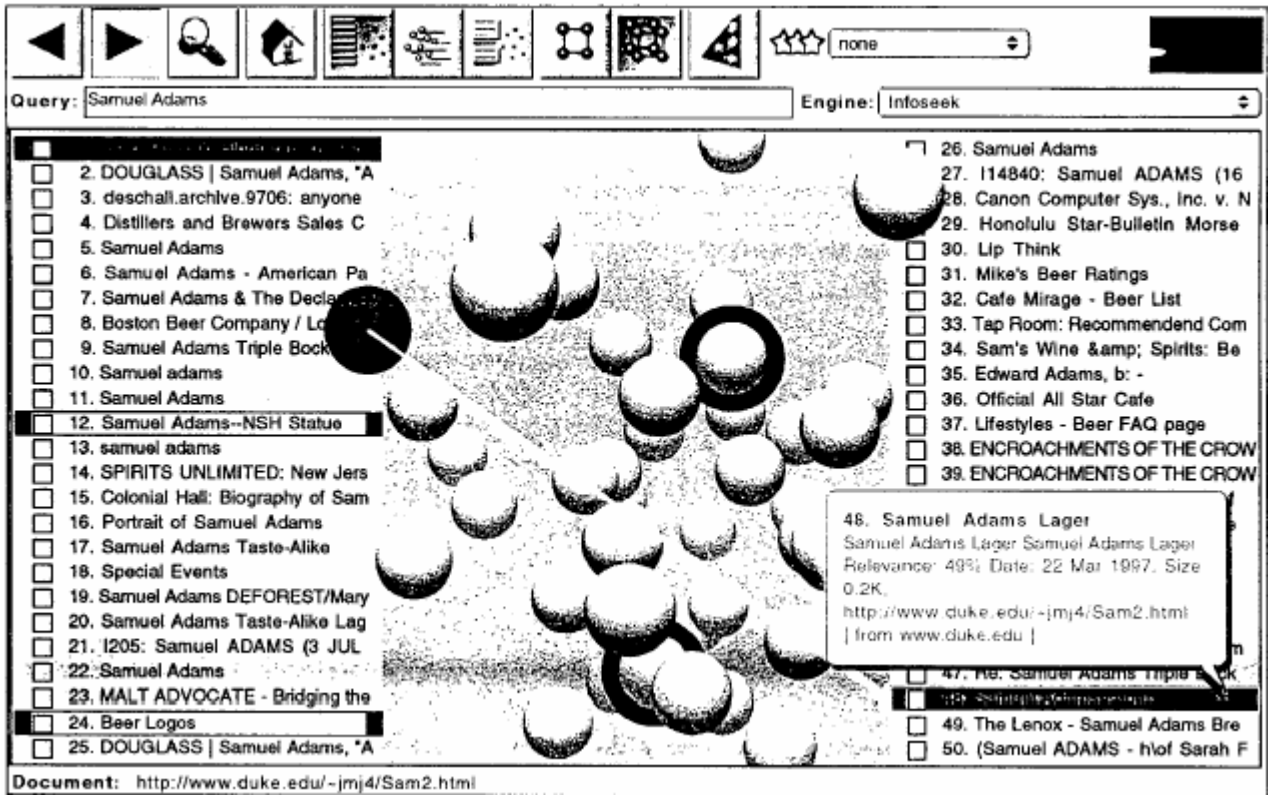
1. Show relationship between keywords—InfoCrystal (1993)



Related Work: Spatial

1. Show relationship between keywords—InfoCrystal (1993)

2. Show clusters only—Lighthouse (2000)

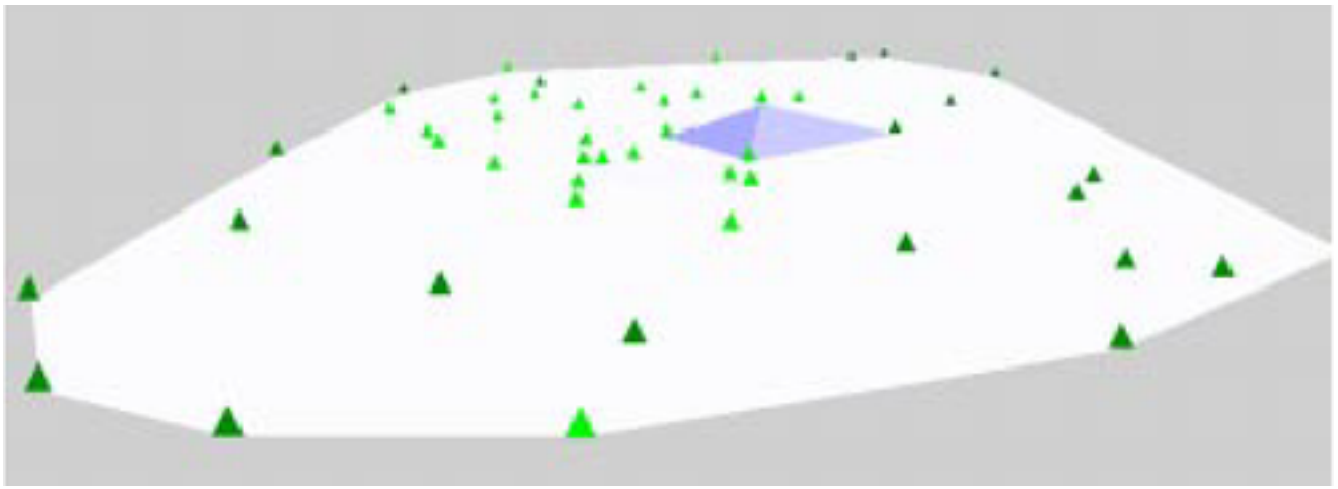


Related Work: Spatial

1. Show relationship between keywords—InfoCrystal (1993)

2. Show clusters only—Lighthouse (2000)

3. 3D terrain—BEAD (1993)



