

- 
- 
- 
- 
- 
- 
- 
- 
- 

# *Humanist Computing*



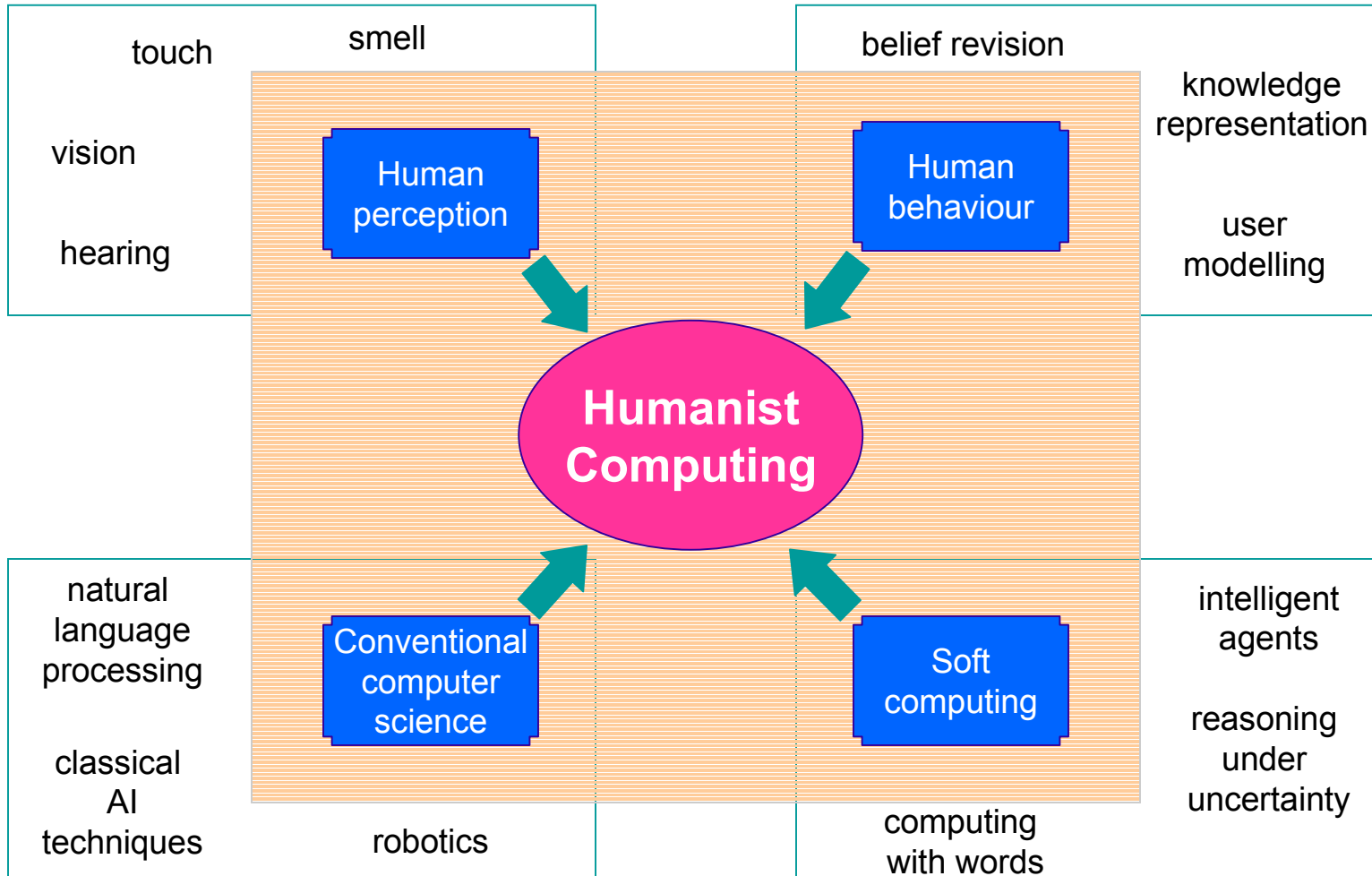
**Dr Jonathan Rossiter**  
Artificial Intelligence Research Group  
Department of Engineering Mathematics  
University of Bristol

[Jonathan.Rossiter@bristol.ac.uk](mailto:Jonathan.Rossiter@bristol.ac.uk)



