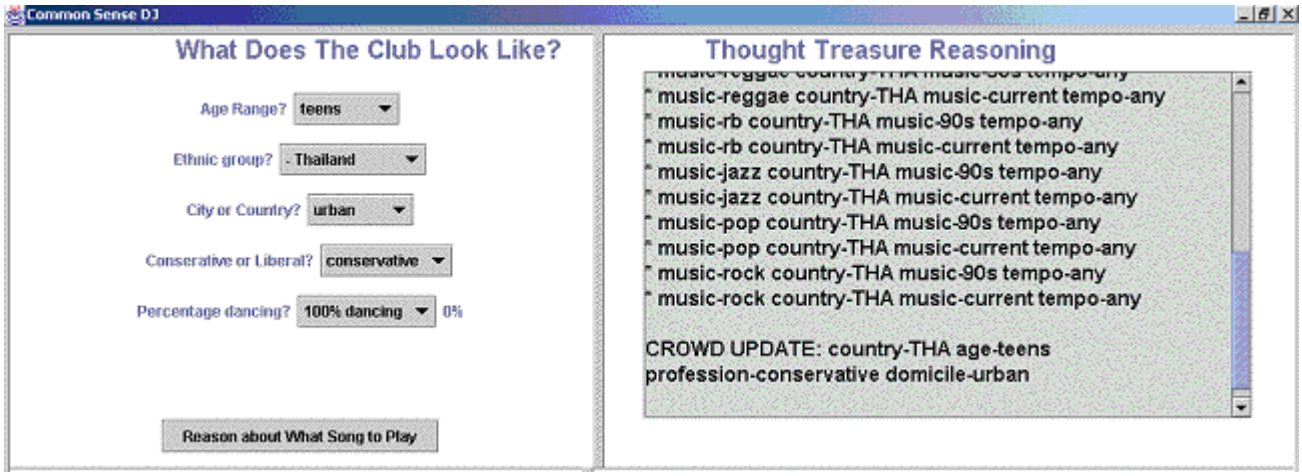
The Common Sense DJ

Joan Morris-DiMico, Carla Gomez, Roger Sipitakiat, Luke Ouko



In this paper, we introduce the Common Sense DJ (CSDJ), a technology to aid DJs in music selection. It uses ThoughtTreasure (TT) as a common sense database to aid in selecting the optimal music for a crowd given certain parameters about the crowd. ThoughtTreasure handles the reasoning aspect and has *critics* to facilitate in making better decisions. A camera captures the crowds' music response level. The CSDJ responds by going into TT's database finding the necessary common sense knowledge and validating its usefulness through the TT Prover. Given certain goals the application is able to learn. We demonstrate that using common sense database such as ThoughtTreasure we can develop more intelligent applications.

The Common Sense DJ

Joan Morris-DiMico, Carla Gomez, Roger Sipitakiat, Luke Ouko

ABSTRACT

In this paper, we introduce the Common Sense DJ (CSDJ), a technology to aid DJs in music selection. It uses ThoughtTreasure (TT) as a common sense database to aid in selecting the optimal music for a crowd given certain parameters about the crowd. ThoughtTreasure handles the reasoning aspect and has *critics* to facilitate in making better decisions. A camera captures the crowds' music response level. The CSDJ responds by going into TT's database finding the necessary common sense knowledge and validating its usefulness through the TT Prover. Given certain goals the application is able to learn. We demonstrate that using common sense database such as ThoughtTreasure we can develop more intelligent applications.

Keywords

Common sense, ThoughtTreasure, Machine Learning, Disk Jockey (DJ)

INTRODUCTION

The Common Sense DJ is an application employing techniques of common sense to build a play list for a dance club environment. It uses a situation's current context to suggest the best decision for the environment. As the context changes over time, the application continually updates its suggestions, resulting in an application that is flexible enough to adapt to and learn from its prevailing context.

Within the domain of a dance club setting, this means the CSDJ observes the current population of the club: "Who are the people? What do they look like? What are they doing?" And thereafter makes a suggestion about the best type of music for the crowd. Imagine the scenario of a dance club in a tourist destination, where the club never knows who will show up on a given night. Tonight it might be Thai teenagers, tomorrow night it might be a mix of Europeans and Americans.