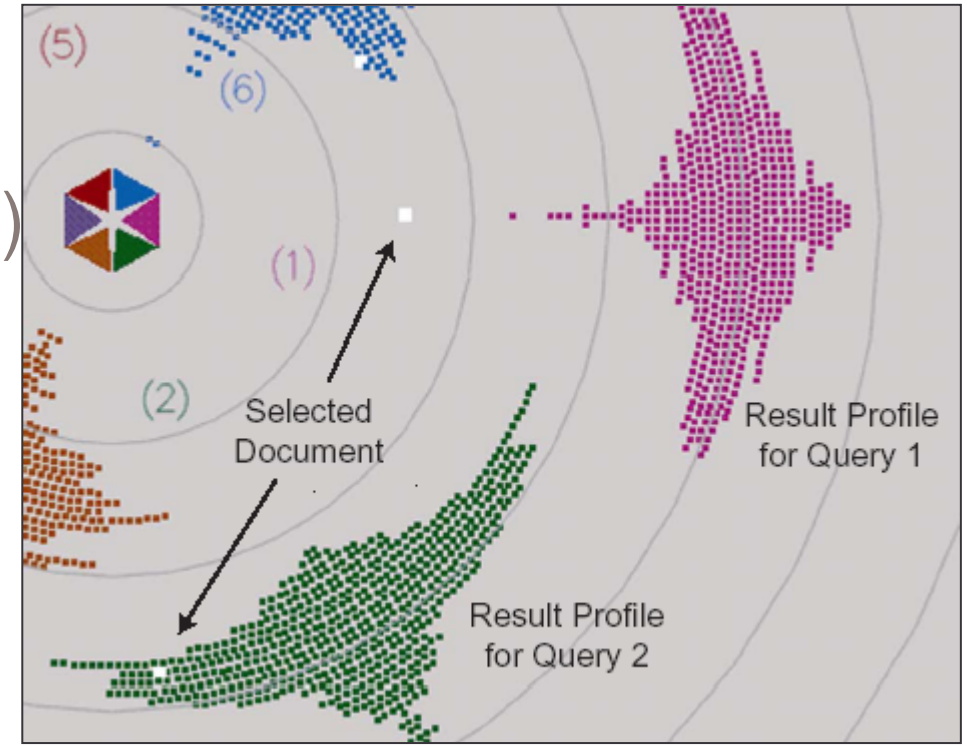
Related Work: Spatial

1. Show relationship between keywords—InfoCrystal (1993)

2. Show clusters only—Lighthouse (2000)

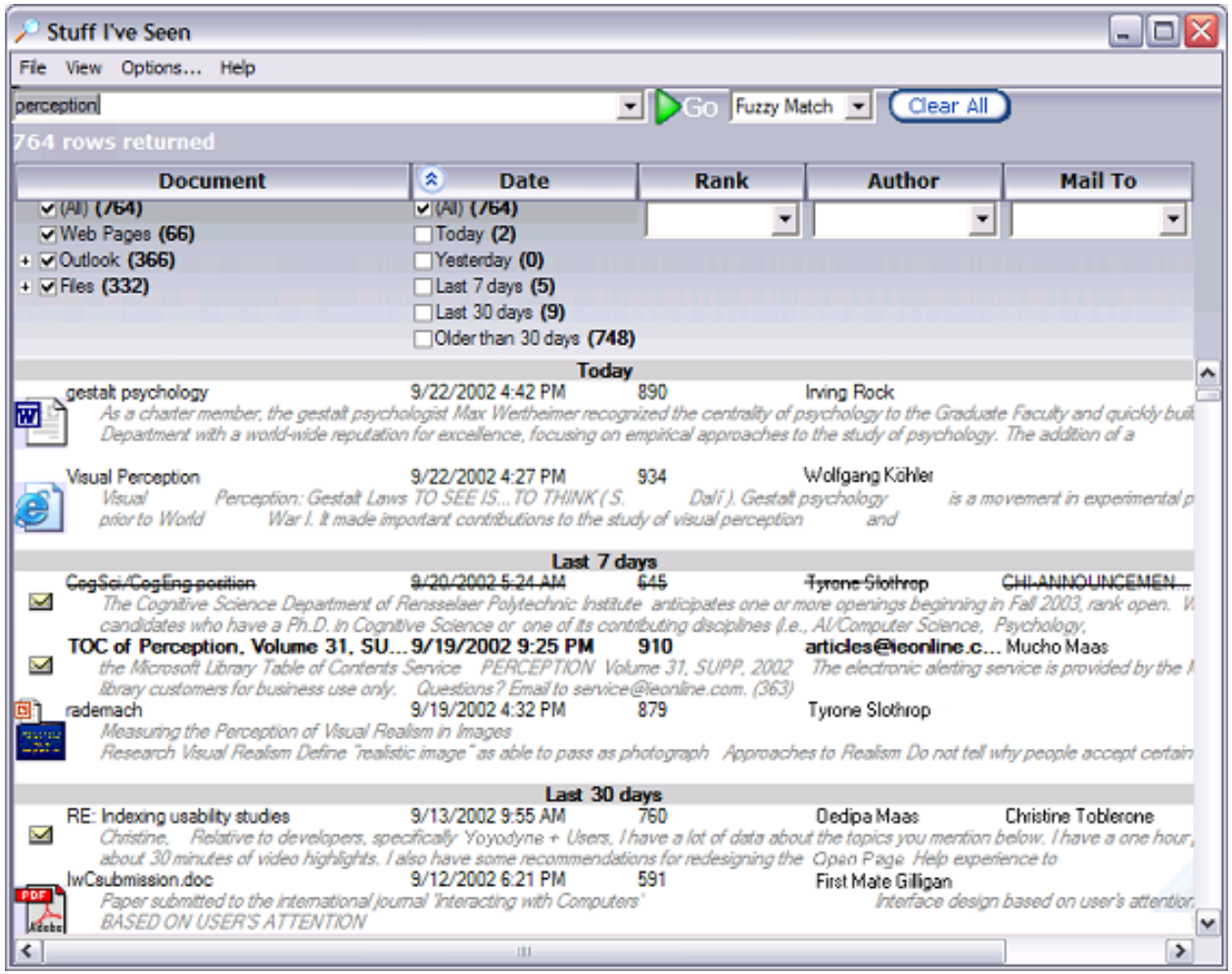
3. 3D terrain—BEAD (1993)