- 
- 
- 
- 
- 
- 
- 
-

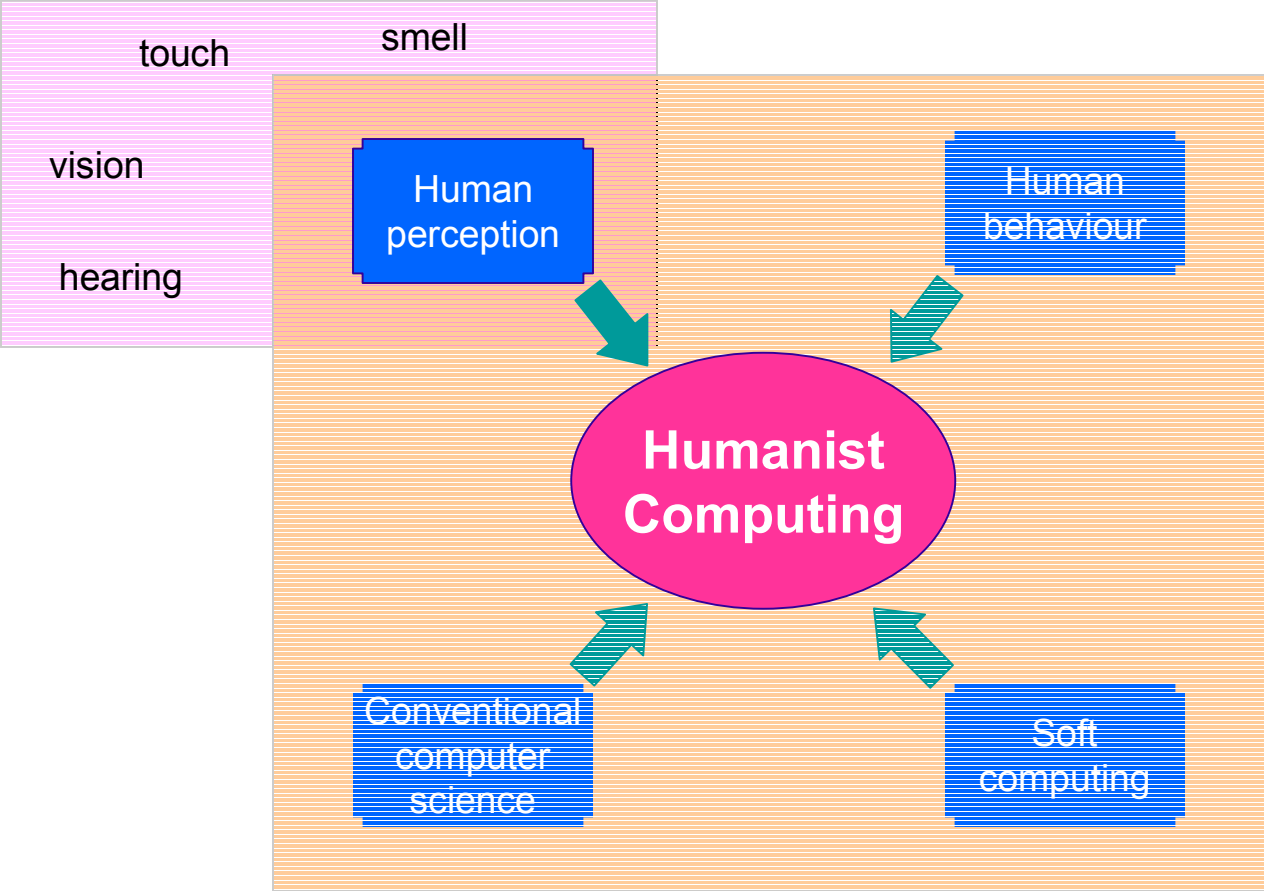
# Humanist computing



Jonathan Rossiter 2000

- 
- 
- 

# 1. Human perception



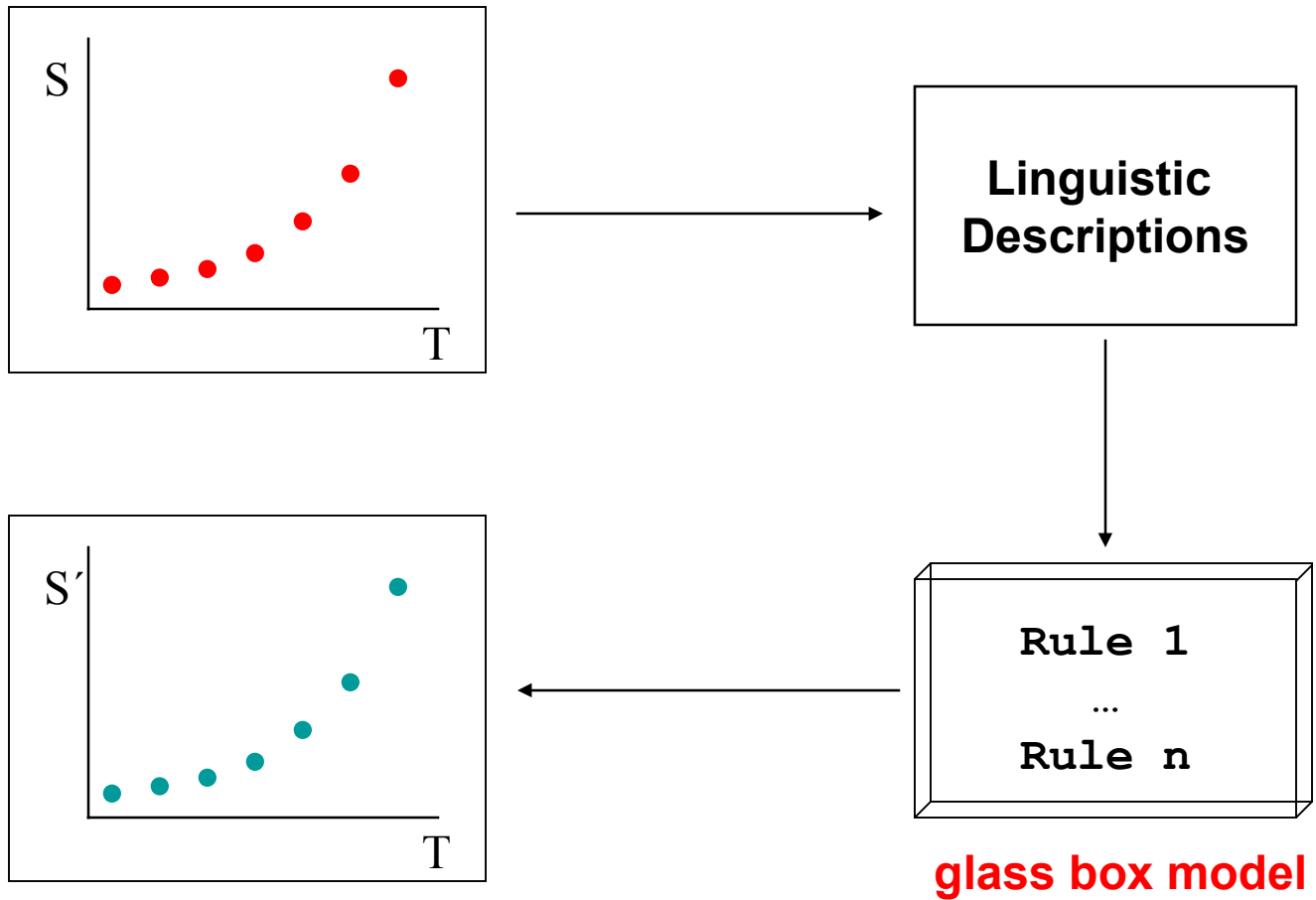
Jonathan Rossiter 2000



- 
- 
- 
- 
- 
- 
- 
- 
-

- 
- 
- 

# Linguistic modelling

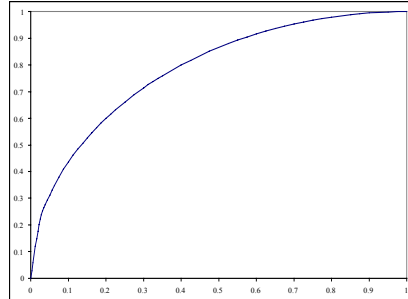


- 
- 
- 
- 
- 
- 
- 
- 
-

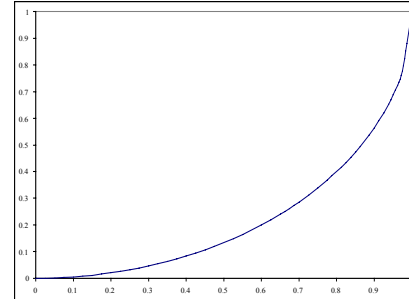
- 
- 
- 

# *Example prototype functions*

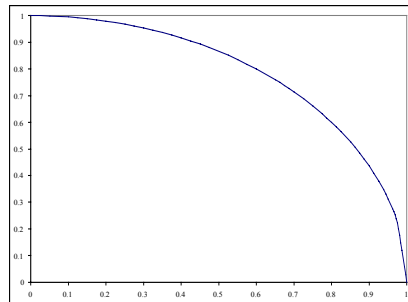
*rising  
less  
steeply*



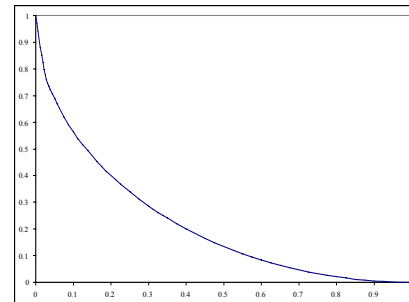
*rising  
more  
steeply*



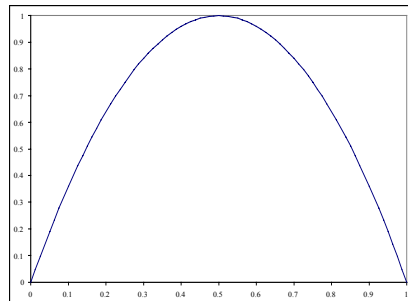
*falling  
more  
steeply*



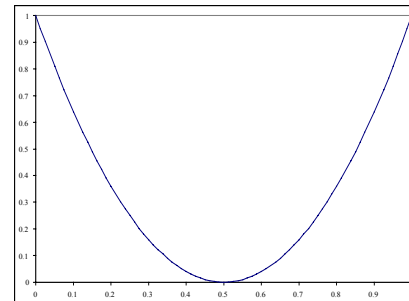
*falling  
less  
