MAI4CAREU

Master programmes in Artificial Intelligence 4 Careers in Europe

University of Ruse

INTELLIGENT COMPUTER SYSTEMS

Svetlana Stefanova

September, 2022

Co-financed by the European Union
Connecting Europe Facility

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

LECTURE 4

EXPERT SYSTEMS

1. Introduction
2. Intelligent computer system vs. expert system
3. Expert system specifics
4. Example expert systems
5. Expert system architecture
6. Toolkit

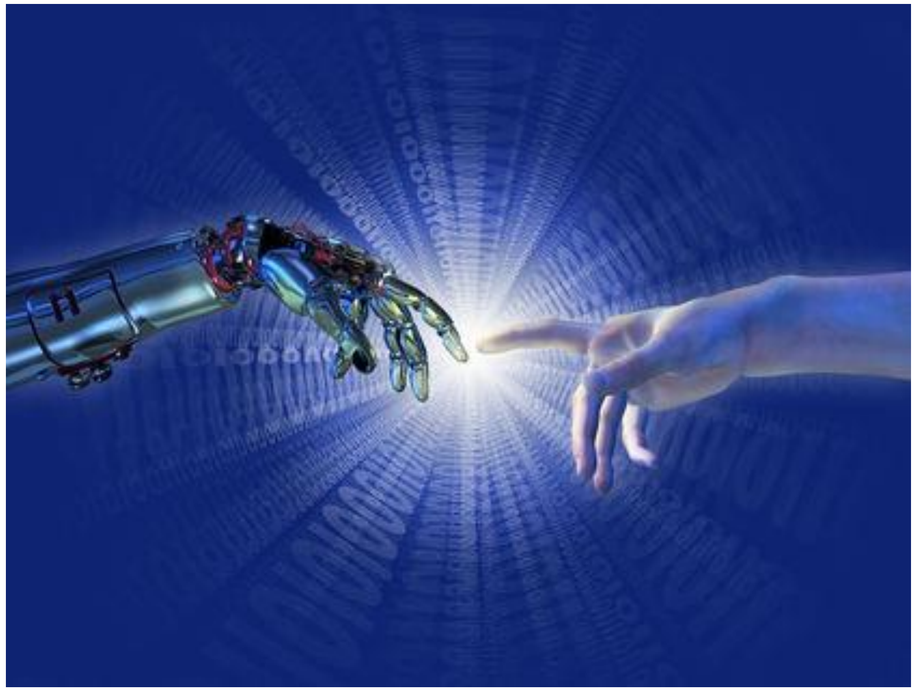
Co-financed by the European Union
Connecting Europe Facility

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

Definition of expert system (ES)

A computer program, processing the accumulated in it knowledge of experts-specialists in a given subject area and providing a solution at an expert level to a certain range of tasks.

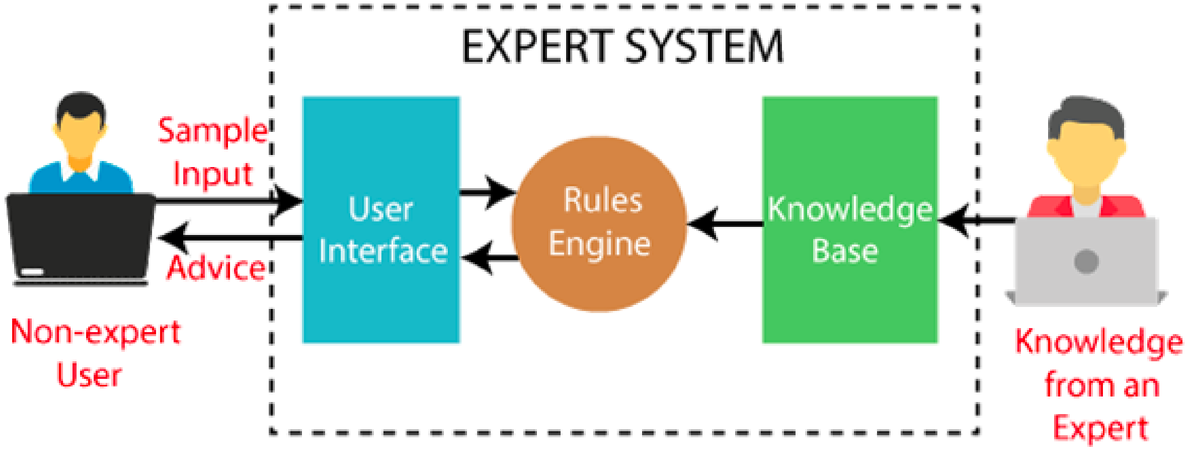
Appearance – the 70s, together with the first practical results.