4. Arrange multiple query results spatially —Sparkler (2001)



Related Work: List

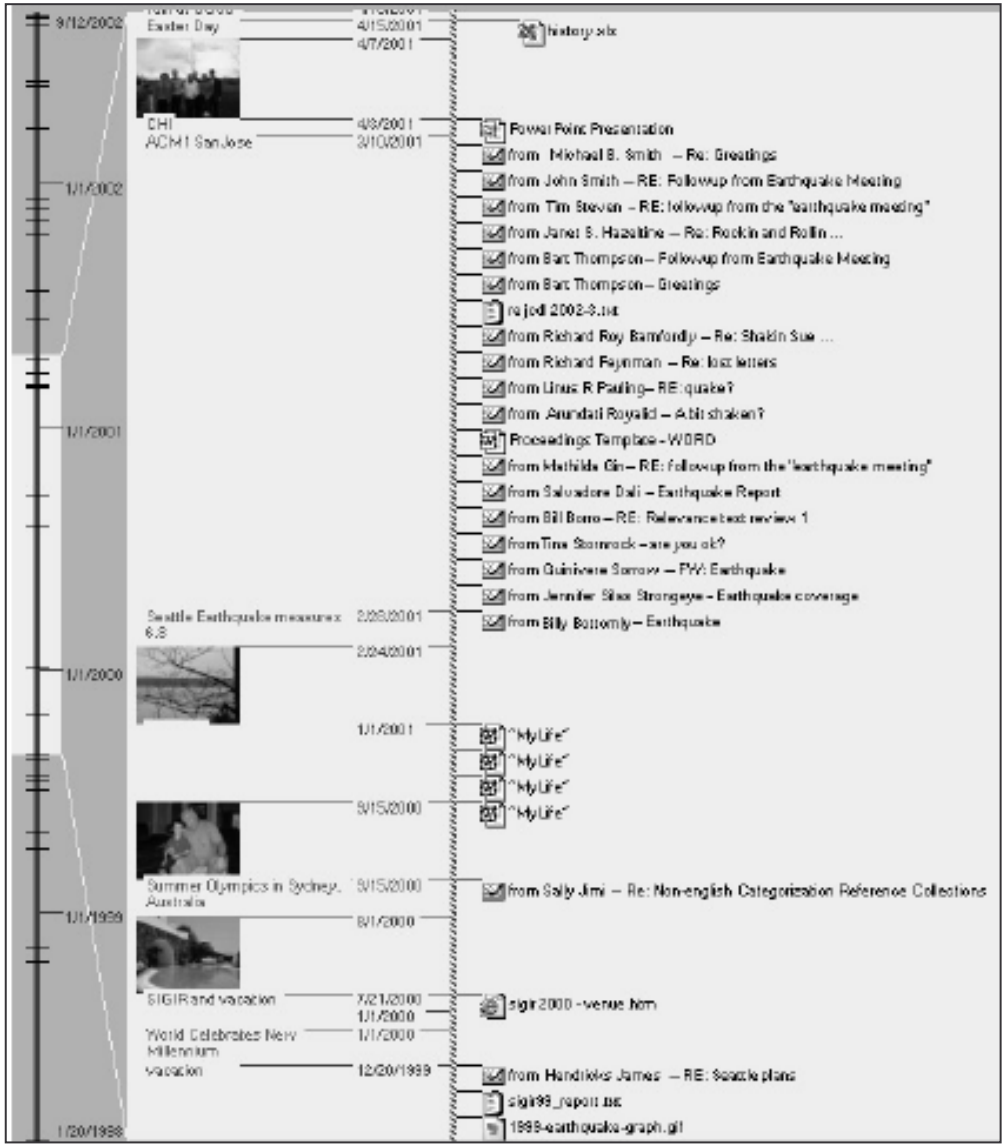
- Google
- Stuff I've Seen (2003)