steeply*



*crest*



*trough*



- 
- 
- 

## Linguistic fuzzy rules

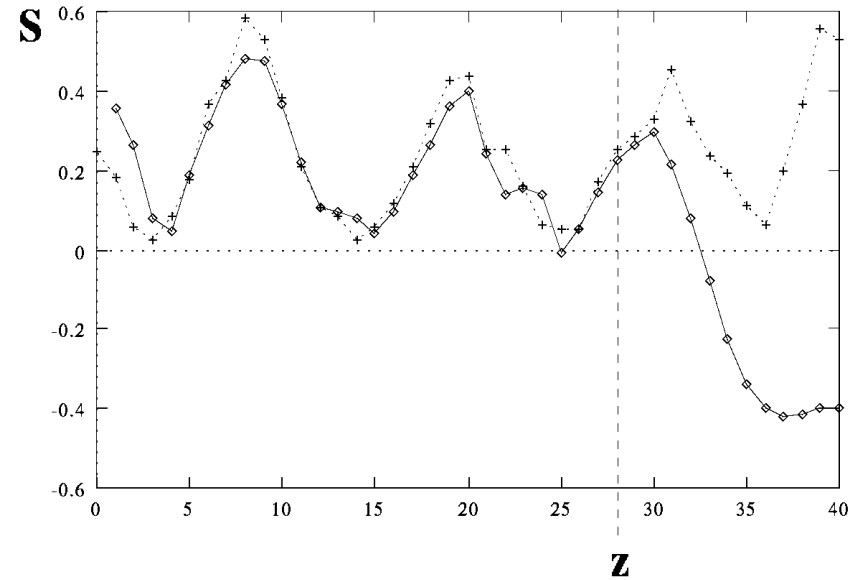
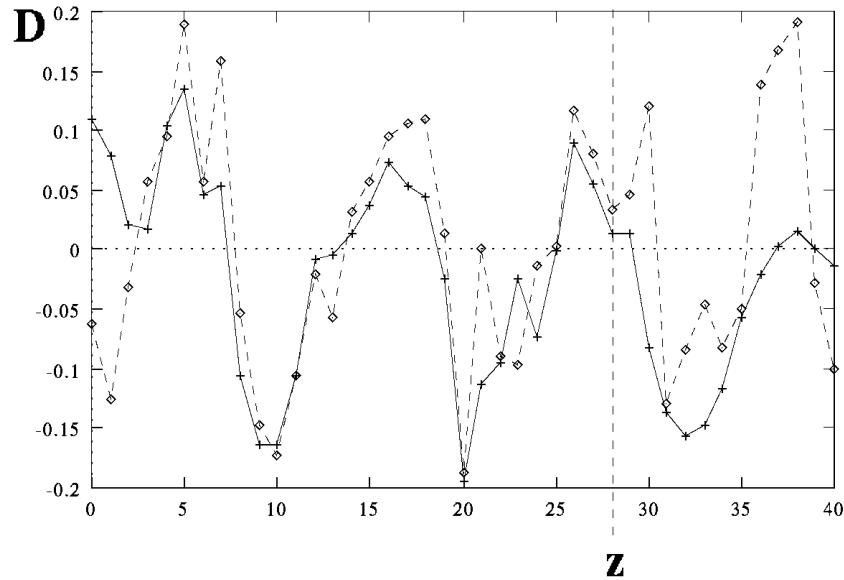
```
((difference is low if)
    (trend was falling less steeply)           0.5
    (trend was falling more steeply)          0.3
    (trend was rising less steeply)           0
    (trend was rising more steeply)          0
    (trend was crest)                         0.2
    (trend was trough)                        0
)
```

- “the next point in the series will be *lower* than the current point if the trend has been *falling less steeply* with weight 0.5 and *falling more steeply* with weight 0.3 and a *crest* has been passed with weight 0.2”