Our human common sense gives us some intuition about what kind of music these different groups might like dancing to, but it is very difficult for us to manually select the best songs on the fly. The CSDJ offers a solution to this problem, by first allowing a real-life DJ to supply basic observations to the system, and then by returning a suggestive play list of the songs that are most likely to get the crowd dancing. The application also has the ability to learn through the night, with a camera sensor detecting the number of people dancing. When the system's best suggestion does not get the crowd dancing, the system will adjust its common sense understanding of the crowd's music preference and suggest different songs to fit the given criteria and the new knowledge.

The common sense knowledge and reasoning is implemented in ThoughtTreasure [1]. ThoughtTreasure (TT) is both a common sense knowledgebase and architecture for natural language processing. We used the knowledge in TT to get facts about people and music styles and then used the TT "Prover" to make inductions about different groups and their music preferences.

THE COMMON SENSE DJ APPLICATION

Gathering Input

Data for TT comes from the Java application, both through a feedback form and through the tracking of the number of people dancing with a video camera.

The Java Interface

The graphical user interface of the CSDJ is a Java Swing application, which in addition to providing the visual interface to the system, also manages all communication between the music database, and camera application. There are four sections to the graphical interface, paralleling the communication with the different components of the CSDJ: the demographic input screen, the reasoning display, the suggested play list, and the music player. Figure 2 shows a screen shot of the interface.

The demographic input screen sends the dance club's current state to TT through TT's Java API, including the type of people in the club and the percentage of people dancing (this value is sent from Dancing Detector). The reasoning display on the upper right outputs the TT reasoning information sent to the interface. This information is then used to filter down the application's music database of wav files to a suggested play list. This play list is displayed in the lower left corner of the interface. When the songs are played, the current song being played is displayed in the lower right corner.

The Dancing Detector

To detect the percentage of people dancing in the room, we modified an existing application built by the MIT Media Lab's House_n research group. The application uses a top-down camera tracker (specifically, an Intel web-cam) for input on the total number of people in a room and the

number of those that are dancing. The application uses a real time multi-person/object tracking algorithm that has multiple hypotheses reasoning to enforce multi person match constrains. Reasoning is achieved by clustering the pixel quantities at a location and determining how the cluster is changing over time. Once the number of people and the number of those dancing is calculated, the camera application sends the percentage of people dancing to the Common Sense DJ's Java application layer. The Java application then updates its current knowledge about the dance club's state and sends it to ThoughtTreasure for suggestions.

Figure 1: Image capture and processing of camera application

ThoughtTreasure

ThoughtTreasure is the core of our application: it manages the collection and storage of common sense knowledge and performs all the reasoning on this knowledge before sending suggested music types to the Java interface. TT uses a hierarchical storage structure and further categorizes its knowledge into three categories: goals, facts, and rules. This structure simplifies the process of reasoning, by enforcing the structure that facts and rules must be combined to achieve goals. TT also has a natural language processing component, which could be, for example, used to receive open-text suggestions from people in the dance club.

ThoughtTreasure Facts

Facts in TT are what we commonly understand as facts: they are statements about relations between concepts and objects which hold true. For the purpose of our application, facts contain knowledge that is *currently* true about the environment. Thus facts can be continuously changing, and this is done through the interaction of the Java interface and TT.

We added numerous new facts into the ThoughtTreasure common sense database, supplementing its knowledge relating to people, cultures, and music. Examples of these facts:

- Culture (each continent and country within)
- Age (Teens, 20s, 30s, 40s, 50s, 60s)
- Professions (Conservative or Liberal)
- Domicile (Rural or Urban)

As mentioned, the data in TT is stored in a structured format. Thus each fact follows a particular syntax. Examples of some of the facts we added are below:

[the-people-of crowd1 Asia]

[age-of crowd1 age-teens]

[profession-of crowd1 profession-unknown]

[music-universality-of crowd1 music-culture-specific]

ThoughtTreasure Rules

Rules in TT generate all possibilities and intersects of the provided facts, drawing ties together such as if the dance club crowd has Thai teenagers, then these teenagers are also from Asia and people from Asia like different types of music from Asia, such as Thai pop music.

Sample TT Code:

[ifthen [and

[domicile-of ?crowd ?domicile]

[ne ?domicile unknown]

[music-category-of ?music-category ?domicile]

[by-domicile-isa music-genre

[music-genre-by-domicile-of-crowd ?music-genre]

]

ThoughtTreasure Goal/s

The "goals" defined within TT drive the reasoning, by defining the goal of the reasoning process. In our application, the goal is to find the best music suggestion for the current crowd.