Related Work: Temporal

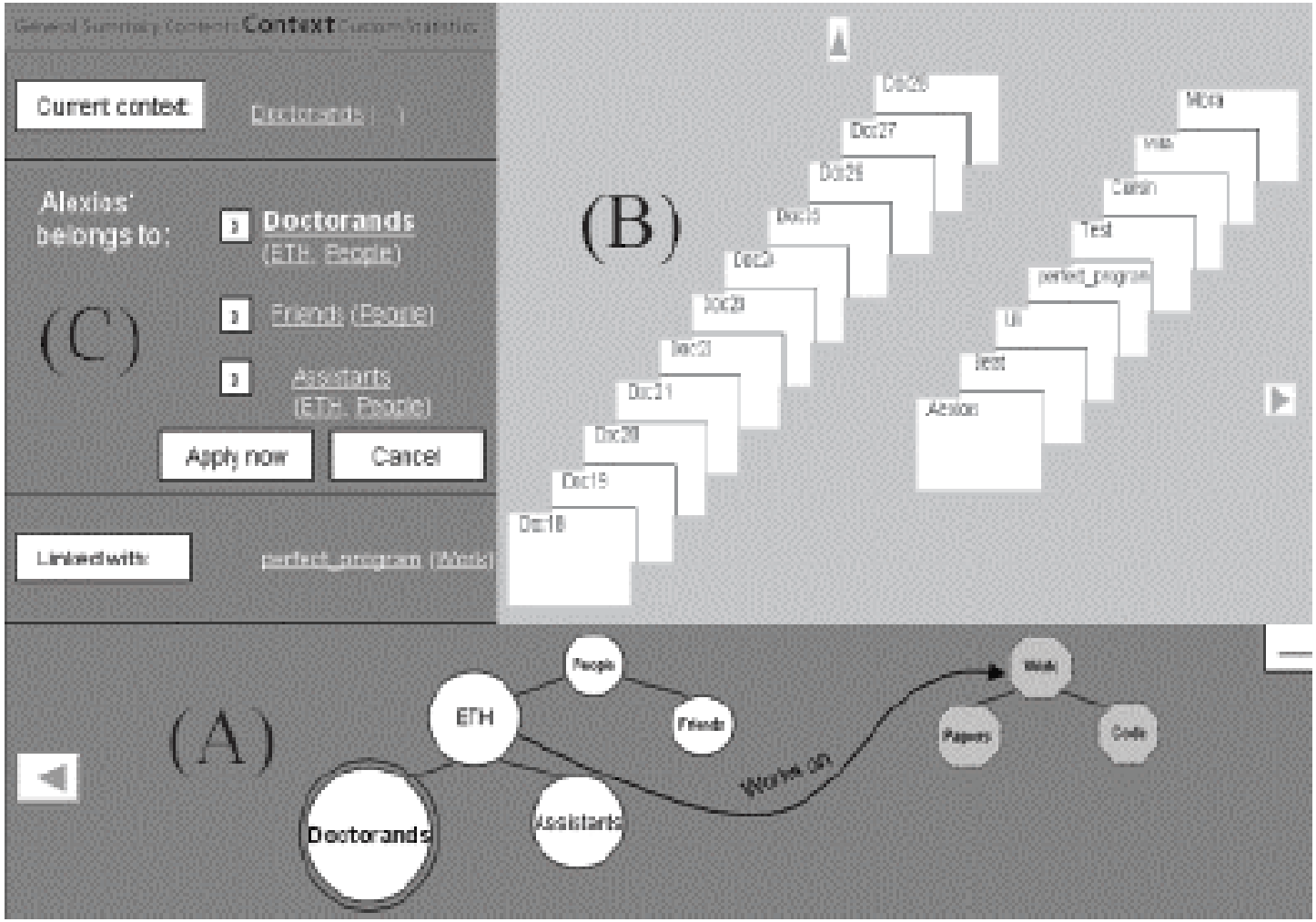
- Milestones in Time (2003)

à provides personal events as landmarks on the time line for the retrieved results