This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

CONTENT 1

Definition of expert system (ES)



This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

CONTENT 2

ES towards the broad class of ICS tasks

- performs difficult tasks at the level of a good specialist;
- uses problem-oriented strategies for solving tasks compared to more general ICS methods;
- uses knowledge about itself (metaknowledge) to conclude how the inference process proceeds and can give explanations for the resulting decisions.

Co-financed by the European Union
Connecting Europe Facility

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

CONTENT 2

AI vs Expert System - Comparison Chart

Artificial Intelligence	Expert System
AI is the simulation of human intelligence in machines that are programmed to imitate human capabilities.	Expert systems refer to computer programs that simulate the thought process of a human expert to solve complex problems in a specific domain.
AI is the study of systems that act in a way to any observer would appear to be intelligent.	Expert systems represent the most successful demonstration of the capabilities of AI.
Components of AI include Natural Language Processing (NLP), knowledge representation, reasoning, problem solving & machine learning.	Expert systems are typically composed of the inference engine, the knowledge base, the user interface and the knowledge acquisition module.
AI systems are used in a wide range of industries, from healthcare to finance, automotive, data security, social media, travel and transport, etc.	Expert systems provide expert advice and guidance in a wide variety of activities, from computer diagnosis to delicate medical surgery.

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

CONTENT 3

ES features

- provide a competent solution to the given problem;
- focused on real-world tasks;
- narrowly oriented in a specific area;
- handle not data, but knowledge;
- flexible to acquire new knowledge;
- understandable through explanations.

This Master is run under the context of Action
No 2020-EU-IA-0087, co-financed by the EU CEF Telecom
under GA nr. INEA/CEF/ICT/A2020/2267423

CONTENT 3

Nature of the solved tasks

- **With narrow scope** - the task must be narrow enough to encode appropriate expertise, but also complex enough to require expertise.
- **Existence of an expert** – an expert system cannot be constructed in a field where there are no experts yet.
- **Agreement between experts** - if there is frequent and significant disagreement between the experts in the problem area, the task is not suitable for the ES.
- **Availability of known data** – since ES is built gradually and knowledge is added in response to observed difficulties, it is necessary to have enough test results to explore the limits of what the system knows.

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

CONTENT 3

Classes, solved tasks

- interpretation;
- prognosis;
- diagnostics;
- setting;
- planning;
- repair;
- management;
- learning.

A number of tasks (logical inference, pattern matching, recognition of sounds and other images, etc.) are not in an obvious state in this list.

Co-financed by the European Union
Connecting Europe Facility

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

CONTENT 3

Why are ES popular?

- solve a wide class of hard-to-formalize problems considered unsolvable by computer;
- in practical tasks, results are achieved that are obtained only by highly qualified specialists-experts;
- allow non-specialists in programming to develop and solve their necessary tasks.

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

CONTENT 3

Reasons for building an ES (excluding scientific)

- **Multiplication of expertise** – with the aim of using it for consulting. Geographical distances and retirement are important reasons for the unavailability of an expert.
- **Unity of expertise** - gathering in one place what different experts know about specialized cases.
- **Documentation** – important in learning, for example.

