

Artificial Intelligence Coursework

Guide to electronic music in Frankfurt/Germany

Steffen Grunwald
Student at
Staffordshire University
gq923377@staffs.ac.uk

1. Abstract

Business processes consist almost always of so called expert systems, which decide based on known facts and knowledge from experts e.g. what to buy, whom to hire, and what to do. Some of these expert systems need to be controlled by humans, some can be approximated by a program using artificial intelligence.

This document describes the planning and implementation of such a system which recommends a night-club to a user based on his music and style preferences, and finances.

2. Purpose of the system

This system should help a user to decide where to spend the evening at his/her weekend. After using the program, the user knows, if there are clubs that fulfill all requirements and if yes, where he can find more information about them (using the internet URL). The results are presented with a summary of which criteria he has entered to come to that decision.

The decision is based on three factors:

- Music preference
The (electronic) music, that he expects is played in that club.
- Club style
The style of the club and the people. Some people prefer clubs with the cleanliness of a clinic, some of them like it dark and shabby to get the right mood.
- The price category of the club
There are cheap and expensive clubs. It is the task of the program to find out, what the user prefers. If a users has chosen a category also cheaper categories apply.

The systems asks questions to find out the preferences of the user.

The price category

It is known that the prices for a club differ from day to day, but mostly fit into the categories "cheap", "fair" and "expensive". The user is asked to enter a price he is willing to pay for his evening pleasure. Using this information, the system finds out, to which category that fits – but it also choses clubs, which are in cheaper categories.

The music preference

It is not user-friendly to expect from the user to know all music styles – a user rather knows the artists' or producers' names from hearing them in the radio.

But you can't assign every artist to each club, where his style is played. That's

why only the styles are attributes of each club and the system tries to figure out, what style the user likes. This is done by suggesting some names of producers. The user can decide for one of them and thus states his style preference.

The atmosphere

It is a matter of individual taste, how the perfect club looks like. Some users prefer a clinic like atmosphere and a dress-code like on the oscar-awards, some like the ambiance of a subway station – thus, the user is asked for the desired atmosphere of the club: “shabby”, “neutral” or “snobby”.

The target user

The target user is a person who likes electronic music and is familiar with some artists in this genre and has an idea of a nice club. Someone, who want to have a good night but is just on a journey, visiting Frankfurt and its clubs for the first time.

3. Knowledge representation

The guide to electronic music uses a semantic network. The representation with the u60311 is shown in illustration 1.

It is clear, which objects are related to other objects. In the illustration example, the user has stated, that he likes the music of Armand van Helden, likes a shabby club and is willing to spent 15 Euro.

Please note that not all styles in the *playsStyle* list and not all artists are listed for brevity.

