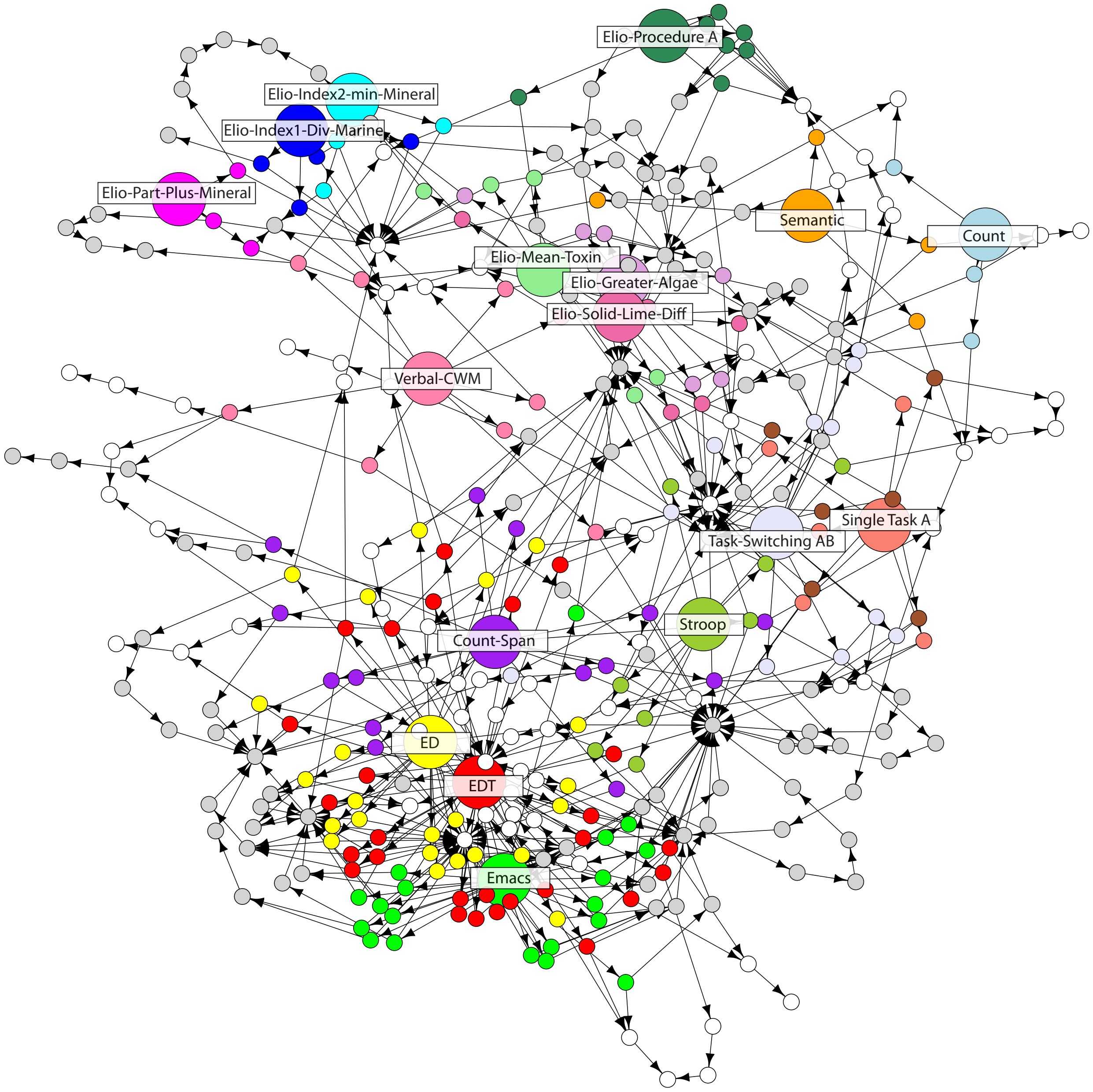
HOW TO RUN THE MODELS



Niels Taatgen
December 2012

Introduction

This document outlines how to load and run Actransfer, and the models outlined in the "The nature and transfer of cognitive skills" paper. Most of the models described in that paper take a substantial amount of time to run. However, in all cases it is possible to run a few trials to get an impression of how the model operates.