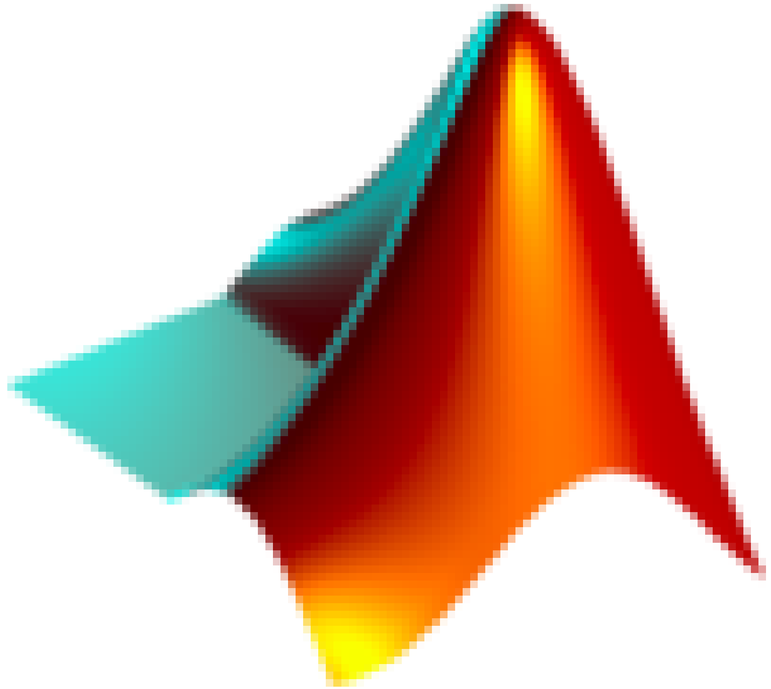
Co-financed by the European Union
Connecting Europe Facility

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

CONTENT 4

Examples of ES

- **MATLAB** - software environment for numerical analysis and independent programming language;
- **MYCIN** - analyzes medical data for a patient with an acute infectious disease, diagnoses the probable causes of the infection and chooses the best treatment;
- **DENDRAL** - in organic chemistry;
- **MACSYMA** - in symbolic integration;
- **PROSPECTOR** - an *expert system* designed for decision-making problems in mineral exploration;
- **PUFF -** analyzes the health status of a person with breathing problems.



Co-financed by the European Union
Connecting Europe Facility

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

CONTENT 4

MYCIN

- Developed in 1976 - 1978 in the Stanford University.
- Based on rules - uses a KB of about 450 rules and 1000 medical facts, primarily for meningitis.

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

CONTENT 4

Extract of a MYCIN consultation session

Answers of the user – with big letters after **, text from the system with*, notes — with *cursive*.

1) Name of the patient: ** PT538

2) Age: **34 YEARS

3) Sex: ** MELE

MYCIN *fixes the answer of the user* = MALE

4) Race: ** WHITE

…….

14) Has PT538 lately had any symptoms of constant headache or any pathological neurological symptoms (dizziness, drowsiness, etc.)? **YES

15) Has PT538 lately had any clearly proven pathological neurological displays (neck rigidity, coma, fainting, etc.), documented through medical supervision or exam? ** YES CSF

* cultures will be considered as related to MENINGITIS.

Note that MYCIN has come to a conclusion and informs the user that the infection is probably meningitis.

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

CONTENT 4

Extract of a MYCIN consultation session

16) Please, give the date, when a clinical proof has been first displayed (symptoms, displays or lab tests) for meningitis? **29. 01. 1977 22:15

…….

23) How many days has PT538 had pathological neurological displays? ** 7,2 HOURS

Note that MYCIN accepts different units for time measurement.

……...

28) Do you know the results of the last CBC* of PT538? **YES

29) Please, give the CBC results in the given table:

CBC	Percentage
In thousands	**PMN****
29 ** K9	UNKNOWN

Note that, MYCIN accepts UNKNOWN as a valid value,

…….

30) Has PT538 had any injuries, disabilities or defects in the central nervous system (except neuro-surgical)? ** NO

Co-financed by the European Union
Connecting Europe Facility

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

CONTENT 4

Extract of a MYCIN consultation session

35) Is meningitis an internal hospital infection? ** WHY

* i.e, WHY is it important to determine if the disease was acquired while the patient was hospitalized

The user asks WHY, to see an explanation of MYCIN's reasoning. The program first paraphrases the question and then displays the rule whose premise it is trying to establish.

*[2.0] This will assist in determining the organisms (other than those seen in cultures and swabs) that could be causing the infection. It has already been established that:

[2.1] the infection requiring treatment is meningitis, and

[2,2] no microorganisms were found during examination of the cerebrospinal fluid, and

[2.3] the type of infection is bacterial, and

[2.4] neurosurgery is not an appropriate solution based on the description of PT538, and [2.5] the duration for PT538 is greater than 10 days.