Sample TT Code:

[music-genre-of-crowd ?music-genre ?country ?music-era ?music-tempo]

ThoughtTreasure Critics

ThoughtTreasure can quickly reason through facts and rules to achieve goals, but frequently this results in too many suggestions, because there is no higher level common sense. An example is that TT might find that Chinese people like Chinese samba. While within the structure of TT's knowledge of music and culture this is logical, it is not common sense because there is no such genre of music. Because of non-common sense results like these and because of situations where TT returns too many suggestions to be practical, we built an enhancement to ThoughtTreasure referred to as *critics*.

We use critics to make validations of the data and to reaffirm certain common sense rules. By checking TT's suggestions and reapplying certain rules, we were able to refine TT's suggestions to the Java interface.

Machine Learning

The aim of having machine learning in this application was to enable the creation of new rules that are understandable i.e. creating new common sense statements. This task would be achieved by learning actively, from prior knowledge as well as incrementally.

Our independent variables include: The demographics of the audience, the number of people dancing, the music preference of the people dancing, the details of the song

playing as well as details on location, time, event, season etc.

Our dependent variable would be the goal to be achieved e.g. Given the crowd, what is the best genre of music to play given their demographics and prevailing of people to ensure dancers > 60%

The application can use data mining techniques to learn new rules from the data.

FUTURE WORK AND CONCLUSIONS

There are many future directions we would like to take with this work. On a technical implementation level, we would like to continue working to refine and enhance ThoughtTreasure's reasoning ability. One way to do this will be to futher refine the critics, which evaluate the reasoning conclusions TT makes. Another enhancement would be to add the ability for more robust data mining in TT. It would aid in creating better rules as well as providing for a better base from which critics would work.

To further demonstrate our goal of building an application using common sense to adapt to a changing environment, we would like to incorporate other sources of common sense into the application, such as the Open Mind common sense data.

In conclusion, this application demonstrates how an application utilizing common sense can build an understanding of an environment, reason about the best decision to make, and produce an adapting list of suggested actions. Within the domain of music selection, our application is a tool for DJs to automatically filter music with common sense knowledge of what kind of people like what kinds of music.

