

ARCEdit - CG Editor

Heather D. Pfeiffer and Roger T. Hartley

Department of Computer Science
New Mexico State University
Las Cruces, NM 88003-8001, USA
email: hdp@cs.nmsu.edu and rth@cs.nmsu.edu

Abstract

ARCEdit (Actor-Relation-Concept Editor) is a special-purpose add-in for Microsoft's Powerpoint presentation program that facilitates the authoring of conceptual graph databases in graphical form. It allows the placing and linking of actor, relation, and concept nodes anywhere on a PowerPoint slide. It also allows nested context graphs, linking context in a graph on one slide to another slide. All graphs can be read and saved as either a Powerpoint presentation or in Conceptual Structures CGIF format.

Keywords: Graph Manipulation Editor, Conceptual Structures and Powerpoint Presentation

1. Introduction

ARCEdit (Actor-Relation-Concept Editor) is a special-purpose add-in for Microsoft's Powerpoint presentation program that facilitates the authoring of conceptual graph [6] databases in graphical form.

Its capabilities are:

- easy placement of labeled concept, relation or actor nodes anywhere on the page
- easy linking of nodes with complete checking for disallowed and duplicate links
- easy changing of node labels
- configurable label fonts and node colors
- representation of labeled nested contexts
- representation of a database as a set of graphs
- saving of the database in Powerpoint form (PPT)
- saving of the graph content in CGIF (Conceptual Graph Interchange Format) form, including graph labels and positional information
- loading of any CGIF database including using default layout characteristics saved in comments
- coreference links between graphs
- comments at the graph and node levels

The idea of using an add-in to an existing tool came about because of the authors experience with adding conceptual structures [6] to another tool [2]. The need for and experience of working with conceptual graphs became apparent when preparing to present previous papers on our conceptual structures environment called CP [3,4]. These presentations used the Microsoft product from the Office package called PowerPoint. A Powerpoint presentation is a sequence of slides each of which can contain any textual or graphical information you desire. A slide can also contain sounds, music, or embedded movies, as well as being animated.

ARCEdit provides a way to place conceptual graphs onto slides through a specialize interface. General visual languages [1] add visual constructs to programs to permit the visualization of program execution [5]. ARCEdit is an editor and allows these visual constructs to be placed on a slide.

This does not preclude the use of other embedded information on the slide, but it should be noted that complex shapes and other complex information could well destroy the ability of ARCEdit to interpret the slide consistently as a valid conceptual graph if graphical information outside of syntactic information for a graph is added to a slide.

2. ARCEdit

2.1. ARCEdit vs Powerpoint

The full power of the Powerpoint editor is available for moving shapes around, changing sizes and colors and other attributes of the CG information. Again, pushing these changes too far will destroy the integrity of the CG information. The correspondences of ARCEdit objects to Powerpoint objects are displayed in the following table:

ARCEdit	Powerpoint
database	presentation (.ppt file and/or .cgf file) consisting of a sequence of slides/graphs
graph	slide with special textbox that contains the graph label
concept	colored rectangular shape with textbox
relation	colored oval shape with textbox
actor	colored diamond shape with textbox
link	link between concepts and relations, and actors and relations;"smart" link between two shapes with directional arrow at the end
nested context	a slide whose defining label is either some concept on another graph or the title of another graph (slide)
comment	colored callout containing arbitrary text

2.2. Implementation

Graphs are actually represented three ways:

1. as Powerpoint slides (.ppt)
2. as CGIF files (.cgf)
3. internally as arrays

The trick is to make all three compatible, i.e. all three must be convertible to the other two forms.

Transforms:

1. .ppt -> internal (ScanAllShapes)
2. internal -> .ppt (DrawSlide)
3. CGIF -> internal (readCG)
4. internal -> CGIF (CGIFdump)

A graph can be any number (including 0) concepts, relations or actors. In this sense even disconnected graphs can be considered part of the same graph, in the same sense that a bunch of unrelated logical expressions are implicitly ANDed together. General graphs can be arbitrarily large and nested to any depth. Nesting of graphs must be removed from a single slide since it is visually impossible. Each nested graph is on a different slide. However, coreference links can be between any set of concept nodes in any graph.

So, ARCEdit will maintain graphs in ppt files as separate slides. Each slide will contain a simple graph (no nesting) with any number of coreference links to other graphs. Each slide, and therefore graph, will have a name, corresponding to the defining label for that graph. Thus two graphs with defining labels g1 and g2, respectively, might look like Figure 1.