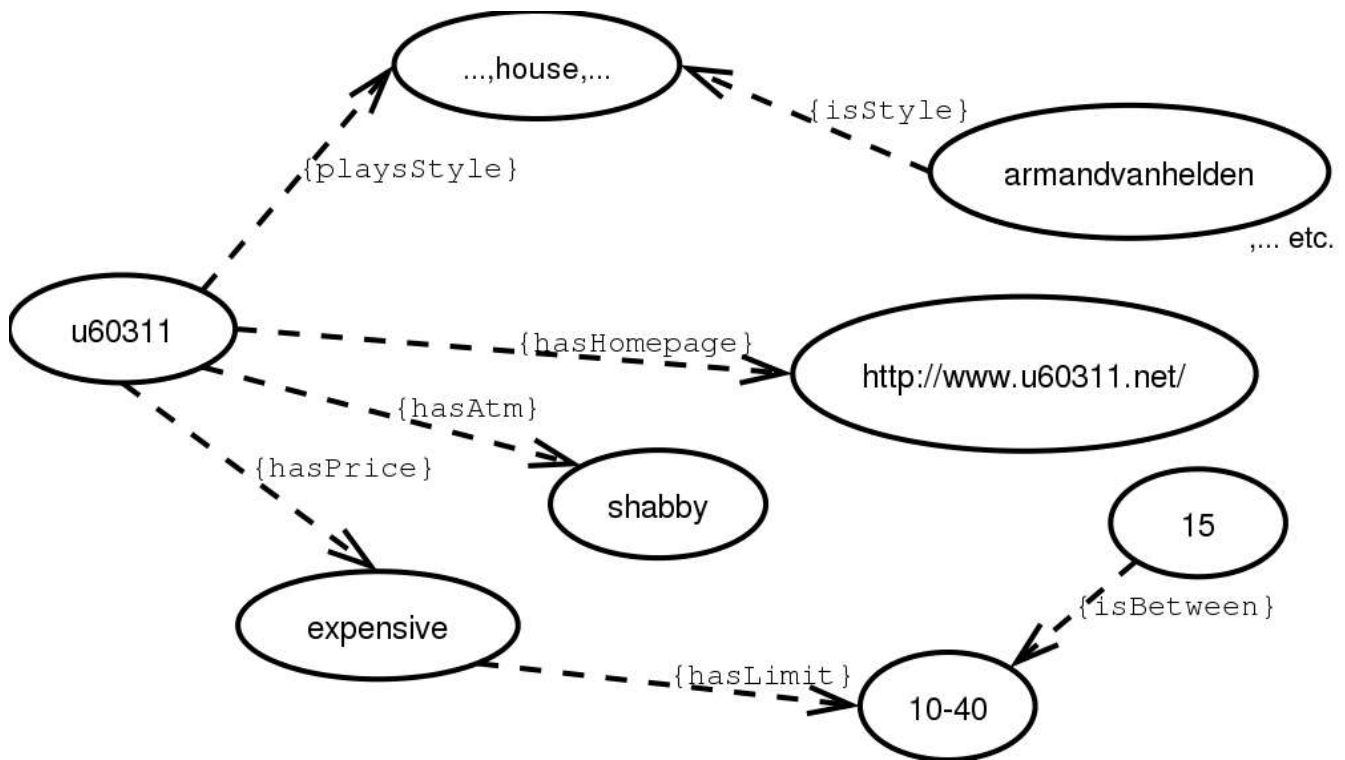


Illustration 1 Semantic net with e.g. the club u60311

Please take some time to mind the knowledge representation of illustration 1 at the end of the prolog program listed in the appendix. You can easily recognize the attributes of the clubs and e.g. the relationship to the artists/producers.

4. Knowledge representation criteria

The representation of the knowledge as a semantic net in this domain is chosen because of its advantages compared to its minor limitations.

Advantages

The representation is quite natural with sentences like “u60311 playsStyle house”.

Also, it is easy to extend. If further details should be involved in the reasoning process, you could for example add: *metersFromRailway(u60311,200)*..

Inheritance could be implemented later, if someone would like to have a fully blown guide to entertainment: Super objects like *club*, *theater*, *pub* could be added. Their properties could be inherited with: *isType(u60311,club)*..

The reasoning is pretty fast because of the simple and flat representation.

Limitations

It is hard to represent exceptions in the knowledgebase. For example: The u60311 plays only techno at Fridays. There are different ways to express these exceptions, but it would mean to extend the knowledge base and change its

structure.

An other limitation is that you can not express easily that a style is not played: *playsStyle(U60311,not(pop))*. won't work. This requirement also would imply a change on the knowledge base representation.

5.Implementation details

Implementation phases

During all of the following phases, the system is constantly tested.

Problem definition

First, the problem is analyzed and the user requirements are analysed: What does a user want and what can he state to come to a conclusion?

Expert knowledge definition

What would a expert know, and how is the information related? The knowledgebase is formed, first only facts without a representation form – later semantic network is chosen.

Creation of dialogs

All dialogs are created taking robustness and human factors into account (e.g. a user could enter "4 EUR" instead of simply "4").

Creation of reasoning

With all information from the dialogs, the decision for a club can happen. The structure for that is implemented.

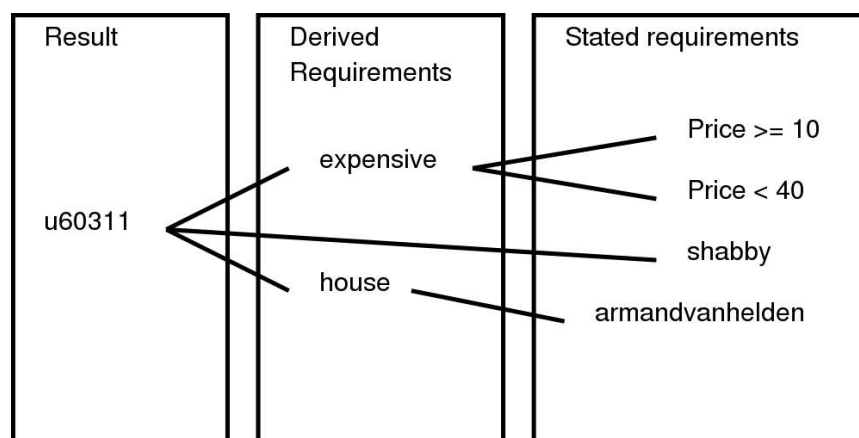


Illustration 2Reasoning

Result output

The output of the result is implemented.