Therefore, if

[2,6] the infection was acquired while the patient was hospitalized, then, there is evidence that the organisms (other than those seen in cultures and swabs) that could cause the infections are: (E. coli (0,75), Staphylococcus-coag-pos (0,3), Pseudcmonas-aeruginosa (0,3) и Klebsiella-pneumoniae (0,5).

.....

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

CONTENT 4

MYCIN - conclusions

Level of expertise - of the teaching staff in a faculty of infectious diseases.

Application - not used for daily work in hospitals.

Main obstacle – human processing.

Ways to overcome the problem (for human processing):

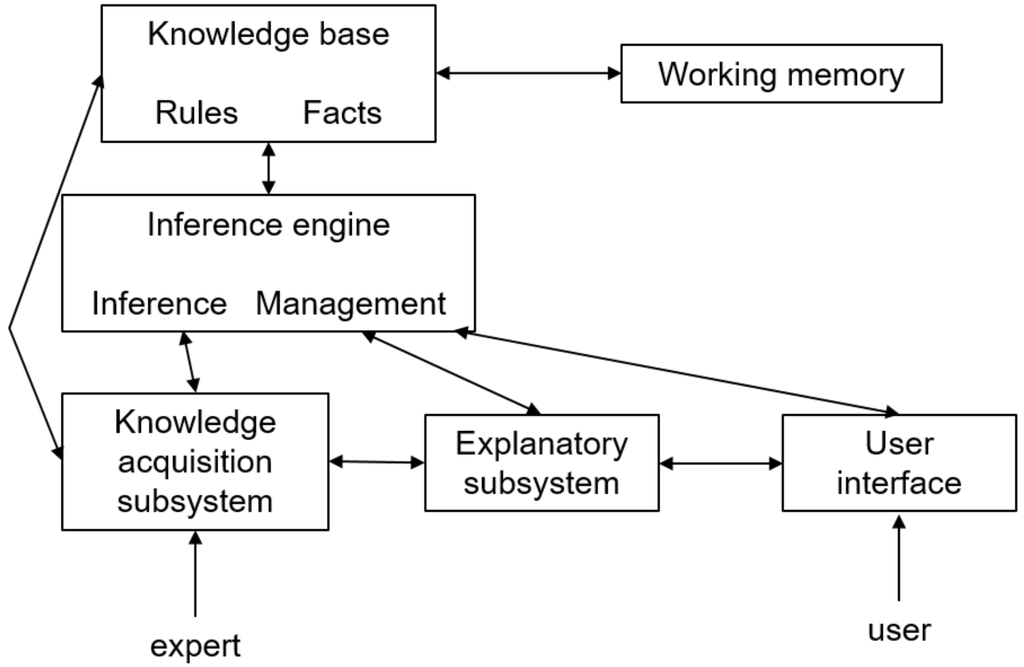
- inclusion of high-quality graphics with user-oriented forms and input/output schemes;
- collecting data from connected hardware instead of users.

Co-financed by the European Union
Connecting Europe Facility

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

CONTENT 5

Architecture of ES



This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

CONTENT 5

Elements of ES architecture

- **Knowledge base** – set of specialist and core knowledge including:
  - facts;
  - rules.
- **Knowledge acquisition subsystem** – uses a knowledge extraction methodology:
  - from the environment;
  - through rules.
- **Inference engine** - set of functions in a particular language that control interactive work and update the current state of knowledge about the given case;
- **Explanatory subsystem** – uses methodology to explain the conclusions drawn;
- **Interface section** - turns user input into an internal representation. A working subset of English is often used for communication.

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

CONTENT 5

Knowledge processing in ES

Knowledge engineering - in the process of constructing the ES, the main difficulty is in extracting the necessary knowledge from the experts and processing it into a form suitable for machine processing.

An important achievement in the late 1970s and early 1980s was the construction of **knowledge processing structures** to help build, debug, interpret and explain ES: EMYCIN, ROSIE, KAS, EXPERT and OPS.