- 
- 
- 

# Sunspot data

dataset	train	test	window	rules
sunspot	$[s_0, \dots, s_{29}]$	$[s_{30}, \dots, s_{59}]$	6	2

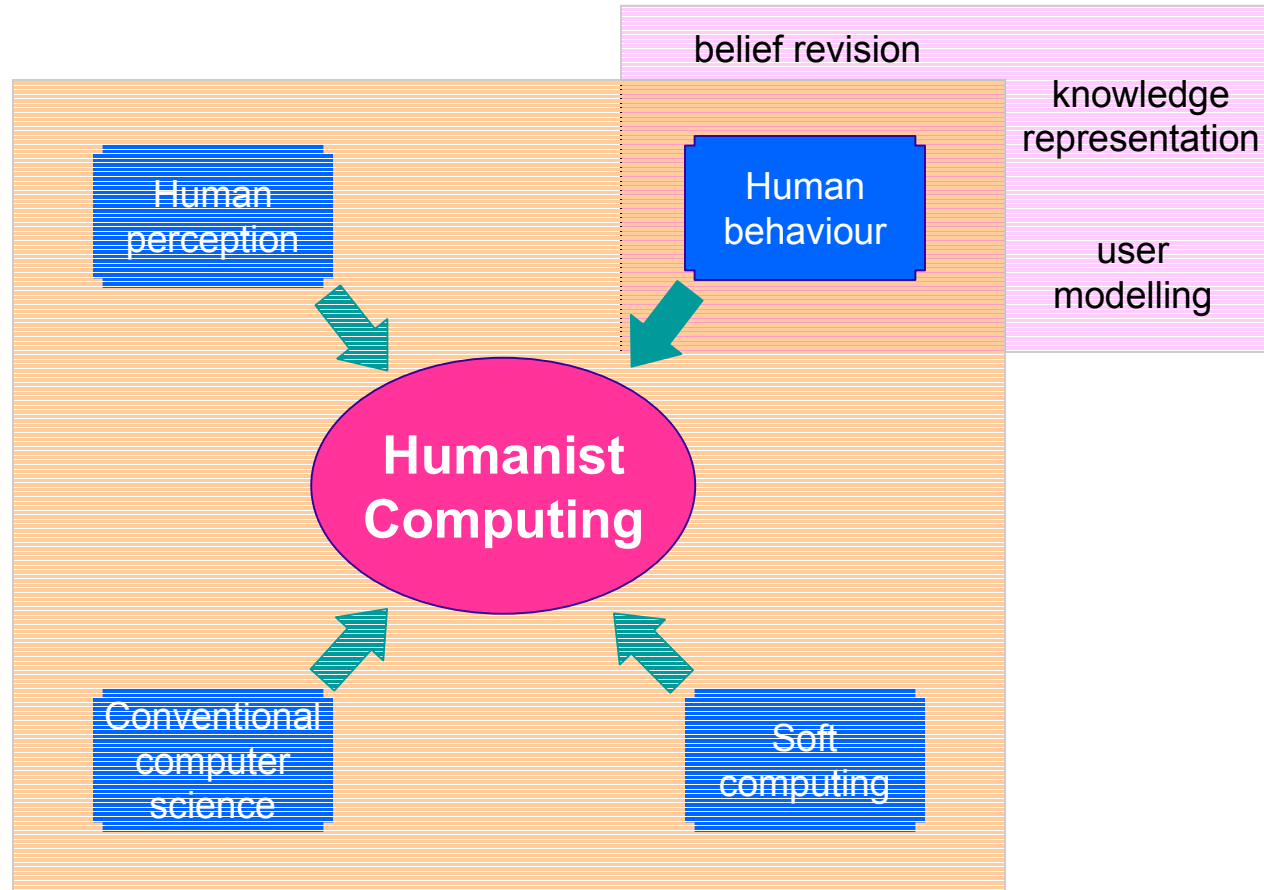


Jonathan Rossiter 2000

- 
- 
- 
- 
- 
- 
- 
-

- 
- 
- 

## 2. Human behaviour



- 
- 
- 

## ***Belief in ordered databases***

- **Example1 - normal student results**

Date	French	Philosophy	Computers	Competence
01-Sep	29	27	32	<b>low</b>
08-Sep	30	27	35	<b>low</b>
15-Sep	40	46	50	<b>medium</b>
22-Sep	52	50	64	<b>medium</b>
29-Sep	63	64	79	<b>high</b>
06-Oct	81	70	78	<b>high</b>

- **as student's grades increase, belief in his/her competence increases**

- 
- 
- 

## ***Belief in ordered databases***

- **Example2 - odd student results**

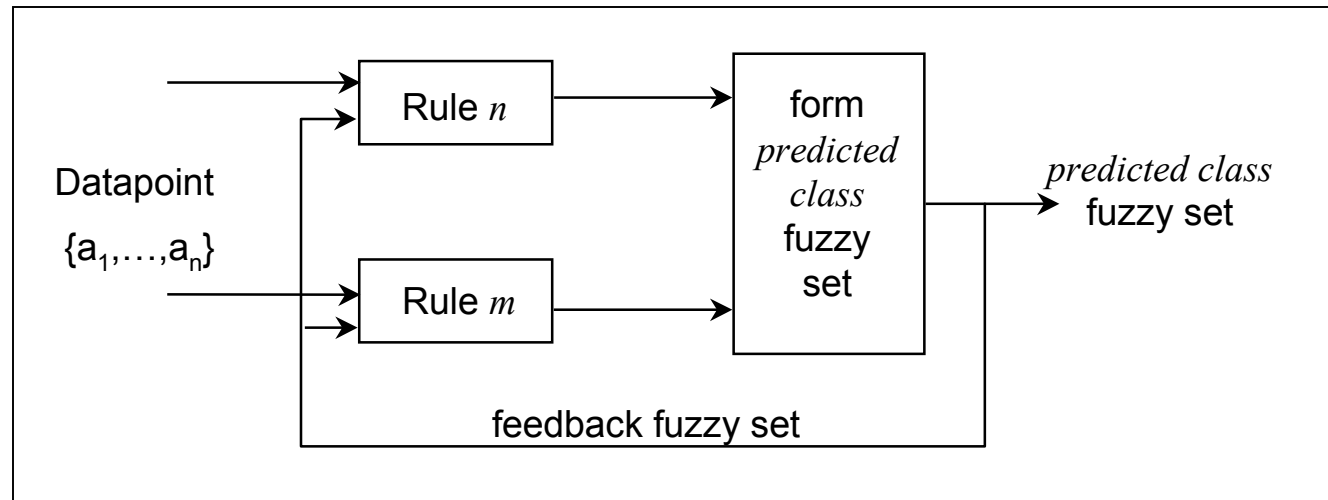
Date	French	Philosophy	Computers	Competence
01-Sep	29	27	32	low
08-Sep	30	27	35	low
15-Sep	37	34	34	low
22-Sep	69	75	78	?
29-Sep	78	73	82	?
06-Oct	81	80	74	?