Testing and debugging

All work is tested in the end and minor changes are done to the program. Please see the next chapter for test runs.

User interface

The user interface consists of dialogs and is basically multiply choice based.

Every time (except the price dialog), the system shows all possible choices.

If a user makes a typo or misunderstands the possible inputs, an error message informs about the wrong input and demands another try.

No prolog knowledge is required to use the program, only a double click or a simple consult command is required to load the program. After that, only "go." starts the dialog mode.

6. Test runs / Demonstration

Program start

This shows the start of the program (here with Linux – but applicable to other operating systems).

```
$ swipl
Welcome to SWI-Prolog (Multi-threaded, Version 5.2.7)
Copyright (c) 1990-2003 University of Amsterdam.
SWI-Prolog comes with ABSOLUTELY NO WARRANTY. This is free software,
and you are welcome to redistribute it under certain conditions.
Please visit http://www.swi-prolog.org for details.
```

```
For help, use ?- help(Topic). or ?- apropos(Word).
```

```
?- consult(clubguide).
% clubguide compiled 0.00 sec, 12,576 bytes
```

```
Yes
?- go.
```

```
*****
*
*   Welcome to the interactive clubguide to Frankfurt   *
*
*           it will find the appropriate CLUB          *
*           for your mood, style preferences           *
*           and finances :-)                          *
*
*           Great stuff!!!                             *
*****
```

```
How much money are you going to spend tonight?? (0-40 EUR)
```

|:

Wrong input/ no result

This test runs shows all types of inappropriate input and the program's reaction – and a test run that shows no result (both in one run, for brevity).

Every wrong input makes the program ask the question again.

```
How much money are you going to spend tonight?? (0-40 EUR)

| 50.
Hey!! Don't cheat on me, that didn't really look like a number between 0 and 40, did it? Please
      retry...

How much money are you going to spend tonight?? (0-40 EUR)

| noNumber.
Hey!! Don't cheat on me, that didn't really look like a number between 0 and 40, did it? Please
      retry...

How much money are you going to spend tonight?? (0-40 EUR)

| 10.
Please enter your favourite artist from the following List

derdritterraum
[...]
taucher

| noArtist.
Hey!! Did you really saw that artist in this list? I don't think so, please try again!

Please enter your favourite artist from the following List

derdritterraum
[...]
taucher

| taucher.
Please choose your favourite atmosphere from the following list.

snobby
shabby
neutral

| noAtmosphere.
Hey!! Did you really saw that in this list? I don't think so, please try again!

Please choose your favourite atmosphere from the following list.

snobby
shabby
neutral

| snobby.
Sorry, I found no club for your entertainment

Thanks for your attention
```

Run with one result

The input of the user in this test run will find one matching club.

```
How much money are you going to spend tonight?? (0-40 EUR)
```

```
| 10.  
Please enter your favourite artist from the following List
```

```
derdritteraum  
[...]  
scooter  
taucher
```

```
| scooter.  
Please choose your favourite atmosphere from the following list.
```

```
snobby  
shabby  
neutral
```

```
| shabby.  
You looked for the expensive club where people dance to rave in a nice and shabby atmosphere.  
I found the following clubs for your individual evening pleasure:
```

```
index - more info: http://www.index-ffm.de/
```

```
Thanks for your attention
```

Run with several results

The input of the user in this test run will find several matching clubs.
How much money are you going to spend tonight?? (0-40 EUR)

```
| 7.  
Please enter your favourite artist from the following List
```

```
derdritteraum  
[...]  
stgermain  
taucher
```

```
| stgermain.  
Please choose your favourite atmosphere from the following list.
```

```
snobby  
shabby  
neutral
```

```
| snobby.  
You looked for the fair club where people dance to newjazz in a nice and snobby atmosphere.  
I found the following clubs for your individual evening pleasure:
```

```
oneninetyeast - more info: http://www.190east.de/
```

```
kingkamehameha - more info: http://www.king-kamehameha.de/
```

```
Thanks for your attention
```