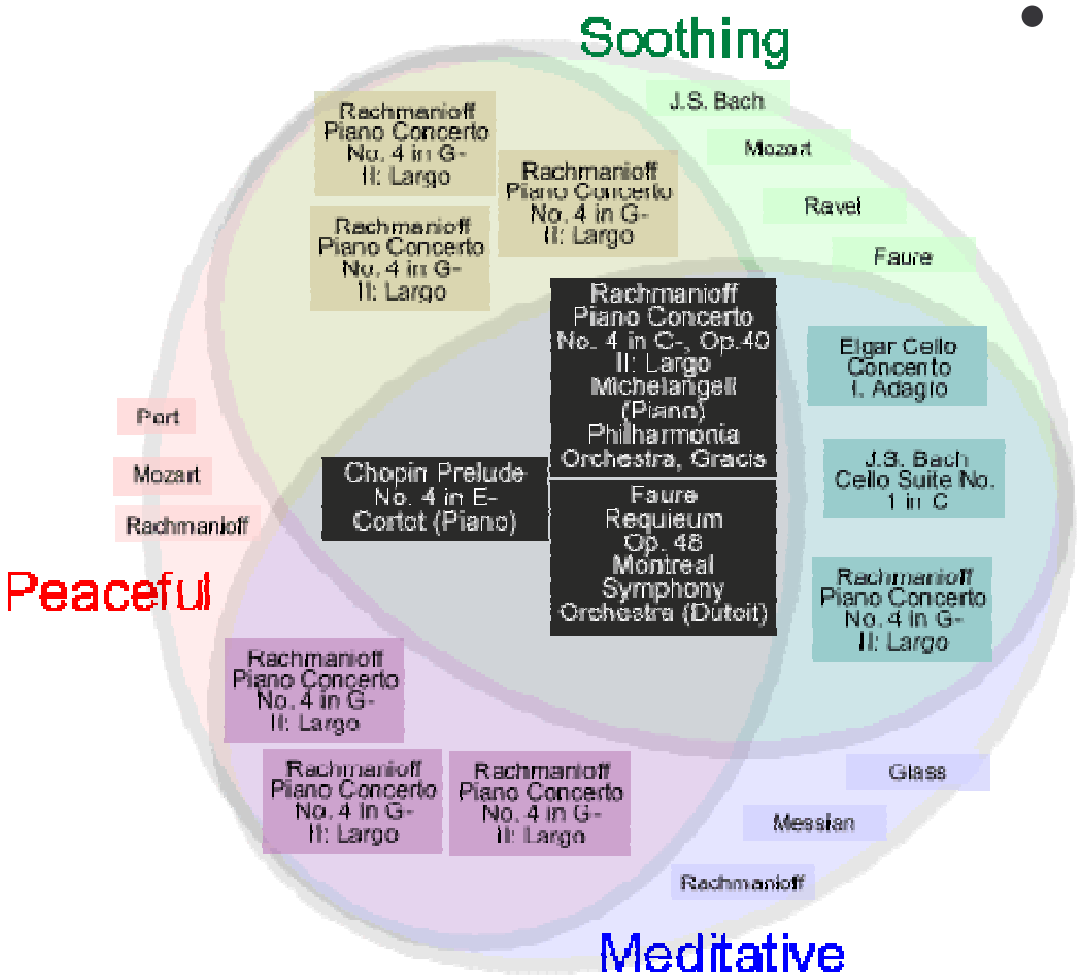


Related Work: Integrated

- InfoSpace (2003): spatial + temporal

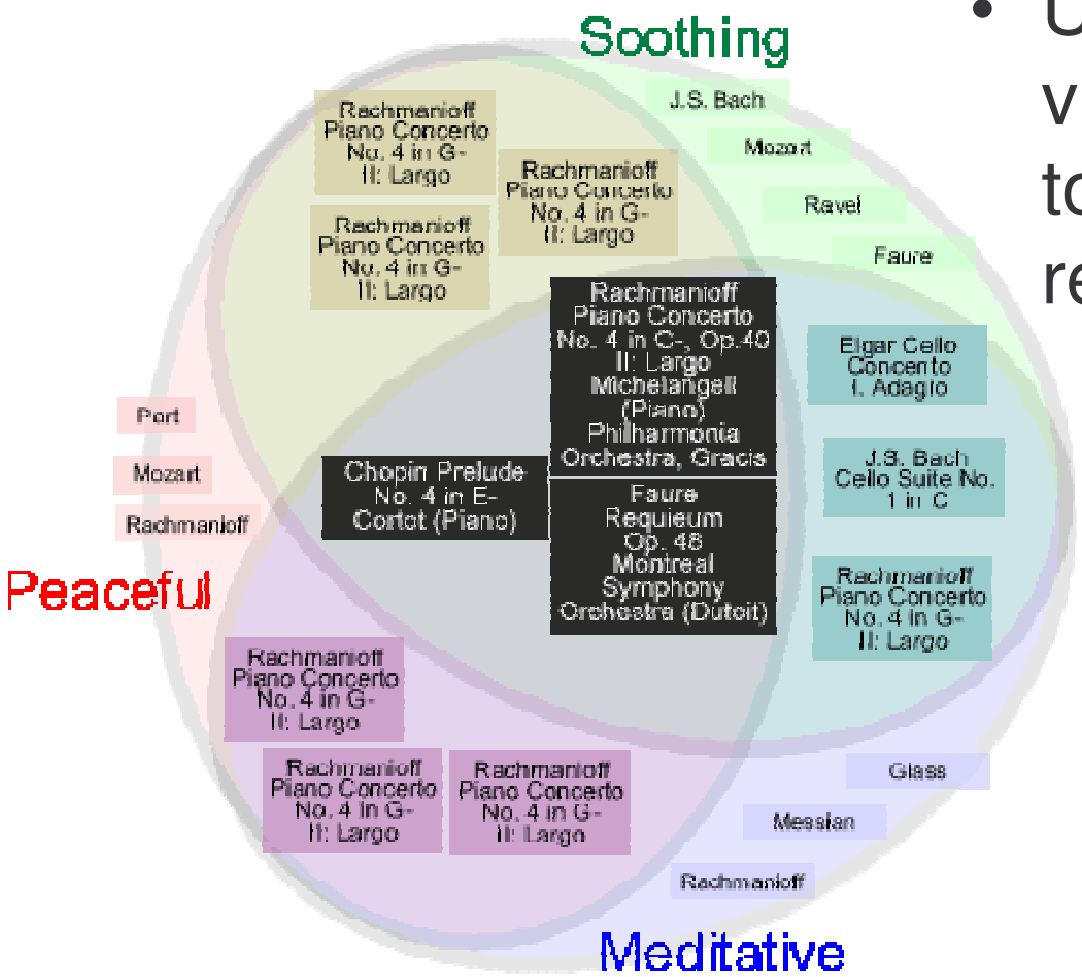


Prototype I



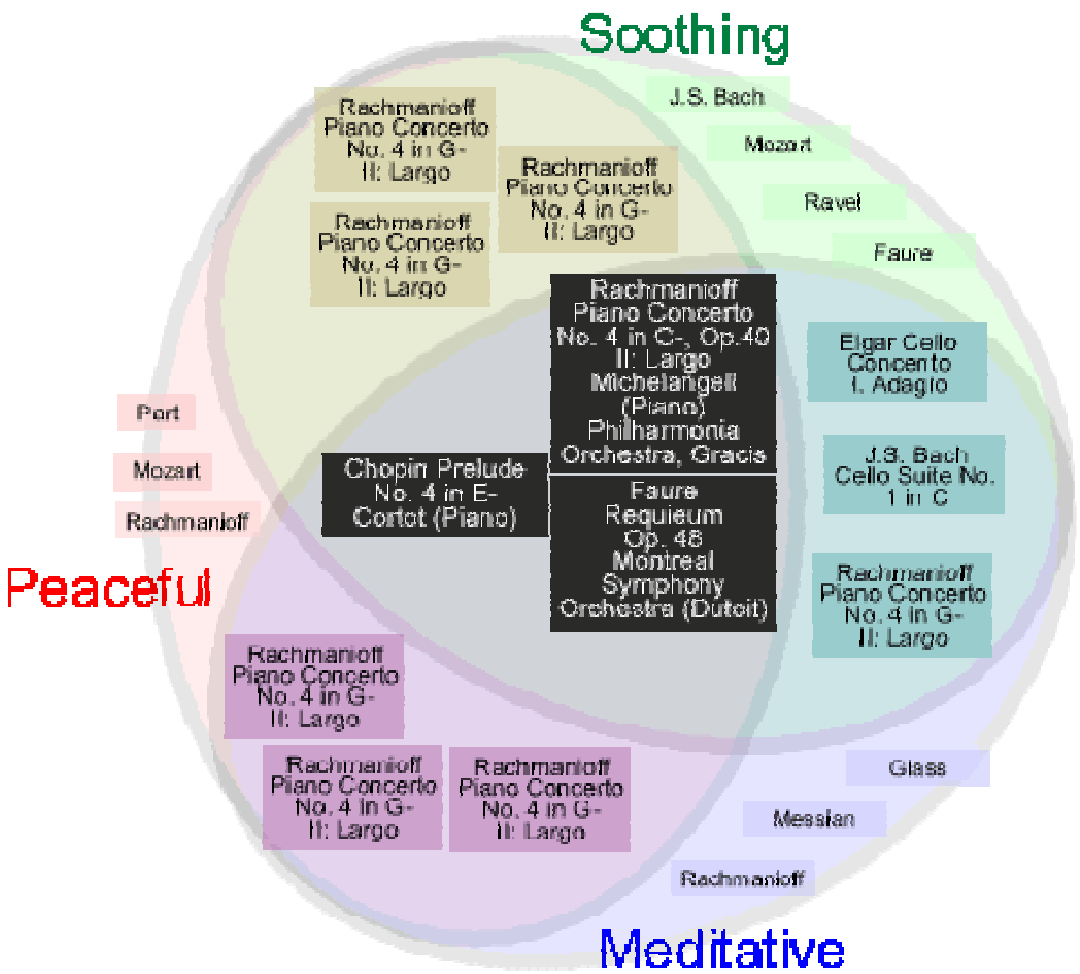
- Arranges query results as a Venn diagram
  - to put results in context of query terms
  - to relate neigbouring regions by a query term