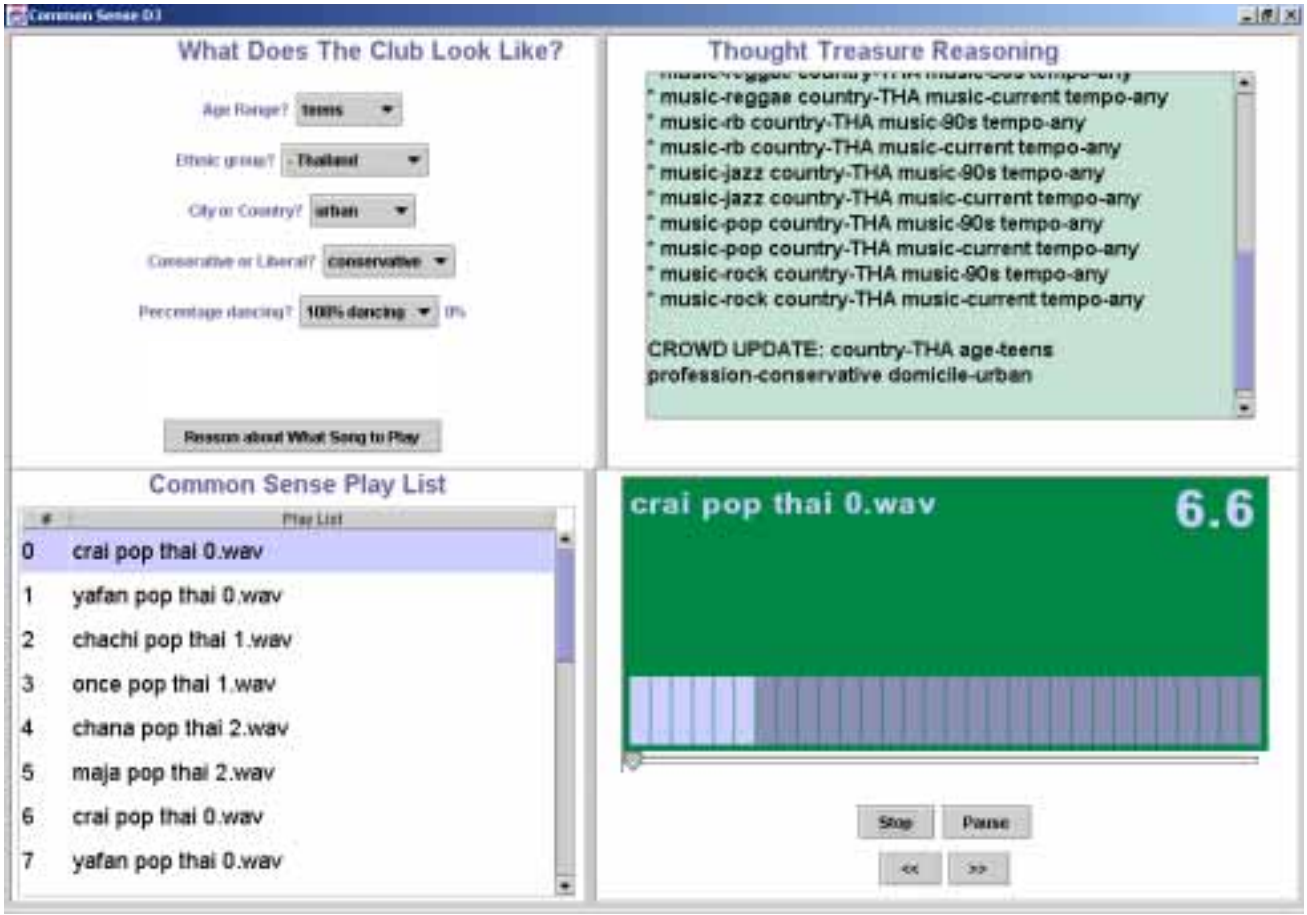
ACKNOWLEDGMENTS

We thank Henry Lieberman for teaching the course Common Sense Reasoning for Adaptive Applications at the MIT Media Lab. Push Singh, Hugo Liu and the members of the Fall 2002 class provided invaluable design suggestions for this project. Lastly, our project could not have been completed without the tireless efforts of Eric Mueller, creator and designer of ThoughtTreasure.

REFERENCES

1. Mueller, Erik T. (1998). Natural language processing with ThoughtTreasure. New York: Signiform. Available: http://www.signiform.com/tt/book/.
2. Thuraisingham., B., *Data Mining, Technologies, Techniques, Tools and Trends*. CRC Press, Boca Ratona, Florida, 1999.

Fig. 2



The Common Sense DJ

Arnan (Roger) Sipitakiat
Carla Gomez Monroy
Joan Morris DiMicco
Luke Ouko

MAS.964
Final Project
December 2002

Project Goals

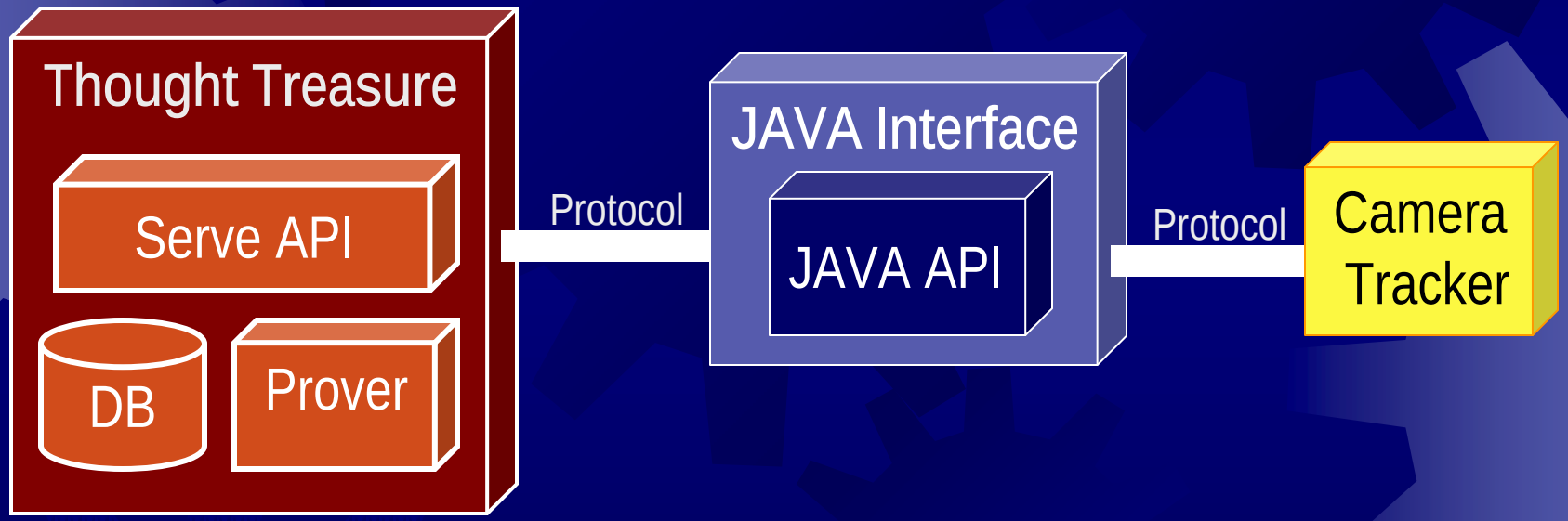
Create a reasoning system that:

- Utilizes common sense knowledge from Thought Treasure.
- Adapts suggestions to the current environmental context.
- Observes reactions to suggestions to learn new or corrective CS.

Overview of the Common Sense DJ

- CSDJ application suggests songs to play through common sense
- Thought Treasure = knowledge source
- Thought Treasure reasons about what song type to play
- Java interface collects feedback from real-life DJ and suggests songs
- Camera senses dancing, allows feedback to Thought Treasure

CSDJ Architecture



Thought Treasure

- Hierarchical knowledge storage structure
- Primary features: NLP, Spatial representation, planning.
- Provides simple rule-based reasoning engine

Music Categorization

- By Culture
  - By Continent: Asian, European, etc.
  - By Country: American, Mexican, Thai, etc.
- By Age
  - Teens, 20s, 30s, 40s, 50s, 60s
- By Profession
  - Classic (Conservative), Artistic (Liberal)
- By Domicile
  - Rural, Urban

Preliminary Reasoning

- 59 countries in Asia x 5 music eras x 18 music genres
- 5,310 possibilities
- When all attributes are known, rules can filter this down to 3-10 possibilities.

Preliminary Reasoning (examples)

- A liberated crowd in their 20s from an urban part of Mexico probably likes: Mexican salsa, electro, alternative rock.
- Conservative Americans in their 50s from an urban city probably likes: rock music from the 60s and 70s (Elvis, the Beatles)

Need for further reasoning

- Too much data and conflicting data when some attributes are missing.

Further Reasoning: Prover Critics

- Analyzes the preliminary output and detect situations when the output is useless or self-conflicts.
- Then, it goes through a set of scenarios to improve the output.

Examples of Scenarios

- While Culture is unknown. It is better to play cross-culture music than to guess.
- If profession or domicile is unknown then try to guess.
- If all attributes are known but people are not dancing then:
  - Try to increase the tempo.
  - Some attributes may not be true anymore.