This document assumes you retrieved all the files from the webpage http://www.ai.rug.nl/~niels/actransfer In addition to the files specific to Actransfer, it also offers a version of ACT-R with which the system is guaranteed to work, and which is the current version of ACT-R at this moment. Modifications in future versions of ACT-R might break things, so it is recommended you use the version of ACT-R on the webpage.

In addition to the materials on the webpage, you will need a version of Lisp. If you also want to plot the results, and make declarative memory graphs, you will also need a copy of the statistical package R. R can be obtained for free at www.r-project.org. I used LispWorks to run ACT-R and Actransfer, which works fine. ACT-R runs on many different Lisps, and as far as I know there is nothing in Actransfer that is specific to Lispworks, so it should work fine in other systems. Note that most Actransfer models are quite demanding in their computation, so versions of Lisp that have limited heap space (for example the free version of LispWorks) might not work very well.

Starting ACT-R and Actransfer

The QuickStart.txt file in the ACT-R/doc directory of the ACT-R distribution has instructions on how to load ACT-R into Lisp. The process is in principle simple: start up Lisp, and load in the file "load-act-r-6.lisp". This is just starts "vanilla" ACT-R without the environment. The QuickStart document has further directions on how to start the ACT-R environment.

After loading ACT-R, compile and load the file "actransfer.lisp". That is all that is needed to start the system!

Just like regular ACT-R, Actransfer needs a model file. For each of the models discussed in the article there is a separate directory with the model file (with extension .lisp), an R file to analyze the model results (with extension .R), and sample data files (with extensions .txt or .dat).

Running a model

To run a model, load in the model file. Each of the models will have to function to run the model:

(run-experiment n)

runs the experiment with n simulated subjects, and writes the output to a file. Running a single experimental subject typically takes a few hours (with LispWorks on a Mac with a 2.53 Ghz Intel 2 Core Duo).

In addition, each model has a predefined function run-sample. This function will do a single trial or a single block of a particular task with tracing on. Because most of the models implement several tasks, run-sample has to be supplied with a number to indicate which task. Run run-sample without parameters to see the options. Example from the Karbach & Kray model:

```
CL-USER 144 > (run-sample)

"1 - Stroop  2 - Countspan  3 - Task Switch 4 - Single Task"

CL-USER 145 > (run-sample 1)

"Running Stroop"

***    73.82: INSTR59: Start Stroop task, wait for stimulus

***    74.62: INSTR61: Object seen, focus on all

***    75.05: ACTION GET-PROPERTY BOTH (RTRAIN RED-COLOR TRAIN-WORD)

***    75.31: INSTR62: Retrieve color concept of the ink color

***    76.00: INSTR63: Say the answer
```

Running a sample shows a trace that shows at what times root instructions are retrieved, with a brief description of what the instruction does. In addition, external actions are shown, typically in combination with the next percept the actions produce. Standard ACT-R tracing is turned off by default, because these traces tend to be long. The ACT-R trace is saved so it can be inspected with the buffer trace in the environment.

Most of the models have some more options to run the model, check the comments to see how they can be used.

Analyzing results

All the models (with the exception of CountSemantic) produce output files with the model's performance. These files can be analyzed with the supplied R script. Most of the R scripts simply average the data and produce graphs. They also have the experimental data for comparisons. The figures in the article have all been prepared with the supplied R scripts, and the example data files are the basis for those figures.

Making Declarative Memory Graphs

To make a graph of declarative memory like the one on the cover of this document, evaluate the function (print-net) in Lisp after the model has been loaded. This will produce two files, vertices.txt and edges.txt. These files can be loaded into the R-script DMGraphMaker.R, which produce a pdf file with the graph.