Their CGIF is (same file):

```
;%Graph*g1%;(happy [cat*x1] )  
.  
;%Graph*g2%;(sad [mouse*x2] )
```

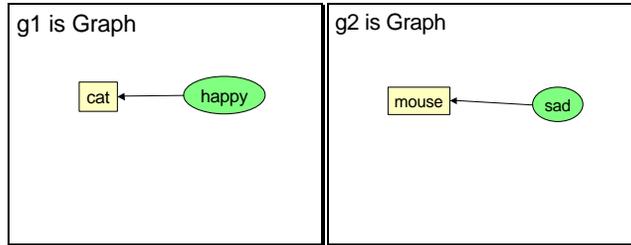


Figure 1: Two slide graphs

Internally, we need to keep track of the links between nodes, and any bindings of nodes to defining labels (or coreferences). There are two kinds of coreferences: 1) to a graph and 2) to another (embedded) concept in the same graph. Actually this is only one, since nested graphs are really just referents of a concept any way so the graph is linked to the concept in the outer graph. A reference to a graph looks like Figure 2. This last object on the slide is the coreference link where the displayed label is the defining label of the node, not its type name (which is absent anyway).

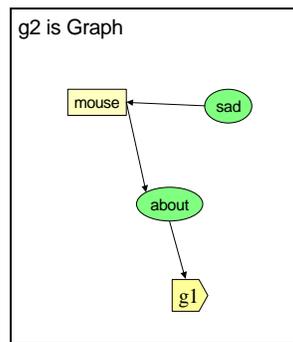


Figure 2: Graph with graph coreference link (rectangle with diagonal side)

Another graph coreference is handled similarly in Figure 3. However here the displayed label is the graph type name because this is a coreference to an actual graph type which is the single concept on the graph.

The transformations are basically done in four functions. Below is a brief description of each routine.

2.2.1. ScanAllShapes

This scans all the shapes on all slides and forms internal structures. Since each shape has all the information for each element, the transformation is straightforward. Each link is also straightforward, since links are also shapes.

2.2.2. DrawSlide

After internal structures have been built, it is scanned to draw shapes onto the slide.

2.2.3. readCG

This parser reads in the graphs from a CGIF (.cgf) file. It makes any needed change to transfer the graphs from CGIF format to internal format.

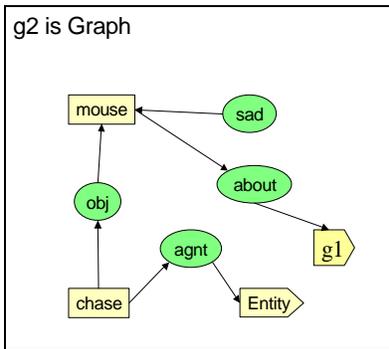


Figure 3: Graph with graph and concept coreference links (rectangle with diagonal side)

2.2.4. CGIFdump

Traversal of each slide drives the writing of CGIF. Since graph names and positioning information is not part the CGIF standard, this information is stored in comments by use by ARCEdit later and to not cause conflicts with any other tool.

2.3. Running Main Window Dialog

Since ARCEdit is an add-in, Powerpoint must have a presentation to edit. The cleanest way to start is to allow ARCEdit to create an empty presentation which can be saved later in a Powerpoint file of your choice. If this file is edited later, ARCEdit will scan the presentation and extract the CG information from it.

To start Powerpoint with an empty presentation, click the cancel button on the opening dialog. ARCEdit is started by clicking the toolbar button labeled **ARCEdit**. If you do not see the ARCEdit button, then your installation has failed, and should be fixed before proceeding (See Appendix A).

The main dialog is shown in Figure 4. This dialog must be visible in order to make use of ARCEdit. Its default position is in Powerpoint's "Normal" view mode on the left side of the slide. It must also be the active window when adding items to the slide. It can be made active by clicking on its title bar.

The menus at the top all perform the commands and actions described by the buttons and choices below except for the "File" and the "Help" menus. The File menu allows the import and export of CGIF Standard files discussed in section 4 and the Help menu brings up content and index menus describing different aspects of how to use the system. A lot of the information in this paper is presented on line through the help system.