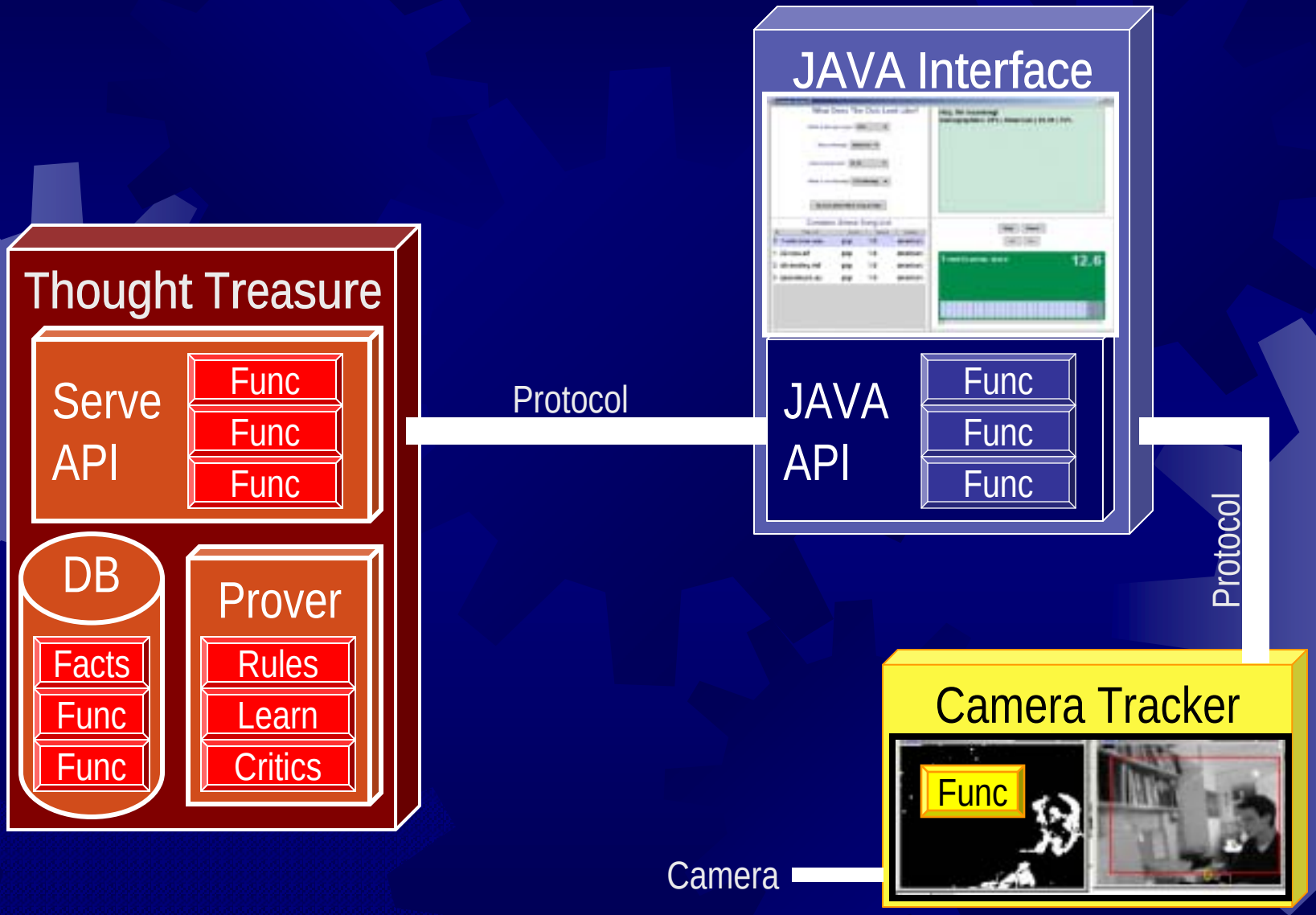
Further Reasoning: The Learning Critic

- A tracking system provides feedback data upon which the system reflects its decision.
- new rules are added when feedback differs from current rules.

Conclusions about TT

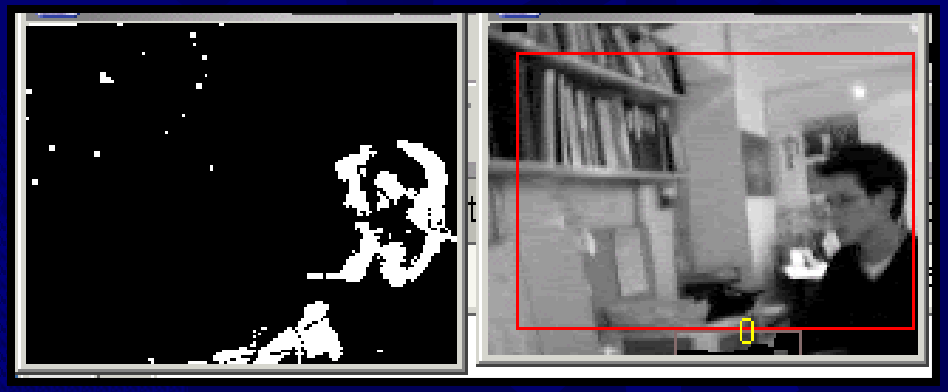
- Chosen because of the built-in structure and reasoning
- Structure restrictive, not enough knowledge
- With Mueller's help, extended TT, extended the Java API, and fixed bugs