Co-financed by the European Union
Connecting Europe Facility

This Master is run under the context of Action
No 2020-EU-IA-0087, co-financed by the EU CEF Telecom
under GA nr. INEA/CEF/ICT/A2020/2267423

CONTENT 5

Knowledge processing in ES - conclusions

- To be able to carry out an analysis, ES need:
  - ➢ basic knowledge;
  - ➢ knowledge about the relevant field.

Main idea - to separate the knowledge of the target area from the procedures that process it. This leads to flexibility and clarity, and the KB can be explored and manipulated like any other data structure.

- Computationally feasible and mathematically accurate methods are known for small classes of problems - a large part of the necessary knowledge is heuristics, which specialists use when solving difficult problems.
- ES can also extract information from reasoning with informal knowledge without trying to simulate the behavior of specialists.

Co-financed by the European Union
Connecting Europe Facility

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

CONTENT 6

Toolkit

Specialized programming languages are required (not used for formal description like standard ones):

- Prolog;
- LISP;
- ML, etc.

The toolkit for building an ES can include:

- procedural languages for programming – PASCAL, PL, LISP;
- declarative languages for programming – OPS5, PROLOG;
- object-oriented languages – Python, SMALLTALK, FLAVORS;
- empty shells – SAVOIR, EXTRAN-7;
- environments – ART, Knowledge Craft, KEE.

Co-financed by the European Union
Connecting Europe Facility

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

CONTENT 6

LISP

LISP (*LISt Processing language*) - the second after Fortran high-level programming language, still in use.

Created in 1958 in MIT by John McCarthy, including programming languages and data.

Dialects have originated from it, the most famous of which are Scheme and Common Lisp.

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

CONTENT 6

Prolog

Prolog (***PRO****gramming in* ***LOG****ic)* - a high-level logic programming language created by Alain Colmerauer, Philippe Roussel and Robert Kowalski in 1972 as an alternative to the American-dominated LISP languages. It is based on first-order predicate calculus with certain limitations.

The goal - a language that uses logical expressions instead of instructions to the computer. Emphasis is on declarativeness in expressing the logical relationships between objects relating to a certain problem rather than on the procedural steps required to solve it. The system decides how to solve the problem, including the instructions for the computer to execute. Prolog solves problems by searching the KB, which can be refined if several processors search different parts of the database.

Co-financed by the European Union
Connecting Europe Facility

This Master is run under the context of Action
No 2020-EU-IA-0087, co-financed by the EU CEF Telecom
under GA nr. INEA/CEF/ICT/A2020/2267423

CONTENT 6

Python

The name comes from BBC's TV show „Monty Python's Flying Circus".

- **Interpreter** - saves development time as no compiling and linking is required to test an application. Like Java, an application written on it is relatively easily portable to other platforms or OS.
- Interactive, object-oriented programming language.
- Has built-in complex data types such as flexible arrays and dictionaries that are difficult to implement in C.
- Allows a program to be split into modules that can be reused in other programs.

- Has:
  - ➢ standard modules to be used as the basis of programs;
  - ➢ built-in modules that provide file input/output (I/O), system functions, sockets, programming interfaces to GUI libraries, etc.

Co-financed by the European Union
Connecting Europe Facility

This Master is run under the context of Action
No 2020-EU-IA-0087, co-financed by the EU CEF Telecom
under GA nr. INEA/CEF/ICT/A2020/2267423

CONTENT 6

Python development environments

- **Integrated Development** Environments (IDE):
  - PyCharm (https://www.jetbrains.com/pycharm);
  - **Visual Studio Code** (https://code.visualstudio.com);
  - **Atom** (https://atom.io);
  - **Eclipse for Python** (https://www.eclipse.org).
- **Online environments** for programming - used for testing short examples or when it is not possible to install a development environment and Python interpreter locally or when sharing a code:
  - **Repl.it;**
  - **PythonAnywhere.**

Co-financed by the European Union
Connecting Europe Facility

This Master is run under the context of Action
No 2020-EU-IA-0087, co-financed by the EU CEF Telecom
under GA nr. INEA/CEF/ICT/A2020/2267423

MAI4CAREU

Master programmes in Artificial Intelligence 4 Careers in Europe

Co-financed by the European Union
Connecting Europe Facility

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423