**Cognitive Architecture *DUAL***

**(the *RoleMap* model included)**

**GETTING STARTED WITH DUAL:**

**General information:**

This program is a portable implementation of the *DUAL* cognitive architecture. It is a shell for building *DUAL*-based models and testing their behavior through simulation experiments. The cognitive architecture *DUAL* was initially proposed by Boicho Kokinov. Some of the main publications describing *DUAL* are the following:

Kokinov, B. (1994a) "A hybrid model of reasoning by analogy". In K.Holyoak & J.Barnden (Eds.), *Advances in connectionist and neural computation theory*. Vol.2: Analogical connections (pp.247-318). Norwood, NJ: Ablex.

Kokinov, B. (1994b) "The DUAL cognitive architecture: A hybrid multi-agent approach". *Proceedings of the eleventh ECAI* (pp.203-207).

Kokinov, B. (1994c) "The context-sensitive cognitive architecture DUAL". *Proceedings of the sixth annual conference of the cognitive science soc*.

A fuller description of DUAL and AMBR can be found in:

Petrov, A. (2013). Associative Memory-Based Reasoning: A Computational Model of Analogy-Making in a Decentralized Multi-Agent Cognitive Architecture. Saarbrücken, Germany: *Lambert Academic Publishing*.

The implementation here follows the conceptual description of the architecture – its context dependence and hybrid nature. Yet, it differs from the earlier implementations done in *LISP*.

Any additional information may be obtained from:

Yolina Petrova: [yolina.petrovaa@gmail.com](mailto:yolina.petrovaa@gmail.com)

Georgi Petkov: [gpetkov@cogs.nbu.bg](mailto:gpetkov@cogs.nbu.bg)

*DUAL* is a complex program and, as such, it is organized into a specifically ordered system of files. Each file constitutes a relatively self-contained module of the program. It provides some programming objects (i.e., functions, classes, etc.) which are used by other modules. Each file is divided in two sections:

*External protocol* – it documents all classes and/or functions that the file contains.

*Implementation* – it contains the actual Python code.

The files composing the *DUAL* cognitive architecture are in the following order:

1.parameters.py

2.variables.py

3.classes.py

4.links.py

5.agents\_manipulation.py

6.requests.py

7.structures.py

8.working\_memory.py

9.activation.py

10.memory\_consolidation.py – it proceeds main\_cycle.py mainly because of the mapping\_into\_concept() function.

11.main\_cycle.py

12.run\_simulation.py

13.dual\_parser.py

14.LTM\_build\_up.py

**SIMULATIONS:**

All simulations can be run through the file run\_simulation.py in the same manner, described in that file.

# End of file ReadMe\_Instructions.docx #