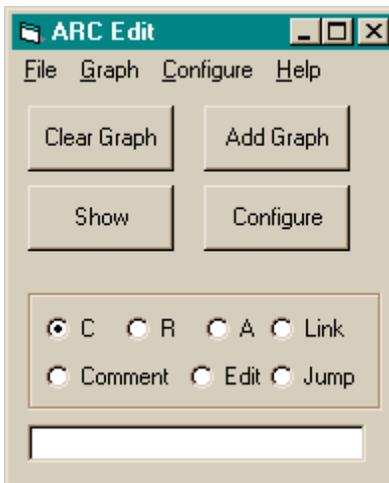


Figure 4: Main Window Dialog

2.4. Main Window Commands

The four buttons below the menus carry out the immediate functions of clearing a graph, adding a new graph to the database, showing a named graph, and running the configuration utility. They are also available from a menu, but are provided here as buttons for ease of use. These buttons produce the following commands:

- **Clear Graph:** Pressing this button removes all nodes and links from the current slide.
- **Add Graph:** Pressing this button with a defining label (beginning with a '*') and/or graph type (text string before the defining label) typed into the text box will add a new graph to the database with that defining label and type. The defining label and type are shown at the top of the slide. If there is no defining label and/or graph type, then a default is generated.
- **Show:** Pressing this button with a graph label typed in the text box will show the named graph in place of the current one. If there is no such labeled graph, or no label is typed, then an error dialog is shown.
- **Configure:** Pressing this button brings up a dialog where you can choose colors for the three node types, and comments, and fonts for node labels and comments. Each choice is made through standard Windows Font and Color choosers. These choices will not affect the information in the presentation, but will affect all new nodes created from that point on. However, the choices will be saved in the Windows registry on exiting ARCEdit, so that they are not lost. The configure dialog is shown in Figure 5.

Below the commands is a frame containing radio buttons for selecting a mode--one of concept, relation, actor, link, comment, edit or jump--and then a textbox for typing labels into.

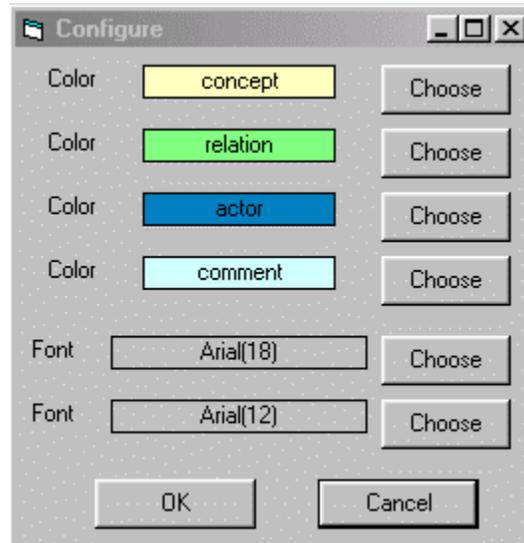


Figure 5: Configure dialog

2.5. Main Window Choices

The radio buttons are mutually exclusive, i.e. only one can be active at any one time. They act as a set of mode buttons and allow the following choice actions:

- **C:** Pressing this radio button means that the next node that is placed will be a concept.
- **R:** Pressing this radio button means that the next node that is placed will be a relation.
- **A:** Pressing this radio button means that the next node that is placed will be an actor.
- **Link:** Pressing this radio button means that link mode is entered in which nodes can be linked.
- **Comment:** Pressing this radio button means that the next node that is placed will be a comment.
- **Edit:** Pressing this radio button means that the label in any node can be changed.
- **Jump:** Pressing this radio button means that a label in a concept box can be interpreted as referring to a nested context or to a concept in an outer context.

The text box below the radio button frame is the point of action of the dialog. A description of each action is described under the next section. The basic idea is that for some actions a label can be typed into the box and then by pressing F4, the label can be used according to which radio button is selected. For other actions, pressing F4 will result in a label being placed in the text box.

3. Actions

This section discusses how to carry out each of the chosen actions from the radio button frame. Each of the following subsections describe the steps needed for each action.

3.1. Adding A New Node

When wanting to add a new node to the graph perform the following steps (they may be repeated):

1. Select the node type: **C**, **R**, or **A**.
2. Type a label into the text box.
3. Move the mouse to the chosen spot on the slide.
(Do NOT click mouse; changes focus to Powerpoint from ARCEdit.)
4. Press F4.

The new node will appear on the slide with the label from the text box. The node type will stay the same, making it easy to add several concepts. Then to add several relations, select the **R** radio button; or to add several actors, select the **A** radio button.

