# Concept Visualization for Ontologies of Learning Agents

Yu Suo
Computer Systems Technology Lab
2006 -2007 Alexandria, Virginia
March 24, 2007

## Abstract

The purpose of this project is to research, create, and program a concept visualization tool to easily browse small to large scale semantic networks and ontologies of an artificial intelligence. This relates closely to graph theory and graph layout algorithms but adds another factor by taking into account the many relationships between different nodes and elements of an artificial intelligence when drawing the graph.

### 1 Inroduction

The Disciple Learning Agent is an artificial intelligence that can be adapted for many purposes such as military operations, teaching, and traveling. It is currently being developed at the George Mason University Learning Agents Center. The project will be used mainly by those who are not proficient at Computer Science. Therefore, this project aims to distribute information about the artificial intelligence in an easy and usable way.

### 2 Background

The JDisciple Agent is an artificial intelligence based model that subject experts in different fields can train and use when making decisions under stressful, complex, and constrained situations. The tool was developed and used under the Defense Advanced Research Projects Agency's High Performance Knowledge Base and Rapid Knowledge Formation programs. Currently, the tool's main applications are focused toward military action planning.

JDisciple attempts to aid the decision making process under pressure in divergent cases by offering an unbiased solution from given facts. In many cases, JDisciple can improve the decision process in terms of accuracy and efficiency. This has been shown to be true in the current deployed versions of JDisciple in military contracts. The paper will conclude by discussing the future potential of software such as JDisciple and concept visualization for artificial intelligences.

## 3 Procedure and Methodology

The Disciple Learning Agent is currently written in JAVA and called JDisciple. Eclipse is used to provide a basic organized structure for the entire project. This part of the project will be coded in many different iterations starting with the simplest of cases and gradually moving to more advanced, cases. This will involve an implementation of a basic element picking algorithm such as a Greedy, and a basic graph layout algorithm to place and make space for elements on the screen. Then, the algorithms will improved upon every iteration by modifying the heuristics to make it more effcient, faster, and usable for larger ontologies.

## 4 Results and Discussion

At this point, most of the basic preparation is complete and work on the algorithmic portion of this research project can proceed. Programming the basic classes has really given me insight on how Object Oriented Programming is useful in real world applications and why there needs to be strict guidelines for implementing new classes into a larger project.

The first algorithmic version of my program involved manipulation of small explanation element lists in a Disciple agent and represent the elements in an understandable, graphical way. The current algorithm only shows two types of relationships and is based on purely heuristic searching. The object layout system is limited to calculations by 50x50 pixel blocks. This makes the tool very space dependent and unintelligent in space searching and element placement.

## 5 Expected Results

Future versions of this tool strives to do the following:

- Expand current algorithm to work on all general types of relationships.
- Add additional or support heuristics for better greedy selection.
- Optimize current space finding algorithm or program new method.

However, it must be noted that all of the above need to be done keeping in mind all the requirements including but not limited to: integration with main project, efficiency, resource consumption, and ease of use. Thus, it will be progressively more difficult to make this tool work as planned as the algorithm increases in potential to process large capacity ontologies.

## References

[1] Cesim Erten, Stephen G. Kobourov, Vu Le and Armand Navabi, ”Simultaneous Graph Drawing: Layout Algorithms and Visualization Schemes”, Journal of Graph Algorithms and Applications, Vol. 9, no. 1, pp. 165-182, 2005.

[2] Javed A. Aslam, Ekaterina Pelekhov and Daniela Rus, ”The Star Clustering Algorithm for Static and Dynamic Information Organization”, Journal of Graph Algorithms and Applications, Vol. 8, no. 1, pp. 95-129, 2004.

[3] Lorch, Waltraut U., ”An Introduction to Graph Algorithms”, 1999. http://www.cs.auckland.ac.nz/ ute/220ft/graphalg/graphalg.html (March 28, 2000)

[4] Shamir, Ron, ”Advanced Topics in Graph Algorithms”, 1997. http://www.math.tau.ac.il/ rshamir/atga/atga.html (July 11, 2001)