- **What happens in weeks 4, 5, and 6?**
  - Is competence *high, medium or low*?

- 
- 
- 

## Recurrent belief updating rules

- **Interactive feedback rules**



- ***predicted class fuzzy set is feedback term***
  - E.g.  $\text{predicted class} = 1/1 + 2/0.2 + 3/0.5$

- 
- 
- 

# Facial feature detection

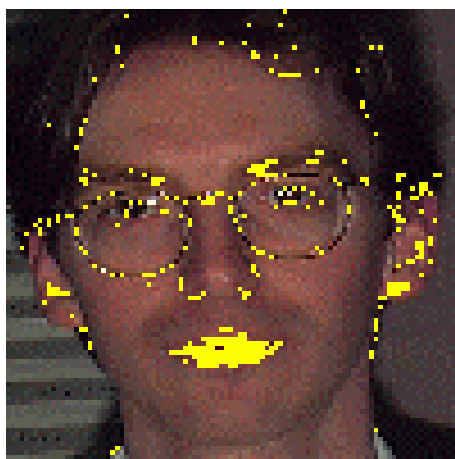
Original



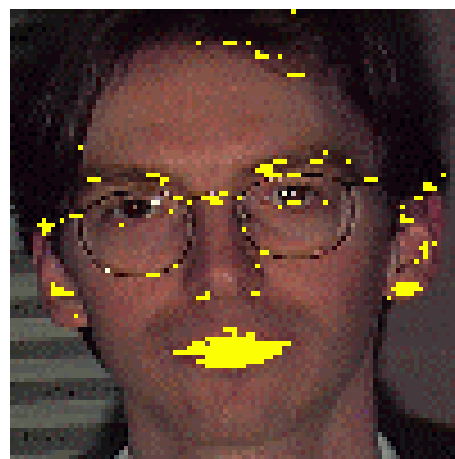
Lip mask



Non-recurrent rules



Recurrent rules



- 
- 
- 
- 
- 
- 
- 
-

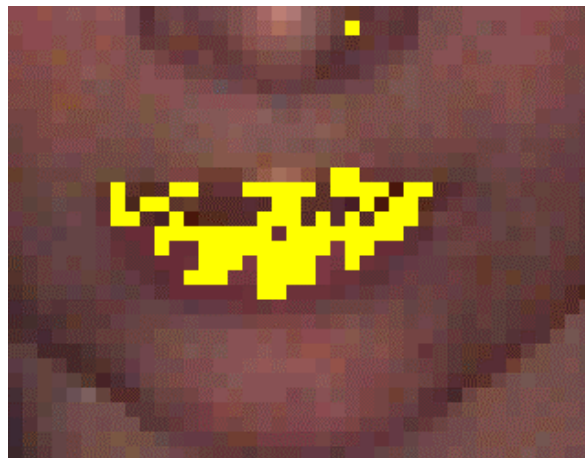
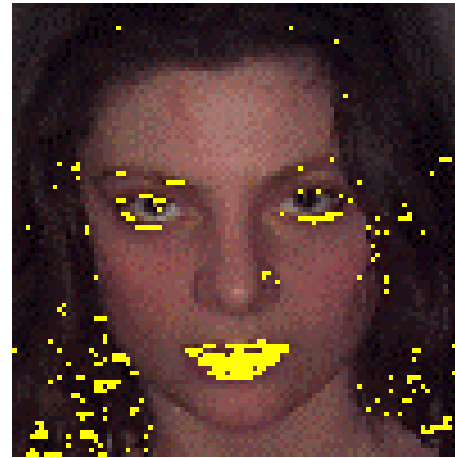
- 
- 
- 