3.2. Adding A Comment

When wanting to add a comment to the graph perform the following steps (they may be repeated):

1. Select Comment.
2. Move the mouse to the desired position on the slide.
(*Do NOT click mouse.*)
3. Press F4.
A callout shape will appear at the mouse position containing a dummy comment, which will be highlighted.
4. Click anywhere in the comment.
This activates the slide so that text can be typed directly into the comment shape to replace the dummy text.
5. Click anywhere outside of comment box.
This saves the text of the comment on the slide.

When you are finished activate the ARCEdit dialog again by clicking its title bar.

3.3. Adding A Link

When wanting to add a link between two nodes in the same graph perform the following steps (they may be repeated):

1. Select Link.
2. Move the mouse to the "from" node.
(*Do NOT click mouse.*)
3. Press F4.
Make sure the node's label appears in the text box.
4. Move the mouse to the "to" node.
(*Do NOT click mouse.*)
5. Press F4.
The nodes are linked with a line and an arrow at the "to" end.

When wanting to add a link between a concept to a nested graph perform the following steps (they may be repeated):

1. Select Link.
2. Move the mouse to the "from" node. This must be a concept node.
(*Do NOT click mouse.*)
3. Press F4.
Make sure the node's label appears in the text box.
4. Select the slide with the nested graph.
(This slide must be before the slide with the "from" node on it in the presentation. If not move it to an earlier position.)
5. Activate the ARCEdit dialog again by clicking its title bar.
6. Move the mouse to the "to" node. This must be a concept node in the graph.
(*Do NOT click mouse.*)
7. Press F4.
The graph is now linked with the "from" concept; the concept now has an arrow on its end.

3.4. Changing A Label

When wanting to partly change the label in a node in a graph perform the following steps:

1. Select Edit.
2. Move the mouse to the node containing the label to be changed.
(Do NOT click mouse.)
3. Press F4.
The node's label will appear in the text box.
4. Edit the label in the text box.
(Do NOT click mouse.)
5. Press F4.
This replaces the node's label.

When wanting to change the whole label in a node in a graph perform the following steps:

1. Select Edit.
2. Insert a new label in the text box.
3. Move the mouse to the node containing the label to be changed.
(Do NOT click mouse.)
4. Press F4.
This replaces the node's label.

For both these options, this change is destructive. There is no undo.

3.5. Jumping To A Context Graph

When needing to jump to a nested context graph from a node in the current graph perform the following steps:

1. Select Jump.
2. Look on the current slide for a concept containing a hyperlink label (the rectangle has a diamond on one side).
3. Move the mouse to the node.
(Do NOT click mouse.)
4. Press F4.
The graph referred to in the hyperlink will replace the current one.

4. CGIF

The Conceptual Graph Interchange Format (CGIF) is intended for easy interchange of conceptual graphs and is a maintained standard [7].¹ It was purposely left without graphical information such as positions, fonts, colors and shapes. However, for a graphical editor such as ARCEdit, it makes sense to try to preserve the essential information from the graphical form across editing sessions. Without altering the standard syntax, the only way this can be done is to utilize the commenting facility, where arbitrary information can be inserted and used. This is done in two ways by ARCEdit:

- the graph defining label and type title are stored as a beginning comment for a graph.
- the position of each box (concept, relation or actor) is stored with the box position as a comment.

A CGIF file that does not adhere to these conventions is broken apart into separate slides so that each slide only contains one graph, and never shows a nested context graph except as a coreference link. If the CGIF file contains no contexts at the top level, then one is created for it.

¹ See <http://www.cs.nmsu.edu/~hdp/CGTools/cgstand/cgstandnmsu.html> for a current updated standard for workshop.

Luckily CGIF gives us the flexibility to represent the same graph or graphs in multiple different ways without altering the semantics. Also, a graph with no positional information ends up with every box placed on top of every other; one can use the Powerpoint editor to move the boxes to make a readable slide.

4.1. Deviations from CGIF Standard Syntax

Every effort has been made to accommodate any valid CGIF file. However there are a few points where ARCEdit files differ from the standard. The use of the node comment to store graph and positional information is the first. Below is an example for each:

- the graph defining label and type title example:
a graph with title "G1 is GRAPH" is written to the CGIF file as `;%GRAPH*G1%;`. The semicolon introduces the comment, the percent sign indicates the title of the graph, while the asterisk separates the graph type and graph defining label. The second percent sign indicates the end of title in case there is more comment, and the second semicolon ends the comment.
- the position of each node example:
a concept Cat with referent 'Fred', positioned at $x = 100$, $y = 200$ is written to the CGIF file as `[Cat:'Fred';100@200;]`. The semicolon introduces the comment, the `at' sign separates x and y coordinates. When the CGIF file is imported the positional information is used to place the concept box on the slide.