7. Appendix

The Prolog code

```
/*--- Interpret Start Command ---*/  
go:-  
    showscreen,  
    askPriceCategory(PriceCategory),!,  
    askForMusicStyle(MusicStyle),!,  
    askForAtmosphere(Atmosphere),!,  
    outputResults(PriceCategory,MusicStyle,Atmosphere),  
    write('Thanks for your attention').
```

```

/*----- DIALOGS -----*/

/*--- show a welcome screen ---*/

showscreen:-    nl,nl,nl,
                write('*****'),nl,
                write('*'),nl,
                write('* Welcome to the interactive clubguide to Frankfurt *'),nl,
                write('*'),nl,
                write('*           it will find the appropriate CLUB *'),nl,
                write('*           for your mood, style preferences *'),nl,
                write('*           and finances :-) *'),nl,
                write('*           Great stuff!!! *'),nl,
                write('*****'),nl,nl,nl.

                /* On a wrong input, the method should show, that the answer was not valid
                and he should try again */

/*--- Ask the user for the money, he wants to spend ---*/

                askPriceCategory(PriceCategory):-
                    write('How much money are you going to spend tonight?? (0-40
EUR)'), nl, nl,
                    read(Price),
                    integer(Price),
                    hasLimits(PriceCategory,LowerLimit,UpLimit),
                    Price >= LowerLimit,
                    Price < UpLimit;
                    write('Hey!! Don\'t cheat on me, that didn\'t really look like a
number between 0 and 40, did it? Please retry...'),nl,nl,
                    askPriceCategory(PriceCategory).

/*--- Ask the user for the musicproducers, he likes ---*/
                askForMusicStyle(MusicStyle):-
                    write('Please enter your favourite artist from the following
List'),nl,nl,
                    printAllMusicians, nl, nl,
                    read(Producer),
                    isStyle(Producer,MusicStyle);
                    write('Hey!! Did you really saw that artist in this list? I don\'t
think so, please try again!'),nl,nl,
                    askForMusicStyle(MusicStyle).

/*--- Ask the user what atmosphere he likes ---*/
                askForAtmosphere(Atmosphere):-
                    write('Please choose your favourite atmosphere from the following
list. '),nl,nl,
                    printAllAtmospheres, nl, nl,
                    read(Atmosphere),
                    isAtmosphere(Atmosphere);
                    write('Hey!! Did you really saw that in this list? I don\'t think
so, please try again!'),nl,nl,
                    askForAtmosphere(Atmosphere).

/*--- Present the results to the user ---*/
                outputResults(Category,MusicStyle, Atmosphere):-
                    findall(Club, findClub(Club,Category,MusicStyle,Atmosphere),

```

```

Clubs),
    not(Clubs = []),
    noRepetition(Clubs, ClubsNoRep),
    write('You looked for the '),
    write(Category),
    write(' club where people dance to '),
    write(MusicStyle),
    write(' in a nice and '),
    write(Atmosphere),
    write(' atmosphere. '),nl,
    write('I found the following clubs for your individual evening
pleasure: '),nl,nl,
    write_clubs(ClubsNoRep),nl,nl;

    write('Sorry, I found no club for your entertainment'),nl,nl.

/*--- Find all clubs, satisfying the criteria ---*/
    findClub(Club, Category, MusicStyle, Atmosphere):-
        hasPrice(Club, ClubCategory),
        matchPrice(ClubCategory,Category),
        hasAtm(Club, Atmosphere),
        playsStyle(Club,StyleList),
        member(MusicStyle,StyleList).

    /* is the club "Category" in the expensive category? */
    matchPrice(Category,expensive):-
        Category = expensive;
        matchPrice(Category,fair).

    matchPrice(Category,fair):-
        Category = fair;
        matchPrice(Category,cheap).

    matchPrice(Category,cheap):-
        Category = cheap.

/*----- GENERAL FUNCTIONS -----*/

/*--- Prints a list of all musicians ---*/
printAllMusicians:-
    findall(Producer, isStyle(Producer,_), ProducerList),
    lists all cities in database ---*/
    noRepetition(ProducerList, ProducerListNoRep),!,
    *--- remove duplicates ----*/
    write_elt(ProducerListNoRep),nl,nl.
/*--- output of list ---*/

/*--- Prints a list of all atmospheres ---*/
printAllAtmospheres:-
    findall(Atm, hasAtm(_,Atm), AtmList),
    cities in database ---*/
    noRepetition(AtmList, AtmListNoRep),!,
    duplicates ----*/
    write_elt(AtmListNoRep),nl,nl.
/*--- output of list ---*/

/*--- repetition removal from lists ---*/

noRepetition([], []).
noRepetition([H| T1], T2):-
    member(H, T1),
    existing in tail, */
    noRepetition(T1, T2). /* don't copy that to
norepetition */
noRepetition([H| T1], [H| T2]):-
    not(member(H, T1)),
    copy it */
    not(member(H, T1)),
    copy it */