Prototype I



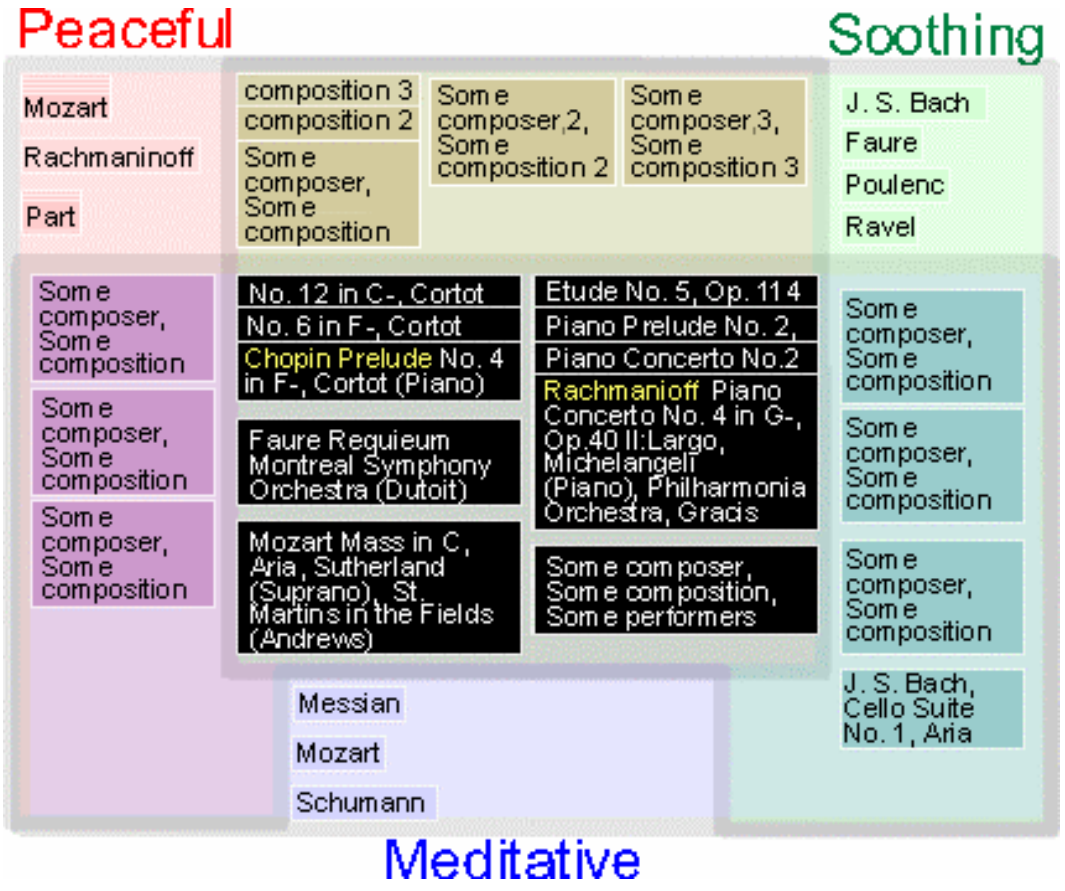
- Uses a number of visualization techniques to convey these relationships
  - Colour-coding the search word with primary colours, and the cross-area with a mix of those colours
  - Perceptual Layering to indicate the relative importance of each result region