# Facial feature detection

Non-recurrent  
rules

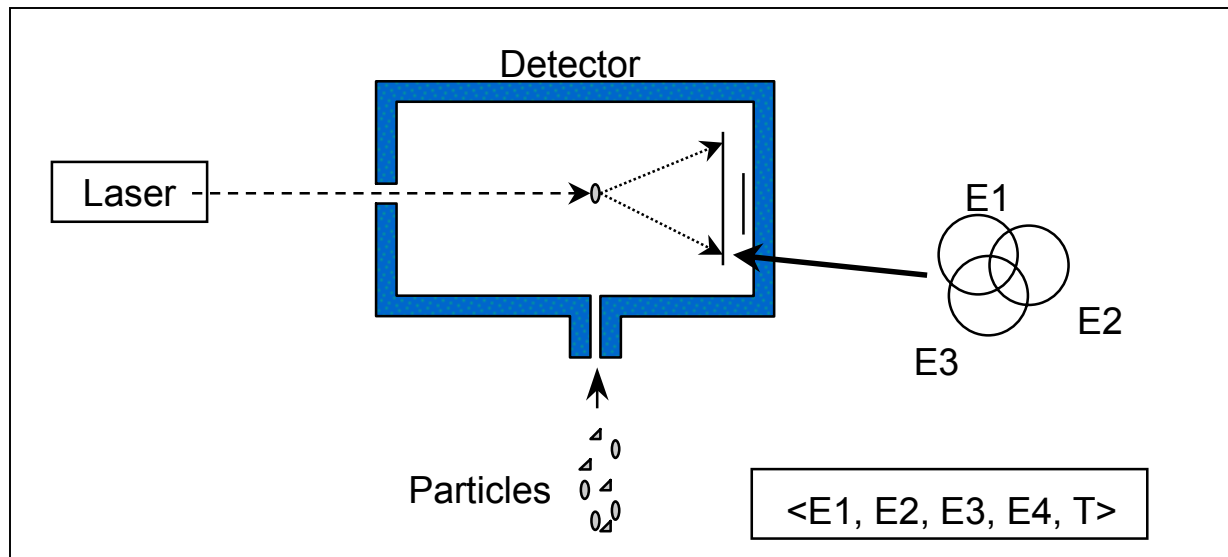


Recurrent  
rules



- 
- 
- 
- 
- 
- 
- 
-

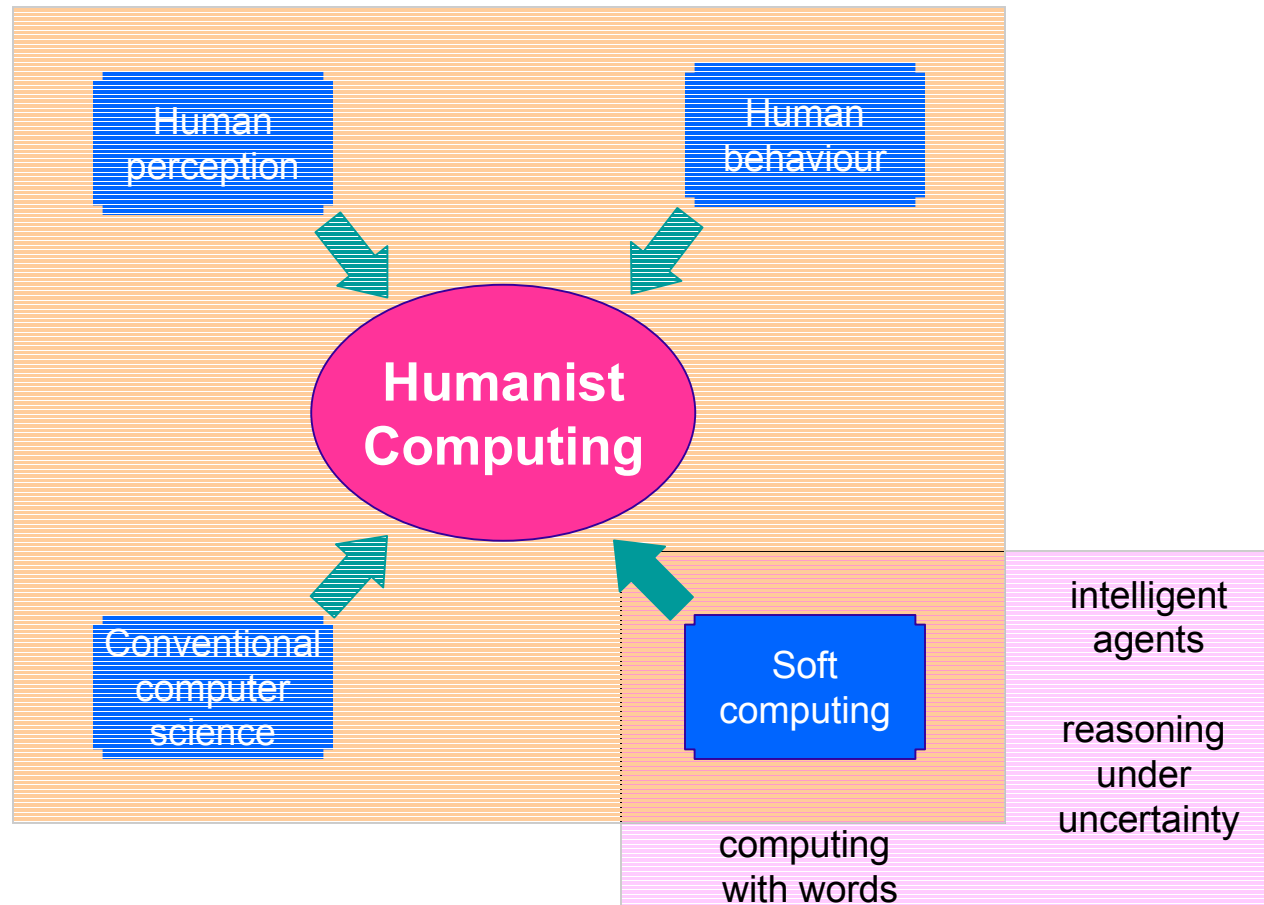
# Gas detection and classification



Dataset	% classified correct	
	Single point	Recurrent
1	56	<b>67</b>
2	59.8	<b>70</b>