The nested graph `(r1 [c1: [...]] ...)` would be altered to be represented as `;%GRAPH*G1%; (r1 [c1:;%GRAPH*G2%; [...]] ...)`. Also the displayed graph for the CGIF graph with no context at all `:[c1*x] [c2*y] (r1 ?x ?y)` becomes `;%GRAPH*G1%;(r1[c1][c2])`.

The second deviation is technicality concerning actors and the extended sequence notation. These both use the diamond brackets, `<...>`. Since a context graph can contain actors, the form `[Graph:<a1 [c1*x] [c2*y]> (r1 ?x ?y) ...]` is possible. However, distinguishing between this and the sequence form in a referent, e.g. `[c1: <1,2,3>]` needs a backtracking parser, whereas the rest of the CGIF grammar does not. For this reason, the graph must be written as `[Graph: (r1 [c1*x] [c2*y]) <a1 ?x ?y> ...]`, with the relation arc before the actor arc. This will be discussed more with the CGIF community.

4.2. Importing and Exporting CGIF Files

Although ARCEdit graphs may be saved as a Powerpoint presentation with no loss of syntactic or semantic content, they may also be saved and loaded again as CGIF text files. The `File` item of the main window dialog menu contains Import and Export options to initiate these actions. The Import item brings up a standard Windows Open file dialog for files of extension `.cgf`. The Export item brings up a Save As file dialog so that all CG content of all slides may be saved in CGIF form.

5. Conclusions and Future Work

ARCEdit is still in a beta test mode. There are some errors that occur, but most commands and actions work as advertised if one follows the instructions or steps given in help menus. Should become a stable release soon.

ARCEdit is currently a syntactic editor only for CGIF files. The only semantics that it does is in relationship to the separation of graphs on to slides and in setting up linkages between coreference links. Currently to link a concept to a nested graph, one must select a concept in the nested graph/slide to link the whole graph. We would like to add a linkage to the defining label in the graph label.

The next elements that we wish to add is semantic processing for type and relation hierarchies and lambda definitions of graphs. Also in the syntactic area of the editor, we will be adding translation from (import) and to (export) XML code.

Appendix A - Installation

The installation program automates the process of installing the software, registering it as a Powerpoint add-in and providing a toolbar button for running the add-in. You may also want to enable manual loading and unloading of the add-in by installing the "COM Add-in" button. This is done by selecting in Tools/Customize in Powerpoint, selecting Tools from the Categories list, and dragging the button labeled "COM Add-Ins..." to any visible toolbar. Pressing the button brings up a list of add-ins, each with a check box beside it. Checking or unchecking the box will load or unload the add-in.

References

- [1] Glinert, E. (1990) Visual Programming Environments, (2 Vols). Los Alamitos, CA: IEEE Computer Science Press.
- [2] Hartley, R.T., H.D. Pfeiffer, and Dihong Qui (1992) Representation for ViewGen: Structures and Reasoning, in Workshop on Propositional Knowledge Representation, Stanford, CA.
- [3] Pfeiffer, H.D. and R.T. Hartley (1991) The Conceptual Programming Environment, CP: Reasoning Representation using Graph Structures and Operations, in Proceedings of IEEE Workshop on Visual Languages, Kobe, Japan.
- [4] Pfeiffer, H.D. and R.T. Hartley (2000) Visual CP Representation of Knowledge, in Supplemental Proceedings of the 8th International Conference On Conceptual Structures, Darmstadt, Germany.
- [5] Price, B.A., R.M. Baecker and I.S. Small (1993) A Principled Taxonomy of Software Visualization, in Journal of Visual Languages and Computing Vol. 4 No. 3, pp. 211-266.
- [6] Sowa, J.F. (1984) Conceptual Structures. Reading, MA: Addison-Wesley.
- [7] Sowa, J.F. et al (2001) Conceptual Graph Standard, American National Standard NCITS.T2/ISO/JTC1/SC 32 WG2 N 000. [Access Online: April 2001], URL: <http://www.bestweb.net/~sowa/cg/cgstand.htm>.