Visual Pattern Recognition based on Bounded Rationality. Experimental Model

Ralf D. Fabian
ralf.fabian@ulbsibiu.ro

“Lucian Blaga” University of Sibiu, Faculty of Engineering
- 2011 -
Experimental model

- Visual Pattern, why?
  
- Here BR is *not related to a product* – deliver Lena at pixel level precision – but rather to providing information about Lena for everyone who needs them.

- Every service is performed by a live entity,
  - if biological – human
  - if virtual – agent

- apply BR to simplify visual complexity to be able to transmit only what’s needed

- focus on user

- level of granularity

- feature – relevance to the target
Experimental model

Dialog

```java
switch (client) {
    case of: client_1
    case of: client_2
        .
        .
    case of: client_n
    default:
}
```

Features of the interest area (technical variables)
- location of the interest area
- precision for the interest area
- imprecision for the complementary area
switch(CLIENT)

  case: CLIENT_1
    call(CLIENT_1) ➔ Dialog_1 : Dialog
    if (dlg_result == 0) then
      call(PROCESSING_1)
    else
      think_about_bankruptcy()

  case: CLIENT_2
    call(CLIENT_2) ➔ Dialog_2 : Dialog
    if (dlg_result == 0) then
      call(PROCESSING_2)
    else
      think_about_bankruptcy()
    ...

  case: CLIENT_N
    call(CLIENT_N) ➔ Dialog_N : Dialog
    if (dlg_result == 0) then
      call(PROCESSING_N)
    else
      think_about_bankruptcy()

  case: OTHERWISE
    do_defaults()
switch(PROCESSING)

    case: PROCESSING_1
        // Lenas’s mother Requirements/specifications/features

            foreach( requirement )       // from Dialog_1
                do_number_crunching(image, requirement)

    case: PROCESSING_2           // Dumy processing

            foreach( requirement )       // from Dialog_2
                do_number_crunching(image, requirement)

    ...

    case: PROCESSING_N           // Detective

            foreach( requirement )       // from Dialog_N
                do_number_crunching(image, requirement)

    case: OTHERWISE               // oioi

    // TO DO List
    --

Common Global Memory
Onion principle – Successive prototyping

Layers:
1 - general dialog
2 - dialog 1 ... dialog n
3 - adding processing for the dialog, processing 1... processing n
4 - for a dialog k, have the sequence dialog + specific dialog for case k
5 - on processing, a general processing
Onion principle – Successive prototyping

- Its exterior gives no clue as to the complexity of the layers within. It is at the same time simple and complex.

- There is no magic number for layers and slices.

- The important thing to do is to work with the level of detail that is useful: the level that works.

- "big picture" person who would rather limit the details - prefer to work with the short list of "essentials".

- detail minded person who likes to analyze the inner workings of everything - may find the long complex list more attractive.
Conclusions and future work

- Precision is against nature and the opposite of precision is fuzziness.
- Ever more services have to be provided in line with the “just in time” (JIT) paradigm;
- Developing applications for JIT services implies both bounded rationality as fact of life and artificial intelligence as powerful IT instrument.
- BR as response to JIT
- Dialog – evolution from textual to non textual/multimodal
Thank You!

AS YOU CAN CLEARLY SEE IN SLIDE 397...

GAAAAH!

“POWEROINT” POISONING.