- 
- 
- 

## 3. *Soft computing*

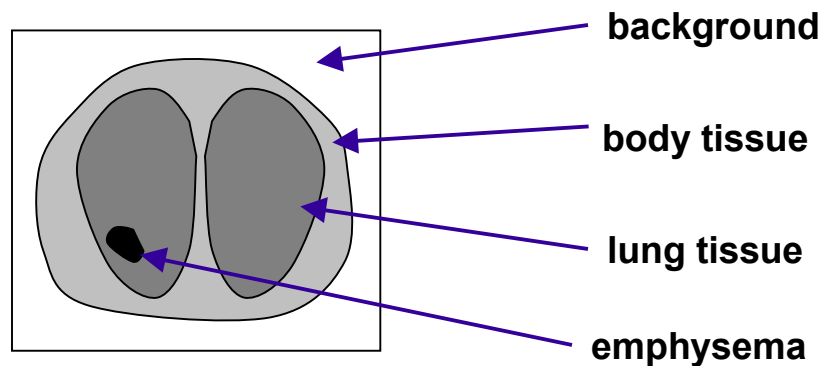


Jonathan Rossiter 2000

- 
- 
- 
- 
- 
- 
- 
-

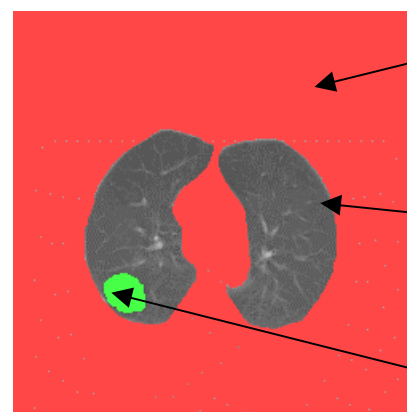
- 
- 
- 

# Detection of emphysema in lung scans



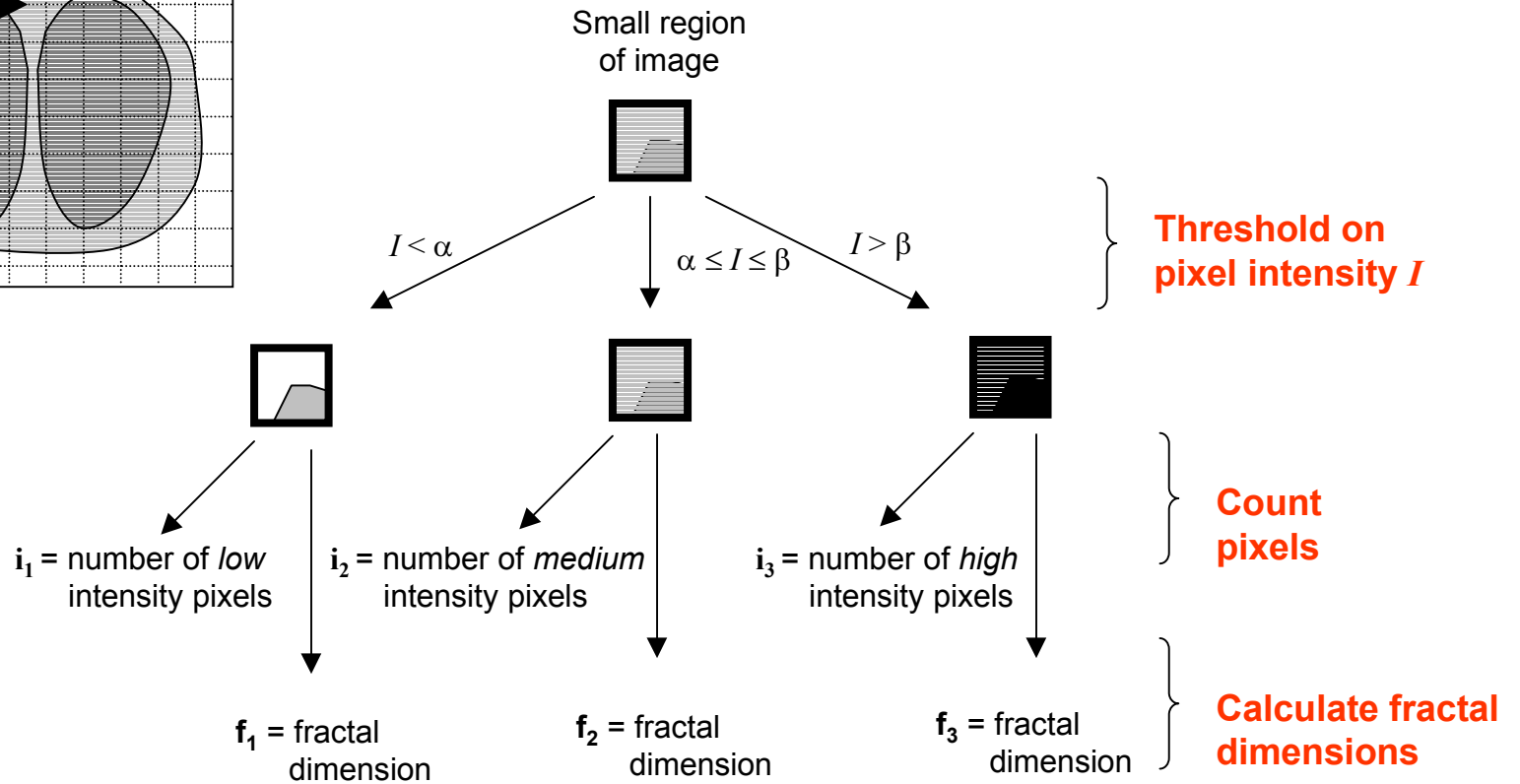
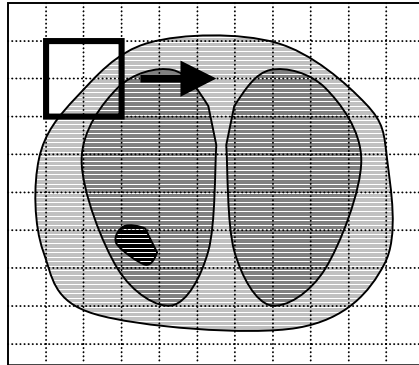
original image