Prototype I



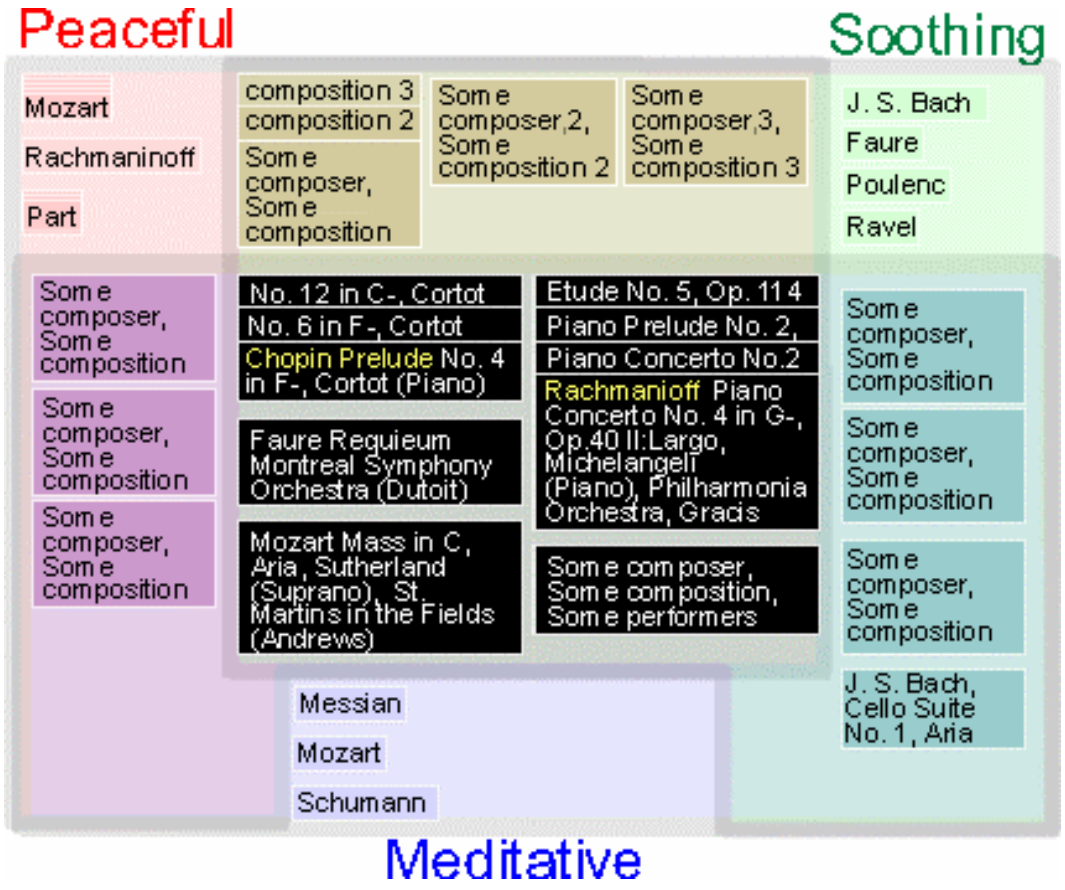
- Difficult to pack squares into non-rectangular containers
  - à Limits max display capacity
- Can further cluster music record displays

Prototype II



- Uses rectangular containers
  - Harder to see “Venn” relationships, but still relates neighbours with a single query term
- “Piles” music by composer (or artist, genre, style)

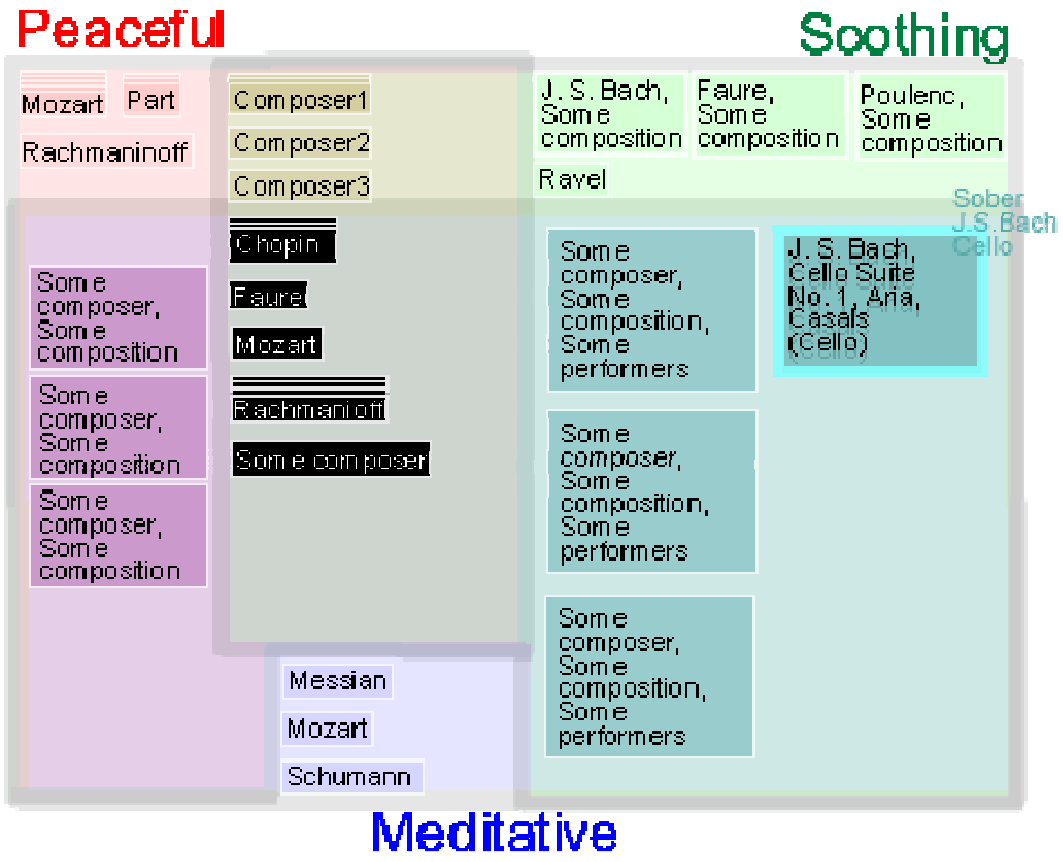
Prototype II



- Semantic zooming

1. Full display (composer, title, performer)
2. Partial display (composer, title)
3. Minimal display (composer)
4. No display: number of results