```

```

noRepetition(T1, T2).

/*--- write down the elements of a list ---*/
write_elt([]).
write_elt([H|T]):-      write(H),nl,write_elt(T).

/*--- write down the list of clubs with their related homepage ---*/
write_clubs([]).
write_clubs([H|T]):-   write(H), write(' - more info: '),
                        hasHomepage(H,Page),
                        write(Page),nl,nl,write_clubs(T).

/*--- check if it is really a atmosphere ---*/
isAtmosphere(Atmosphere):-
                        hasAtm(_,Atmosphere).

/*----- KNOWLEDGEBASE -----*/

/*
    Clubs are:
        u60311
        deelight
        190east
        kingkamehameha
        index
        robertjohnson
        o25

    price categories are:
        cheap
        fair
        expensive

    (electronic) music styles are:
        house
        techno
        electro
        chill

    atmosphere can be:
        snobby
        neutral
        shabby

    songs:
        derdritterraum
        chrisliebing
        armandvanhelden
        stgermain

*/

/*--- Define Clubs' Price ---*/

hasPrice(oneninetyeast,fair).
hasPrice(kingkamehameha,fair).
hasPrice(deelight,cheap).
hasPrice(index,expensive).
hasPrice(u60311,expensive).
hasPrice(o25,cheap).
hasPrice(robertjohnson,fair).

```

```

/*--- Define Clubs' Music Style ---*/

    playsStyle(oneninetyeast,[house,newjazz]).
    playsStyle(kingkamehameha,[house,chill,newjazz]).
    playsStyle(deelight,[house]).
    playsStyle(index,[techno,electro,chill,rave]).
    playsStyle(u60311,[house,techno,electro,techhouse]).
    playsStyle(o25,[techno,techhouse]).
    playsStyle(robertjohnson,[techno,electro,techhouse]).

/*--- Define Clubs' Atmosphere ---*/

    hasAtm(oneninetyeast,snobby).
    hasAtm(kingkamehameha,snobby).
    hasAtm(deelight,neutral).
    hasAtm(index,shabby).
    hasAtm(u60311,shabby).
    hasAtm(o25,neutral).
    hasAtm(robertjohnson,neutral).

/*--- Define Songs' Style ---*/

    isStyle(derdritteraum,electro).
    isStyle(anthonyrother,electro).
    isStyle(chrisliebing,techno).
    isStyle(svenvaeth,techno).
    isStyle(rush,techno).
    isStyle(armandvanhelden,house).
    isStyle(daftpunk,house).
    isStyle(stgermain,newjazz).
    isStyle(cafedelmar,chill).
    isStyle(sensorama,chill).
    isStyle(stevebug,techhouse).
    isStyle(villalobos,techhouse).
    isStyle(djkoze,techhouse).
    isStyle(scooter,rave).
    isStyle(taucher,trance).

/*--- Define Price Categories ---*/

    hasLimits(cheap,0,5).
    hasLimits(fair,5,10).
    hasLimits(expensive,10,41).

/*--- Define homepage of each club ---*/

    hasHomepage(u60311,'http://www.u60311.net/').
    hasHomepage(oneninetyeast,'http://www.190east.de/').
    hasHomepage(kingkamehameha,'http://www.king-kamehameha.de/').
    hasHomepage(deelight,'No homepage available').
    hasHomepage(index,'http://www.index-ffm.de/').
    hasHomepage(o25,'http://www.o25.de/').
    hasHomepage(robertjohnson,'http://www.robertjohnson.de/').

```