classify →



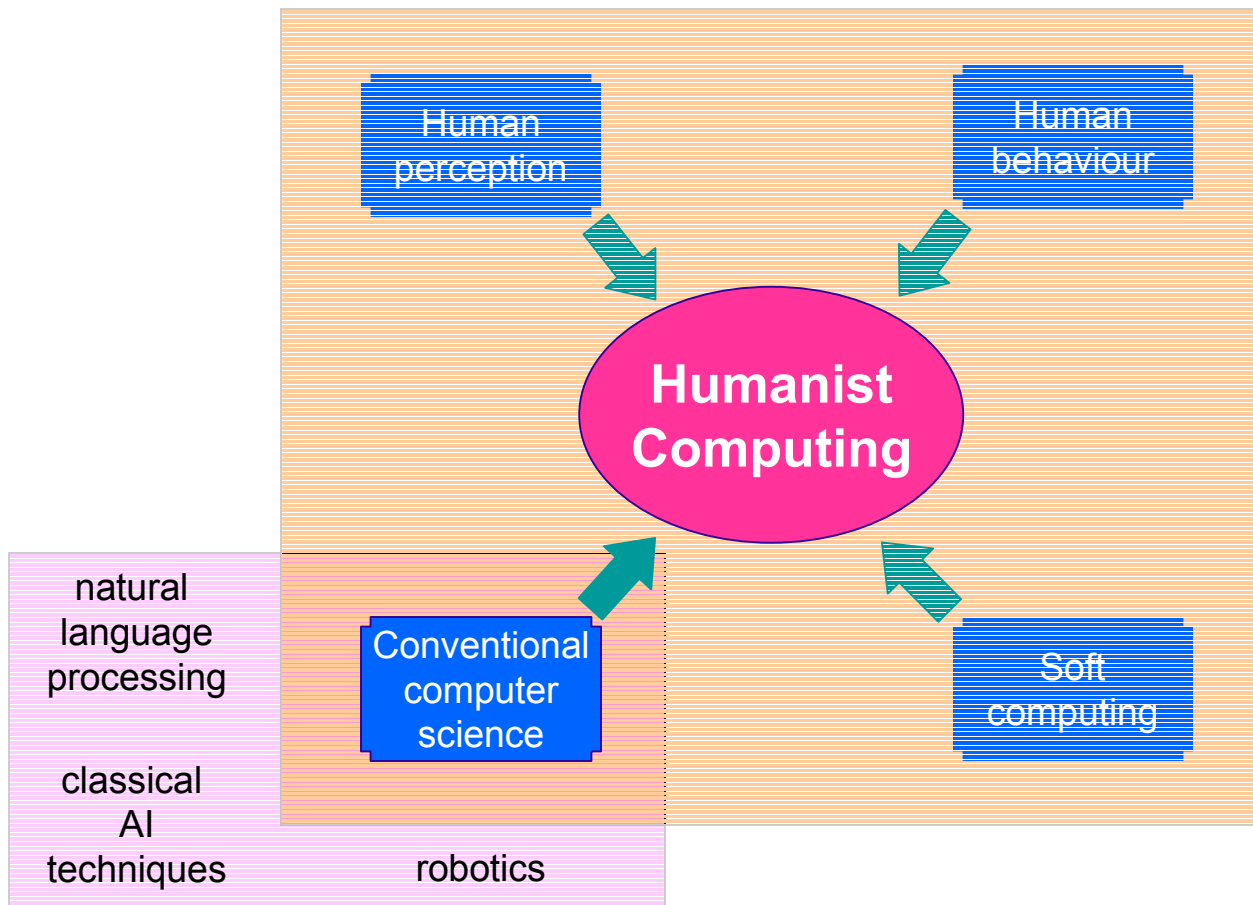
classified image

# Detection of emphysema in lung scans



- 
- 
- 

## 4. Conventional computer science

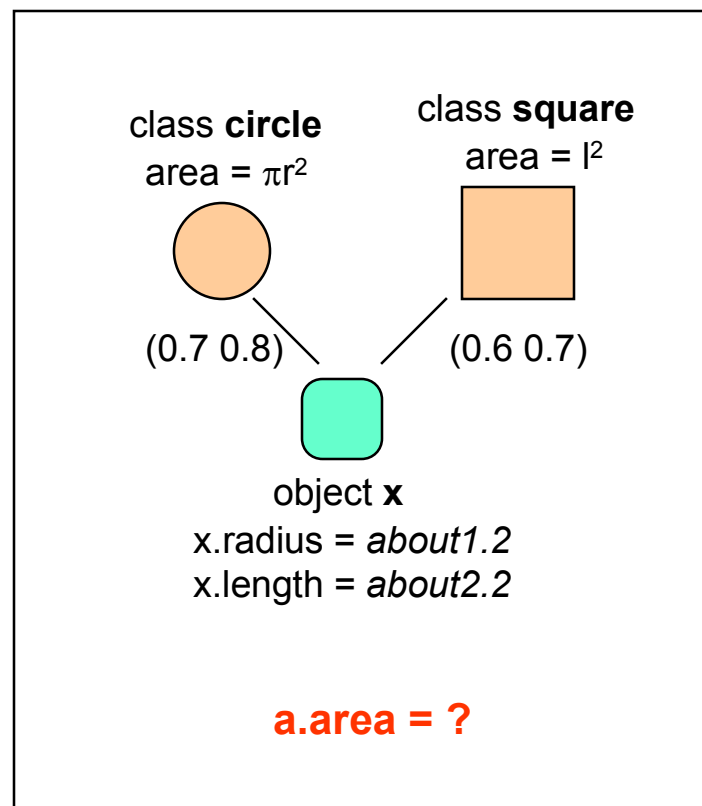


Jonathan Rossiter 2000

- 
- 
- 

# The Fril++ programming language

- Object-oriented
- Logic programming language
- Extension of Fril
- Fuzzy sets
- Mass assignment
- Support logic

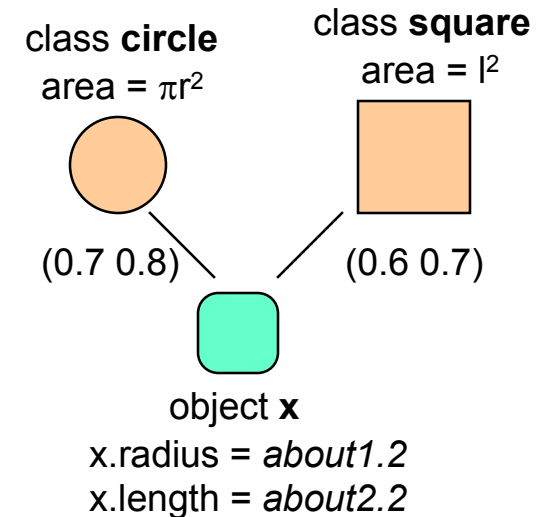


# Example

```
((public class circle extends (fpp))
  (attributes ((radius _)) )
  (methods ((area A)
            (this.radius R)
            (times R R R2)
            (times R2 3.141 A)) ))

((public class square extends (fpp))
  (attributes ((length _)) )
  (methods ((area A)
            (this.length L)
            (times L L A)) ))

((public class mainClass extends (fpp))
  (constants (about1.2 [1:0 1.2:1 1.4:0])
             (about2.2 [1.8:0 2.2:1 2.6:0]))
  (methods ((main)
            (new x ((circle (0.7 0.8))
                    (square (0.6 0.7))))
            (x.set radius about1.2)
            (x.set length about2.2)
            (x.area A)
            (p "x's area is" A)
            (pp)) ))
```



by **circle** method

a.area = [3.14:0 4.5216:1 6.1544:0]

by **square** method

a.area = [3.24:0 4.84:1 6.76:0]

**x's area is 4.65336**

- 
- 
- 

## Summary

- **Humanist computing includes**
  - Human perception
  - Human behaviour
  - Soft computing
  - Conventional computer science and AI
- **Applications include**
  - Facial feature recognition
  - Series modelling
  - Detection of emphysema from lung images
- **Fril++**
  - Tool to implement humanist and soft computing
  - Object-oriented logic programming with fuzzy sets and support logic