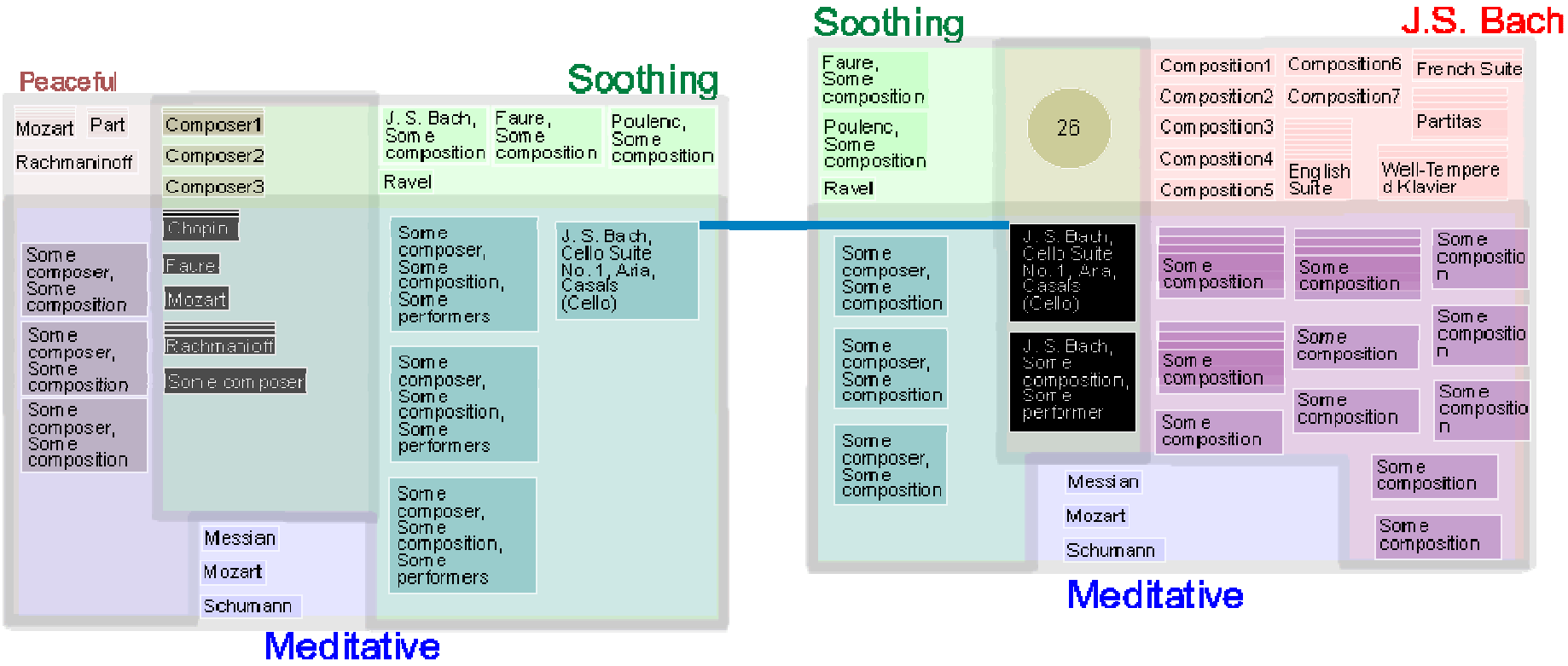
Prototype II



- Semantic zooming

1. Full display (composer, title, performer)
2. Partial display (composer, title)
3. Minimal display (composer)
4. No display: number of results

Prototype II: New Query



- New query is an “extension” of old, linked by line, colour, and position
- Old queries fade and shrink with time

Dataset

- 8556 mp3 files extracted from 714 albums by 315 different artists
- Rock/pop and electronica
- Labeled with English terms (by Eric Brochu)

ALB	Fever to Tell
ART	Yeah Yeah Yeahs
REL	Apr 29, 2003
GEN	Rock
STY	Indie Rock, Garage Punk
TON	Cathartic, Exuberant, Boisterous, Passionate, Brittle
PAT	/cs/beta/SCRATCH/music/mp3library/Yeah Yeah Yeahs/Fever to Tell

Implementation

- Architecture
  - Flat (at the moment): since the amount of data processing required is not extensive
- Platform and language:
  - Java using Eclipse IDE on Windows
- Libraries
  - swt.jar
  - No other graphics library used (yet…)

Current status & Next steps

7	8	9	10	11	12	13
Familiarize with database structure, refine prototype design						
14	15	16	17	18	19	20
Implement basic layout and individual element selection						
21	22	23	24	25	26	27
Implement semantic zooming, F+C with animation						
28	29	30	1	2	3	4
Implement new keyword query (spatial layout)						
5	6	7	8	9	10	11
Implement new keyword query (animation)						
12	13	14	**15**			
Preparation of report and presentation						

Demo