Technical Implementation

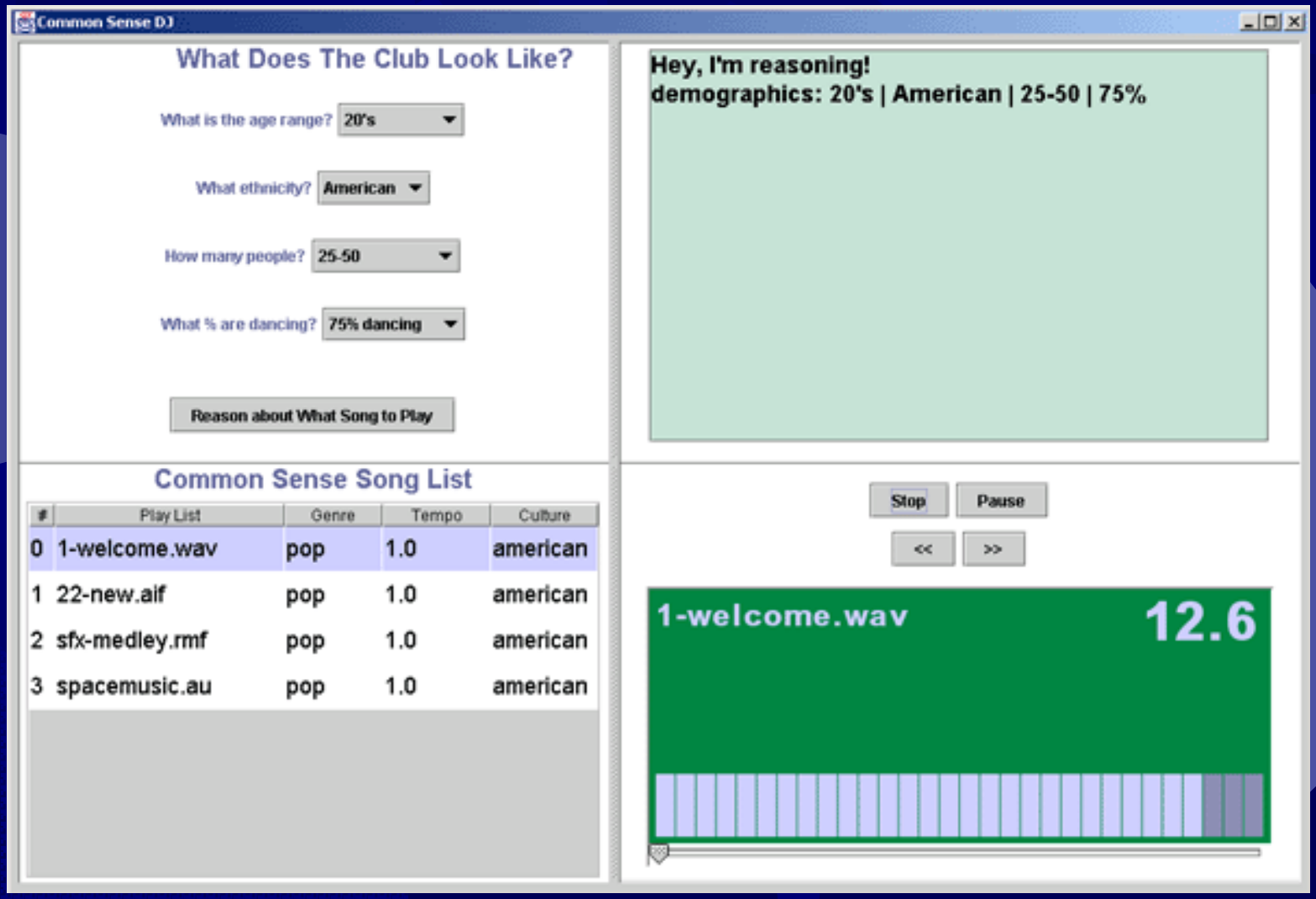


Camera Sensor

- House_n technology
- Detects number of people in view and number dancing
- Sends feedback to Common Sense DJ for learning



Demo!


Common Sense DJ
What Does The Club Look Like?
What is the age range? 20's
What ethnicity? American
How many people? 25-50
What % are dancing? 75% dancing
Reason about What Song to Play
Hey, I'm reasoning!
demographics: 20's | American | 25-50 | 75%
Common Sense Song List
Play List Genre Tempo Culture
0 1-welcome.wav pop 1.0 american
1 22-new.aif pop 1.0 american
2 sfx-medley.rmf pop 1.0 american
3 spacemusic.au pop 1.0 american
Stop
Pause
<<
>>
1-welcome.wav
12.6

Conclusions

- Built application using TT's knowledge and reasoning power
- The CSDJ builds suggested play list based on dance club's appearance
- System refines TT's CS knowledge based on crowd's reaction to songs

Thanks!

Arnan (Roger) Sipitakiat

Carla Gomez Monroy

Joan Morris DiMicco

Luke Ouko